Philipp Cimiano Oscar Corcho
Valentina Presutti Laura Hollink
Sebastian Rudolph (Eds.)

LNCS 7882

The Semantic Web: Semantics and Big Data

10th International Conference, ESWC 2013
Montpellier, France, May 2013
Proceedings

Lecture Notes in Computer Science 7882

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Philipp Cimiano Oscar Corcho
Valentina Presutti Laura Hollink
Sebastian Rudolph (Eds.)

The Semantic Web: Semantics and Big Data

10th International Conference, ESWC 2013
Montpellier, France, May 26-30, 2013
Proceedings

Springer

Volume Editors

Philipp Cimiano
University of Bielefeld, Germany
E-mail: cimiano@cit-ec.uni-bielefeld.de

Oscar Corcho
Universidad Politécnica de Madrid, Boadilla del Monte, Spain
E-mail: ocorcho@fi.upm.es

Valentina Presutti
National Research Council, Rome, Italy
E-mail: valentina.presutti@cnr.it

Laura Hollink
Vrije Universiteit Amsterdam, The Netherlands
E-mail: l.hollink@vu.nl

Sebastian Rudolph
Technische Universität Dresden, Germany
E-mail: sebastian.rudolph@tu-dresden.de

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-38287-1 e-ISBN 978-3-642-38288-8
DOI 10.1007/978-3-642-38288-8
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2013937518

CR Subject Classification (1998): H.3, I.2.4, I.2.6-7, H.2, H.5, H.4, I.7.4

LNCS Sublibrary: SL 3 – Information Systems and Application,
incl. Internet/Web and HCI

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

ESWC celebrated its 10th anniversary this year and took place in Languedoc-Roussillon, in the beautiful city of Montpellier, during May 26–30.

As motto for this year's edition of the conference, we chose the topic "Semantics and Big Data." Big Data Analytics is one of the top technological trends identified by Gartner Inc., stressing the increasingly important role of information governance. The capability of processing, analyzing, and interpreting large amounts of continuously changing, complex, and heterogeneous data is a challenge that will engage the research community's attention for the years to come. The ESWC community can and should contribute to this challenge, focusing in particular on the role that semantic technologies can play in the strive for scalable interpretation of not only large but also heterogeneous and complex data with many explicit and implicit relations.

The 10th edition of ESWC featured an exciting program including four keynotes by Enrico Motta (Knowledge Media Institute, Open University), David Karger (MIT), and Manfred Hauswirth (National University of Galway) as well as by Márta Nagy-Rothengass from the European Commission, whose invited talk took place during the EU Project Networking Session. In addition, the conference program featured 13 workshops, 7 tutorials as well as a PhD Symposium, an EU Project Networking Session, a panel on the motto of the conference, and a Semantic Mashup Challenge. The main program of the conference was rounded off by a demonstration and poster session in which researchers had the chance to present their latest results and advances in the form of live demos. We are also happy that OWLED decided to collocate again their annual workshop with ESWC to discuss future directions for the Web Ontology Language.

The program of the conference comprised 42 contributed paper talks (37 research and 5 in-use papers), selected among 162 submissions, which corresponds to an acceptance rate of 25.9%. This year the reviewing procedure was improved in terms of transparency and quality. We introduced a rebuttal phase that in some cases was crucial for taking final decisions, and a limited number of papers were accepted after a second reviewing round aimed at verifying specific acceptance conditions. The PhD Symposium attracted 23 submissions, of which 7 were accepted as full papers for oral presentation and 11 as short papers for poster presentation, corresponding to an acceptance rate of 30%. In order to foster interaction with other disciplines and to inspire the ESWC research community to venture into new problems and challenges, the conference also featured a special track on "Cognition and the Semantic Web."

As General and PC Chairs we would like to thank everybody who was involved in the organization of ESWC 2013. First of all, our thanks go to the Local Chairs François Scharffe and Clement Jonquet for doing a great job with the local arrangements of the conference, but also in the acquisition of addi-

tional funding and sponsoring. Further, we would like to thank all our Track Chairs who played a key role in helping the PC Chairs to select and compile an outstanding technical program: Aldo Gangemi, Eva Blomqvist, Pascal Hitzler, Luciano Serafini, María Esther Vidal, Axel Polleres, Jun Zhao, Jens Lehmann, Marta Sabou, Andreas Hotho, Alfio Gliozzo, Malvina Nissim, Josiane Parreira, Payam Barnaghi, Claudia d'Amato, Dunja Mladenic, Terry Payne, José Luis Ambite, Sören Auer, Peter Boncz, Krzysztof Janowicz, Kai-Uwe Kühnberger, Sofia Angeletou, and José Manuel Gómez-Pérez.

Special thanks go to this year's PhD Symposium Chairs Laura Hollink and Sebastian Rudolph, who gave their very best to contribute to the progress and education of our research offspring. We would also like to thank our Workshop Chair, Johanna Völker, as well as our Tutorial Chair, Stefan Schlobach, for putting together an exciting tutorial and workshop program that attracted the interest of many attendees of the conference. Vanessa Lopez and Miriam Fernández did an excellent job in selecting a number of very interesting and relevant posters and demos for the conference. We are very happy that Brigitte Endres-Niggemeyer, Giuseppe Di Fabbrizio, and Ioannis Papadakis kindly agreed to organize again the AI Mashup Challenge, this year with an emphasis on "Semantic and Intelligent Mashups." We would also like to thank Marko Grobelnik for chairing a panel on the motto of the conference and Achim Rettinger for organizing the European Project Networking Session. We are very grateful to Fabien Gandon as our Publicity and Communication Chair for spreading news about the conference in a timely manner and to Axel Ngonga as our Sponsorship Chair for his help with the acquisition of sponsoring.

This conference would not have been possible without the support of STI International. We thank Serge Tymaniuk from STI for administrating the website. Thanks also to our treasurer and financial officer Alex Wahler from STI for diligently taking care of the budget and financial/administrative issues.

We would also like to acknowledge the great work of youvivo GmbH, in particular of Edith Leitner and Martina Hartl, in organizing the conference. Thanks also to our Proceedings Chair, Katja Temnow, who made it possible that you are reading these lines right now. We are grateful to our Metadata Chairs: Dieter Fensel, Birgit Leitner, Alex Oberhauser, and Cord Wiljes.

Last but not least, we would like to thank all our sponsors. You will find their logos on the following pages. We kindly acknowledge the support of Springer in the preparation of this volume as well as a gift from Springer as prize for the AI Mashup Challenge.

March 2013

Philipp Cimiano
Valentina Presutti
Oscar Corcho

Organization

Organizing Committee

General Chair

Philipp Cimiano	Bielefeld University, Germany

Program Chairs

Oscar Corcho	Ontology Engineering Group, UPM, Spain
Valentina Presutti	Institute of Cognitive Sciences and Technologies, Italy

Local Chairs

Clement Jonquet	LIRMM, Université Montpellier 2, France
François Scharffe	LIRMM, Université Montpellier 2, France

Poster and Demo Chairs

Vanessa Lopez	IBM, Dublin, Ireland
Miriam Fernandez	The Knowledge Media Institute, Open University, UK

Workshops Chair

Johanna Völker	University of Mannheim, Germany

Tutorials Chair

Stefan Schlobach	VU University of Amsterdam, The Netherlands

PhD Symposium Chairs

Laura Hollink	VU University of Amsterdam, The Netherlands
Sebastian Rudolph	Karlsruhe Institute of Technology, Germany

Semantic Technologies Coordinators

Dieter Fensel	STI Innsbruck, Austria
Birgit Leiter	STI Innsbruck, Austria
Alex Oberhauser	STI Innsbruck, Austria
Cord Wiljes	Bielefeld University, Germany

Project Networking Session Chair

Achim Rettinger	Karlsruhe Institute of Technology, Germany

Sponsorship Chair

Axel Ngonga	University of Leipzig, Germany

Publicity Chair

Fabien Gandon	INRIA, France

Panel Chair

Marko Grobelnik	Jozef Stefan Institute Ljubljana, Slovenia

Proceedings Chair

Katja Temnow	Bielefeld University, Germany

Treasurer

Alexander Wahler	STI, Germany

Local Organization and Conference Administration

STI International	Austria

Webmaster

STI International	Austria

Program Committee

Program Chairs

Oscar Corcho	Ontology Engineering Group, UPM, Spain
Valentina Presutti	Institute of Cognitive Sciences and Technologies, Italy

Track Chairs

Ontologies Track

Eva Blomqvist	Linköping University, Sweden
Aldo Gangemi	LIPN-Paris 13-Sorbonne Cité, France and STLab ISTC-CNR, Italy

Reasoning Track

Pascal Hitzler	Wright State University Dayton, Ohio, USA
Luciano Serafini	FBK, Trento, Italy

Semantic Data Management Track

María Esther Vidal	Universidad Simón Bolivar, Venezuela
Axel Polleres	Siemens AG, Austria

Linked Open Data Track

Jun Zhao	University of Oxford, UK
Jens Lehmann	University of Leipzig, Germany

Social Web and Web Science Track

Marta Sabou	MODUL University Vienna, Austria
Andreas Hotho	University of Würzburg, Germany

Natural Language Processing and Information Retrieval Track

Alfio Gliozzo	IBM T.J. Watson Research Center, USA
Malvina Nissim	University of Bologna, Italy

Mobile Web, Sensors, and Semantic Streams Track

Josiane Parreira	DERI, Ireland
Payam Barnaghi	University of Surrey, UK

Machine Learning Track

Claudia d'Amato	University of Bari, Italy
Dunja Mladenic	Jozef Stefan Institute Ljubljana, Slovenia

Special Track: Cognition and Semantic Web

Krzysztof Janowicz	University of California, Santa Barbara, USA
Kai-Uwe Kühnberger	University of Osnabrück, Germany

In-use and Industrial Track

Sofia Angeletou	BBC, UK
José Manuel Gomez-Pérez	iSOCO, Spain

Members (All Tracks)

Fabian Abel
Sven Abels
Benjamin Adams
Harith Alani
Renzo Angles
Lora Aroyo
Sören Auer
Nathalie Aussenac-Gilles
Claudio Baldassarre
Jie Bao
Pierpaolo Basile
Matthias Bauer
Sean Bechhofer
Paolo Bellavista
Dominik Benz
Bettina Berendt
Michael K. Bergman
Luis Bermudez
Isabelle Bichindaritz
Antonis Bikakis
Christian Bizer
Gosse Bouma
Paolo Bouquet
Marco Brambilla
Charalampos Bratsas
John Breslin
Christopher Brewster
Dan Brickley
Lorenz Bhmann
Paul Buitelaar
Elena Cabrio
Jean-Paul Calbimonte
David Carral
Gerard Casamayor
Jean Charlet
Vinay K. Chaudhri
Gong Cheng
Vassilis Christophides
Lin Clark
Michael Compton
Bonaventura Coppola
Paulo Costa
Isabel Cruz
Bernardo Cuenca Grau
John Davies
Brian Davis
Ernesto William De Luca
Emanuele Della Valle
Gianluca Demartini
Ian Dickinson
Stefan Dietze
Leigh Dodds
Martin Dzbor
Thomas Eiter
Henrik Eriksson
Orri Erling
Vadim Ermolayev
Jérôme Euzenat
James Fan
Nicola Fanizzi
Tim Finin
Flavius Frasincar
Irini Fundulaki
Fabien Gandon
Roberto Garcia
Manolis Gergatsoulis
Chiara Ghidini
Claudio Giuliano
Asunción Gómez-Pérez
John Goodwin
Jorge Gracia
Tom Grahame
Alasdair J.G. Gray
Gregory Grefenstette
Pierre Grenon
Gunnar Grimnes
Marko Grobelnik
Gerd Gröner
Paul Groth
Michael Gruninger
Christophe Guéret
Giancarlo Guizzardi
Peter Haase
Harry Halpin
Andreas Harth
Olaf Hartig

Oktie Hassanzadeh
Marek Hatala
Ralf Heese
Sebastian Hellmann
Cory Henson
Martin Hepp
Michiel Hildebrand
Rinke Hoekstra
Aidan Hogan
Matthew Horridge
Eero Hyvönen
Giovambattista Ianni
Robert Isele
Prateek Jain
Anja Jentzsch
Kristian Kersting
Haklae Kim
Ross King
Matthias Klusch
Spyros Kotoulas
Manolis Koubarakis
Markus Krötzsch
Udo Kruschwitz
Werner Kuhn
Christoph Lange
Josep-L. Larriba-Pey
Agnieszka Lawrynowicz
Danh Le Phuoc
Alessandro Lenci
Wenwen Li
Dong Liu
Yong Liu
Vanessa Lopez
José-Manuel López-Cobo
Frederick Maier
David Martin
Jarred McGinnis
Alexander Mehler
Roger Menday
Pablo N. Mendes
Sergi Mesquida
Peter Mika
Pascal Molli
Enrico Motta
Roberto Navigli
Adeline Nazarenko
Matteo Negri
Thomas Neumann
Axel-Cyrille Ngonga Ngomo
Matthias Nickles
Andriy Nikolov
Olaf Noppens
Andrea Giovanni Nuzzolese
Alessandro Oltramari
Jeff Z. Pan
Massimo Paolucci
Alexandre Passant
Carlos Pedrinaci
Tassilo Pellegrini
Lushan Han
Silvio Peroni
Ricardo Pietrobon
H. Sofia Pinto
Marco Pistore
Guilin Qi
Yuzhong Qu
Jorge-Arnulfo Quiané Ruiz
Yves Raimond
Dnyanesh Rajpathak
Jinghai Rao
Paul Rissen
Massimo Romanelli
Riccardo Rosati
Matthew Rowe
Edna Ruckhaus
Sebastian Rudolph
Carlos Ruiz
Harald Sack
Manuel Salvadores
Kurt Sandkuhl
Kai-Uwe Sattler
Bernhard Schandl
François Scharffe
Thomas Scharrenbach
Simon Scheider
Ansgar Scherp
Stefan Schlobach
Ute Schmid
Daniel Schwabe
Juan F. Sequeda

Barış Sertkaya
Amit Sheth
Pavel Shvaiko
Elena Simperl
Evren Sirin
Steffen Staab
Claus Stadler
Milan Stankovic
Thomas Steiner
Markus Strohmaier
Heiner Stuckenschmidt
Mari Carmen Suárez-Figueroa
Fabian M. Suchanek
Vojtěch Svátek
György Szarvas
Pedro Szekely
Kerry Taylor
Andreas Thor
Sara Tonelli
Thanh Tran
Volker Tresp
Tania Tudorache
Jörg Unbehauen
Jacopo Urbani
Alejandro A. Vaisman
Herbert Van De Sompel
Willem Robert Van Hage
Boris Villazón-Terrazas
Johanna Völker
Piek Vossen
Denny Vrandecic
Wei Wang
Chang Wang
Chonggang Wang
Wei Wang
Erik Wilde
Gregory Todd Williams
Fouad Zablith
Antoine Zimmermann

Referees

Jonathan van Pumbrouck
Tope Omitola
Johannes Knopp
László Török
Alex Stolz
Bene Rodriguez
Aibo Tian
Martin Ugarte
Lorena Etcheverry
Kalpa Gunaratna
Armin Haller
Eufemia Tinelli
Sanjaya Wijeratne
Maryam Panahiazar
Mathaios Damigos
Eleftherios Kalogeros
Jedrzej Potoniec
Maribel Acosta
Mauro Dragoni
Sergio Tessaris
Marco Rospocher
Pablo Rodríguez-Mier

Steering Committee

Chair

John Domingue — KMI, The Open University, UK and STI International, Austria

Members

Grigoris Antoniou	Forth, Greece
Lora Aroyo	VU University of Amsterdam, The Netherlands
Fabio Ciravegna	University of Sheffield, UK
Eero Hyvnen	Aalto University, Finland
Axel Polleres	Siemens AG, Austria
Elena Simperl	University of Southampton, UK
Paolo Traverso	FBK, Center for Information Technology, IRST, Italy

Sponsoring Institutions


ONTOLOGOS CORP.
WWW.ONTOLOGOS-CORP.COM


ontotext


PeerAssist


Planet Data


PortDial


PRELIDA
PRESERVING LINKED DATA


RENDER
Reflecting Knowledge Diversity


ROBUST


SIFR project


videolectures.net
exchange ideas & share knowledge


SUPPORT

YAHOO! LABS

Local Sponsors

cnrs

ENJOY Montpellier

Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier
LIRMM

la Région Languedoc Roussillon

M Montpellier

POLYTECH MONTPELLIER

UNIVERSITÉ MONTPELLIER 2
SCIENCES ET TECHNIQUES

A Semantic Web for End Users

David Karger

MIT, Cambridge, USA
karger@mit.edu

For whom are we creating the Semantic Web? As we wrestle with our ontologies, alignments, inference methods, entity extractions and triple stores, it is easy to lose track of the vast majority of users who have no idea what any of these things are, who they help, or what problems they will solve.

In this talk, I will adopt the perspective of these end users. I will identify a number of information management problems faced by them – such as organizing their personal information, communicating effectively on the web, and handling their incoming information overload. The Semantic Web can play a key role in solving these problems. But what will matter most to end users is not the details of the Semantic Web's syntax, model, or algorithms, but rather the interfaces and workflows through which end users interact with it. I will describe key characteristics of these interfaces and workflows, and offer an overview of the research that needs to be done to develop them as effective solutions for end users.

What Does It Mean to Be Semantic? On the Effective Use of Semantics in the Semantic Web

Enrico Motta

Knowledge Media Institute, The Open University, UK
enrico.motta@open.ac.uk

Twelve years after the publication of the seminal article by Tim Berners-Lee, James Hendler and Ora Lassila, which expounded the vision of a Semantic Web characterised by dynamic and large scale agent interoperability, the Semantic Web still distinctly lacks a wow factor. Many SW applications exist, but too often they are characterised by few data sources put together at compile time to drive some relatively simple user functionality. In many cases it is difficult to identify the competitive advantage that being semantic affords these applications, compared to systems built using conventional technologies. Of course, one could argue that this is not necessarily a problem: the success of an area is measured in terms of its academic vitality and its impact on commerce and society. However, I would argue that there is actually a problem here and in my talk I will analyse these issues by examining how the notion of semantics is used in our community, highlighting the productive and unproductive uses of the term, and in particular describing the different ways in which semantics can be effectively exploited to provide added value to applications. The key message is that while there are many ways to exploit semantics to develop better functionalities, as a community we need to develop a better understanding (both fundamentally and pragmatically) of the value proposition afforded by the use of semantics. Without such understanding there is a risk that we will fail to take full advantage of the technologies that we are developing and the opportunities they create for us.

It's a Dynamic World – Ubiquitous Streams and the Linked Data Web

Manfred Hauswirth

Digital Enterprise Research Institute (DERI), Ireland
manfred.hauswirth@deri.org

It is well established that we produce humongous amounts of information – technical infrastructures (smart grid, smart cities), the Social Web (Twitter, social networks, blogs), information systems (e-commerce, e-health), the media (newspapers, broadcasters), the Internet of Things, mobile phones, and many more – and that these amounts are growing exponentially. Linked Data gives us the technical means to network all this information and enables us to develop new forms of analytics on networked data from many sources instead of traditional "monolithic" data analytics. But this network of information is "in-discrete" as the data is produced continuously and at potentially high speeds with varying loads and demands on the producer and the consumer sides. This calls for new data/knowledge management approaches and as a result, the Linked Data world is slowly moving from a simplifying discrete model to a more realistic continuous view. This development impacts on and changes research problems in all areas and for all layers and requires well-orchestrated research efforts in and across research communities to support "streaming" as an integrated paradigm. In this talk, I will present a comprehensive stack of Linked Stream management approaches for all layers – from the Internet of Things to backend information systems, and will discuss the impact of streams on big data, analytics, and privacy.

Table of Contents

Research Track

Ontologies

Linked Open Data

Semantic Data Management

Mobile Web, Sensors and Semantic Streams

Reasoning

Natural Language Processing and Information Retrieval

Machine Learning

Social Web and Web Science

Cognition and Semantic Web

In-Use and Industrial Track

PhD Symposium

A Unified Approach for Aligning Taxonomies and Debugging Taxonomies and Their Alignments

Valentina Ivanova and Patrick Lambrix

Department of Computer and Information Science and the Swedish e-Science Research Centre
Linköping University, 581 83 Linköping, Sweden

Abstract. With the increased use of ontologies in semantically-enabled applications, the issues of debugging and aligning ontologies have become increasingly important. The quality of the results of such applications is directly dependent on the quality of the ontologies and mappings between the ontologies they employ. A key step towards achieving high quality ontologies and mappings is discovering and resolving modeling defects, e.g., wrong or missing relations and mappings. In this paper we present a unified framework for aligning taxonomies, the most used kind of ontologies, and debugging taxonomies and their alignments, where ontology alignment is treated as a special kind of debugging. Our framework supports the detection and repairing of missing and wrong is-a structure in taxonomies, as well as the detection and repairing of missing (alignment) and wrong mappings between ontologies. Further, we implemented a system based on this framework and demonstrate its benefits through experiments with ontologies from the Ontology Alignment Evaluation Initiative.

1 Motivation

To obtain high-quality results in semantically-enabled applications such as the ontology-based text mining and search applications, high-quality ontologies and alignments are both necessary. However, neither developing nor aligning ontologies are easy tasks, and as the ontologies grow in size, it is difficult to ensure the correctness and completeness of the structure of the ontologies. For instance, some structural relations may be missing or some existing or derivable relations may be unintended. This is not an uncommon case. It is well known that people who are not expert in knowledge representation often misuse and confuse equivalence, is-a and part-of (e.g., [2]). Further, ontology alignment systems are used for generating alignments and, as shown in the Ontology Alignment Evaluation Initiative (OAEI, http://oaei.ontologymatching.org/), alignments usually contain mistakes and are incomplete. Such ontologies and alignments, although often useful, lead to problems when used in semantically-enabled applications. Wrong conclusions may be derived or valid conclusions may be missed.

A key step towards high-quality ontologies and alignments is debugging the ontologies and alignments. During the recent years several approaches have been proposed for debugging semantic defects in ontologies, such as unsatisfiable concepts or inconsistent ontologies (e.g., [24,14,15,8]) and related to mappings (e.g., [22,11,23,28]) or integrated ontologies [13]. Further, there has been some work on detecting modeling defects (e.g., [9,3]) such as missing relations, and repairing modeling defects [19,18,16].

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 1–15, 2013.

The increased interest in this field has also led to the creation of an international workshop on this topic [20]. In a separate sub-field of ontology engineering, ontology alignment, the correctness and completeness of the alignments has traditionally received much attention (e.g., [25]). Systems have been developed that generate alignments and in some cases validation of alignments is supported.

In this paper we propose a unified approach for ontology debugging and ontology alignment, where ontology alignment can be seen as a special kind of debugging. We propose an integrated framework that, although it can be used as an ontology debugging framework or an ontology alignment framework, presents additional benefits for both and leads to an overall improvement of the quality of the ontologies and the alignments. The ontology alignment provides new information that can be used for debugging and the debugging provides new information that can be used by the ontology alignment. Further, the framework allows for the interleaving of different debugging and alignment phases, thereby in an iterative way continuously generating new information and improving the quality of the information used by the framework.

In sections 3, 4, 5 and 6 we propose our unified approach for ontology alignment and debugging. To our knowledge this is the first approach that integrates ontology debugging and ontology alignment in a uniform way and that allows for a strong interleaving of these tasks. We present a framework (Section 3), algorithms for the components (Sections 4 and 5) and their interactions (Section 6). Further, we show the advantages of our approach in Section 7 through experiments with the ontologies and alignment of the OAEI 2011 Anatomy track. Related work is given in Section 8 and the paper concludes in Section 9. However, we start with some preliminaries.

2 Preliminaries

In this section we introduce notions that are needed for our approach. This paper focuses on **taxonomies** $\mathcal{O} = (\mathcal{C}, \mathcal{I})$, the most widely used type of ontologies, where $\mathcal{C}$ is a set of atomic concepts and $\mathcal{I} \subseteq \mathcal{C} \times \mathcal{C}$ represents a set of atomic concept subsumptions (is-a relations). In the following we use 'ontologies' and 'taxonomies' interchangeably. An **alignment** between ontologies $\mathcal{O}_i$ and $\mathcal{O}_j$ is represented by a set $\mathcal{M}_{ij}$ of mappings between concepts in different ontologies. The concepts that participate in mappings are called **mapped concepts**. Each mapped concept can participate in multiple mappings and alignments. We currently consider equivalence mappings ($\equiv$) and is-a mappings (subsumed-by ($\rightarrow$) and subsumes ($\leftarrow$)).

The output of ontology alignment systems are **mapping suggestions**. These should be validated by a domain expert and if accepted, they become part of an alignment.

Definition 1. *A* **taxonomy network** $\mathcal{N}$ *is a tuple* $(\mathbb{O}, \mathbb{M})$ *with* $\mathbb{O} = \{\mathcal{O}_k\}_{k=1}^{n}$ *the set of the ontologies in the network and* $\mathbb{M} = \{\mathcal{M}_{ij}\}_{i,j=1;i<j}^{n}$ *the set of representations for the alignments between these ontologies.*

Figure 1 shows a small ontology network with two ontologies (concepts are represented by nodes and the is-a structures are represented by directed edges) and an alignment

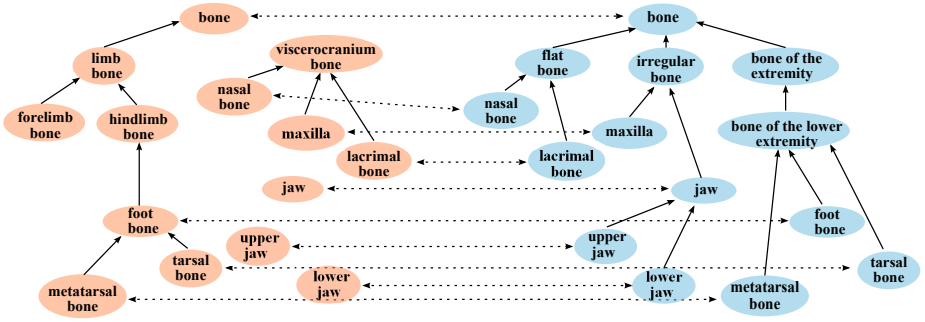


Fig. 1. (Part of an) Ontology network

(represented by dashed edges).[1] The alignment consists of 10 equivalence mappings. One of these mappings represents the fact that the concept *bone* in the first ontology is equivalent to the concept *bone* in the second ontology.

The domain knowledge inherent (logically derivable) in the network is represented by its **induced ontology**, an ontology that consists of the set of all concepts from the taxonomies, all asserted is-a relations in the taxonomies and all mappings.

In our algorithms we use **knowledge bases** (KBs) related to the taxonomies and taxonomy networks that allow us to do deductive inference.

3 Approach and Algorithms

Our framework consists of two major components - a debugging component and an alignment component. They can be used independently or in close interaction. The alignment component detects and repairs missing and wrong mappings between ontologies, while the debugging component additionally detects and repairs missing and wrong is-a structure in ontologies. Although we describe the two components separately, in our framework ontology alignment can be seen as a special kind of debugging.

The workflow (Figure 2) in both components consists of three phases during which wrong and missing is-a relations/mappings are detected (**Phase 1**), validated (**Phase 2**) and repaired (**Phase 3**) in a semi-automatic manner by a domain expert. Although the algorithms for repairing are different for missing and wrong is-a relations/mappings, the repairing goes through the same phases as shown in the figure - the generation of repairing actions (**Phase 3.1**), the ranking of is-a relations/mappings (**Phase 3.2**), the recommendation of repairing actions (**Phase 3.3**) and finally, the execution of repairing actions (**Phase 3.4**). In our approach we repair ontologies and alignments one at a time since dealing with all ontologies and alignments simultaneously would be infeasible. The is-a relations are handled in the context of the selected ontology, while the mappings are handled in the context of the selected alignment and its pair of ontologies.

[1] The first ontology is a part of AMA, the second ontology is a part of NCI-A, and the alignment is a part of the alignment between AMA and NCI-A as defined in OAEI 2011.

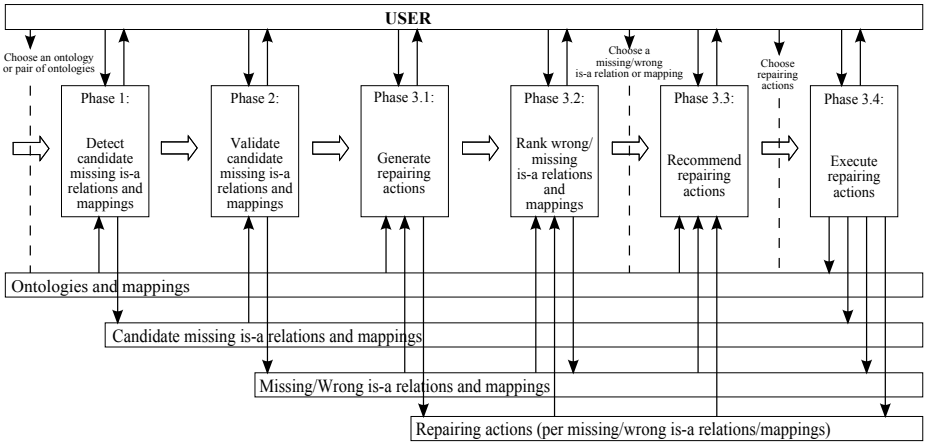


Fig. 2. Workflow

We note that at any time during the debugging/alignment workflow, the user can switch between different ontologies and the different phases shown in Figure 2. We also note that the repairing of defects often leads to the discovery of new defects, i.e., leading to additional debugging opportunities. Thus several iterations are usually needed for completing the debugging/alignment process. The process ends when no more missing or wrong is-a relations and mappings are detected or need to be repaired.

In the next three sections we describe the components and their interactions, and present algorithms for the different components and phases.

4 Debugging Component

The input for the debugging component is a taxonomy network, i.e., a set of taxonomies and their alignments. The output is the set of repaired taxonomies and alignments.

Phase 1: Detect Candidate Missing Is-a Relations and Mappings. In this component we focus on detecting wrong and missing is-a relations and mappings in the ontology network, based on knowledge that is inherent in the network. Therefore, given an ontology network, we use the domain knowledge represented by the ontology network to detect the deduced is-a relations and mappings in the network.

In our algorithm we initialize a KB for the ontology network (KB_N), KBs for each ontology (KB_k) and for each pair of ontologies and their alignment (KB_{ij}). For each ontology in the network, the set of **candidate missing is-a relations** (CMIs) derivable from the ontology network consists of is-a relations between two concepts of the ontology, which can be inferred using logical derivation from the domain knowledge inherent in the network, but not from the ontology alone. Similarly, for each pair of ontologies in the network, the set of **candidate missing mappings** (CMMs) derivable from the ontology network consists of mappings between concepts in the two ontologies, which can be inferred using logical derivation from the domain knowledge inherent in the network, but not from the two ontologies and their alignment alone.

Definition 2. *Let $\mathcal{N} = (\mathbb{O}, \mathbb{M})$ be an ontology network, with $\mathbb{O} = \{\mathcal{O}_k\}_{k=1}^{n}$, $\mathbb{M} = \{\mathcal{M}_{ij}\}_{i,j=1;i<j}^{n}$ and induced ontology $\mathcal{O}_N = (\mathcal{C}_N, \mathcal{I}_N)$. Let $\mathcal{O}_k = (\mathcal{C}_k, \mathcal{I}_k)$. Then, we define the following.*

(1) $\forall k \in 1..n$: $\mathbf{CMI}_k = \{(a,b) \in \mathcal{C}_k \times \mathcal{C}_k \mid \mathcal{O}_N \models a \rightarrow b \wedge \mathcal{O}_k \not\models a \rightarrow b\}$ *is the set of candidate missing is-a relations for $\mathcal{O}_k$ derivable from the network.*
(2) $\forall i,j \in 1..n, i < j$: $\mathbf{CMM}_{ij} = \{(a,b) \in (\mathcal{C}_i \times \mathcal{C}_j) \cup (\mathcal{C}_j \times \mathcal{C}_i) \mid \mathcal{O}_N \models a \rightarrow b \wedge (\mathcal{C}_i \cup \mathcal{C}_j, \mathcal{I}_i \cup \mathcal{I}_j \cup \mathcal{M}_{ij}) \not\models a \rightarrow b\}$ *is the set of candidate missing mappings for $(\mathcal{O}_i, \mathcal{O}_j, \mathcal{M}_{ij})$ derivable from the network.*
(3) $\mathbf{CMI} = \cup_{k=1}^{n}$ $\mathbf{CMI}_k$ *is the set of* **candidate missing is-a relations derivable from the network**.
(4) $\mathbf{CMM} = \cup_{i,j=1;i<j}^{n}$ $\mathbf{CMM}_{ij}$ *is the set of* **candidate missing mappings derivable from the network**.

In the network in Figure 1 the CMIs are *(nasal bone, bone)*, *(maxilla, bone)*, *(lacrimal bone, bone)*, *(jaw, bone)*, *(upper jaw, jaw)* and *(lower jaw, jaw)* in AMA, and *(metatarsal bone, foot bone)* and *(tarsal bone, foot bone)* in NCI-A.

Our algorithms for detecting CMIs/CMMs rely on the knowledge inherent in the network where the ontologies are connected in a network through mapped concepts. Thus the derivation paths of all CMIs and CMMs, which can be found using the knowledge inherent in the network, go through mapped concepts. Therefore, instead of checking whether the is-a relations between all pairs of concepts are derivable in the network, we only check all pairs of mapped concepts.[2,3]

Phase 2: Validate Candidate Missing Is-a Relations and Mappings. Since the structure of the ontologies may contain wrong is-a relations and the alignments may contain wrong mappings, some of the CMIs and CMMs may be derived due to some wrong is-a relations and mappings. Therefore they have to be validated by a domain expert. During Phase 2 the domain expert validates the CMIs/CMMs and partitions them into **wrong** and **missing** is-a relations/mappings. As an aid to the domain expert, we have developed recommendation algorithms based on the existence of is-a and part-of relations in the ontologies and external domain knowledge (WordNet [29] and UMLS [27]). In addition, the domain expert is provided with the derivation paths (*justifications*) for the CMI/CMM under validation.

In the network in Figure 1 *(upper jaw, jaw)* and *(lower jaw, jaw)* are validated as wrong since an upper/lower jaw is *a part-of* (not *is-a*) a jaw. The others are missing.

Phase 3: Repair Wrong and Missing Is-a Relations and Mappings. Once missing and wrong is-a relations and mappings have been obtained[4], we need to repair them. For each ontology in the network, we want to repair the is-a structure in such a way

[2] In the worst case scenario the number of mapped concept pairs is equal to the total number of concept pairs. In practice, the use of mapped concepts may significantly reduce the search space, e.g., when some ontologies are smaller than other ontologies in the network or when not all concepts participate in mappings. For instance, in the experiments in Section 7 the search space is reduced by almost 90%.

[3] For large ontologies or ontology networks, checking all pairs of concepts is also infeasible.

[4] Using the technique for detection described above or the techniques used by the alignment component or any other technique.

that (i) the missing is-a relations can be derived from their repaired host ontologies and for each pair of ontologies, we want to repair the mappings in such a way that (ii) the missing mappings can be derived from the repaired host ontologies of their mapped concepts and the repaired alignment between the host ontologies of the mapped concepts. Further (iii) the wrong is-a relations and (iv) the wrong mappings should no longer be derivable from the repaired ontology network. The notion of **structural repair** formalizes this. It contains is-a relations and mappings that should be added to or removed from the ontologies and alignments to satisfy these requirements. These is-a relations and mappings are called **repairing actions**.

Definition 3. *Let $\mathcal{N} = (\mathbb{O}, \mathbb{M})$ be an ontology network, with $\mathbb{O} = \{\mathcal{O}_k\}_{k=1}^{n}$, $\mathbb{M} = \{\mathcal{M}_{ij}\}_{i,j=1;i<j}^{n}$ and induced ontology $\mathcal{O}_N = (\mathcal{C}_N, \mathcal{I}_N)$. Let $\mathcal{O}_k = (\mathcal{C}_k, \mathcal{I}_k)$. Let $\mathcal{MI}_k$ and $\mathcal{WI}_k$ be the missing, respectively wrong, is-a relations for ontology $\mathcal{O}_k$ and let $\mathcal{MI}_N = \cup_{k=1}^{n}\mathcal{MI}_k$ and $\mathcal{WI}_N = \cup_{k=1}^{n}\mathcal{WI}_k$. Let $\mathcal{MM}_{ij}$ and $\mathcal{WM}_{ij}$ be the missing, respectively wrong, mappings between ontologies $\mathcal{O}_i$ and $\mathcal{O}_j$ and let $\mathcal{MM}_N = \cup_{i,j=1;i<j}^{n}\mathcal{MM}_{ij}$ and $\mathcal{WM}_N = \cup_{i,j=1;i<j}^{n}\mathcal{WM}_{ij}$. A* **structural repair for $\mathcal{N}$ with respect to** $(\mathcal{MI}_N, \mathcal{WI}_N, \mathcal{MM}_N, \mathcal{WM}_N)$*, denoted by $(\mathcal{R}^+, \mathcal{R}^-)$, is a pair of sets of is-a relations and mappings, such that*

(1) $\mathcal{R}^- \cap \mathcal{R}^+ = \emptyset$

(2) $\mathcal{R}^- = \mathcal{R}_M^- \cup \mathcal{R}_I^-$; $\mathcal{R}_M^- \subseteq \cup_{i,j=1,i<j}^{n}\mathcal{M}_{ij}$; $\mathcal{R}_I^- \subseteq \cup_{k=1}^{n}\mathcal{I}_k$

(3) $\mathcal{R}^+ = \mathcal{R}_M^+ \cup \mathcal{R}_I^+$; $\mathcal{R}_M^+ \subseteq \cup_{i,j=1,i<j}^{n}((\mathcal{C}_i \times \mathcal{C}_j) \setminus \mathcal{M}_{ij})$; $\mathcal{R}_I^+ \subseteq \cup_{k=1}^{n}((\mathcal{C}_k \times \mathcal{C}_k) \setminus \mathcal{I}_k)$

(4) $\forall k \in 1..n : \forall (a,b) \in \mathcal{MI}_k$: $(\mathcal{C}_k, (\mathcal{I}_k \cup (\mathcal{R}_I^+ \cap (\mathcal{C}_k \times \mathcal{C}_k))) \setminus \mathcal{R}_I^-) \models a \rightarrow b$

(5) $\forall i, j \in 1..n, i < j : \forall (a,b) \in \mathcal{MM}_{ij}$: $((\mathcal{C}_i \cup \mathcal{C}_j), (\mathcal{I}_i \cup ((\mathcal{C}_i \times \mathcal{C}_i) \cap \mathcal{R}_I^+) \cup \mathcal{I}_j \cup ((\mathcal{C}_j \times \mathcal{C}_j) \cap \mathcal{R}_I^+) \cup \mathcal{M}_{ij} \cup ((\mathcal{C}_i \times \mathcal{C}_j) \cap \mathcal{R}_M^+)) \setminus \mathcal{R}^-) \models a \rightarrow b$

(6) $\forall (a,b) \in \mathcal{WI}_N \cup \mathcal{WM}_N \cup \mathcal{R}^-$: $(\mathcal{C}_N, (\mathcal{I}_N \cup \mathcal{R}^+) \setminus \mathcal{R}^-) \not\models a \rightarrow b$

In our algorithm, at the start of the repairing phase we add all missing is-a relations and mappings to the relevant KBs. As these are validated to be correct, this is extra knowledge that should be used in the repairing process. Adding the missing is-a relations and mappings essentially means that we have repaired these using the least informative repairing actions ($\ll_I$ preference in [19]). Then during the repairing process we try to improve this and find more informative repairing actions. We say that a repairing action is more informative than another repairing action if adding the former to the ontology also allows to derive the latter. In general, more informative repairing actions that are correct according to the domain are preferred.

Definition 4. *Let (x_1, y_1) and (x_2, y_2) be two different is-a relations in the same ontology $\mathcal{O}$ (i.e., $x_1 \not\equiv x_2$ or $y_1 \not\equiv y_2$), then we say that (x_1, y_1) is* **more informative than** *(x_2, y_2) iff $\mathcal{O} \models x_2 \rightarrow x_1 \wedge y_1 \rightarrow y_2$.*

As an example, consider the missing is-a relation *(nasal bone, bone)* in Figure 1. Knowing that *nasal bone* $\rightarrow$ *viscerocranium bone*, according to the definition of more informative, we know that *(viscerocranium bone, bone)* is more informative than *(nasal bone, bone)*. As *viscerocranium bone* actually is a sub-concept of *bone* according to the domain, a domain expert would prefer to use the more informative repairing action.

Further, we initialize global variables for the current sets of missing ($\mathcal{MI}$) and wrong ($\mathcal{WI}$) is-a relations, and the current sets of missing ($\mathcal{MM}$) and wrong ($\mathcal{WM}$) mappings based on the validation results. Further, the sets of added ($\mathcal{R}_I^+$, $\mathcal{R}_M^+$) and removed

1. Compute $AllJust(w, r, \mathcal{O}_e)$
   where $\mathcal{O}_e = (\mathcal{C}_e, \mathcal{I}_e)$ such that $\mathcal{C}_e = \cup_{k=1}^{n} \mathcal{C}_k$ and
   $\mathcal{I}_e = ((\cup_{k=1}^{n} \mathcal{I}_k) \cup (\cup_{i,j=1;i<j}^{n} \mathcal{M}_{ij}) \cup \mathcal{MI}_N \cup \mathcal{MM}_N \cup \mathcal{R}_I^+ \cup \mathcal{R}_M^+) \setminus (\mathcal{R}_I^- \cup \mathcal{R}_M^-)$;
2. For every $\mathcal{I}' \in AllJust(w, r, \mathcal{O}_e)$:
   choose one element from $\mathcal{I}' \setminus (\mathcal{MI}_N \cup \mathcal{MM}_N \cup \mathcal{R}_I^+ \cup \mathcal{R}_M^+)$ to remove;

Fig. 3. Algorithm for generating repairing actions for wrong is-a relations and mappings

($\mathcal{R}_I^-$, $\mathcal{R}_M^-$) repairing actions for is-a relations and mappings, and the current sets of CMIs ($\mathcal{CMI}$) and CMMs ($\mathcal{CMM}$) are initialized to $\emptyset$.

Phase 3.1: Generate Repairing Actions. The structural repairs generated from the repairing algorithms below follow the preferences defined in [19].

Wrong Is-a Relations and Mappings. The algorithm for generating repairing actions (Figure 3) computes all justifications (AllJust) for all wrong is-a relations ($\mathcal{WI}$) and mappings ($\mathcal{WM}$). A justification for a wrong is-a relation or mapping can be seen as an explanation for why this is-a relation or mapping is derivable from the network.

Definition 5. *(similar definition as in [13]) Given an ontology $\mathcal{O} = (\mathcal{C}, \mathcal{I})$, and $(a, b) \in \mathcal{C} \times \mathcal{C}$ an is-a relation derivable from $\mathcal{O}$, then, $\mathcal{I}' \subseteq \mathcal{I}$ is a* **justification** *for (a, b) in $\mathcal{O}$, denoted by $\mathsf{Just}(\mathcal{I}', a, b, \mathcal{O})$ iff (i) $(\mathcal{C}, \mathcal{I}') \models a \rightarrow b$; and (ii) there is no $\mathcal{I}'' \subsetneq \mathcal{I}'$ such that $(\mathcal{C}, \mathcal{I}'') \models a \rightarrow b$. We use $\mathsf{All\_Just}(a, b, \mathcal{O})$ to denote the set of all justifications for (a, b) in $\mathcal{O}$.*

Our algorithm initializes a KB taking into account repairing actions up to now and computes the minimal hitting sets for each wrong is-a relation or mapping. The wrong is-a relation or mapping can then be repaired by removing at least one element in every justification.

In the network in Figure 1 *(upper jaw, jaw)* in AMA is validated as wrong. Its justification is *AMA:upper jaw* $\equiv$ *NCI-A:Upper_Jaw* $\rightarrow$ *NCI-A:Jaw* $\equiv$ *AMA:jaw*. To repair it *NCI-A:Upper_Jaw* $\rightarrow$ *NCI-A:Jaw* should be removed from NCI-A.

Missing Is-a Relations and Mappings. It was shown in [16] that repairing missing is-a relations (and mappings) can be seen as a generalized TBox abduction problem. Figure 4 shows our solution, an extension of the algorithm in [19], for the computation of repairing actions for a missing is-a relation or mapping. The main component of the algorithm (GenerateRepairingActions) computes, for a missing is-a relation or mapping, the more general concepts of the first concept (Source) and the more specific concepts of the second concept (Target) in the KB. To not introduce non-validated equivalence relations where in the original ontologies and alignments there are only is-a relations, we remove the super-concepts of the second concept from Source, and the sub-concepts of the first concept from Target. The already known wrong is-a relations or mappings and their repairing actions are removed from Repair (Source $\times$ Target). Adding an element from Repair to the KB makes the missing is-a relation or mapping derivable.

Repair missing is-a relation (a,b) with a $\in \mathcal{O}_k$ and b $\in \mathcal{O}_k$:
Choose an element from GenerateRepairingActions(a, b, KB_k);

Repair missing mapping (a,b) with a $\in \mathcal{O}_i$ and b $\in \mathcal{O}_j$:
Choose an element from GenerateRepairingActions(a, b, KB_{ij});

GenerateRepairingActions(a, b, KB):
1. $Source(a,b)$:= super-concepts(a) $-$ super-concepts(b) in KB;
2. $Target(a,b)$:= sub-concepts(b) $-$ sub-concepts(a) in KB;
3. $Repair(a,b)$:= $Source(a,b) \times Target(a,b)$;
4. For each $(s,t) \in Source(a,b) \times Target(a,b)$:
   if $(s,t) \in \mathcal{WI} \cup \mathcal{WM} \cup \mathcal{R}_I^- \cup \mathcal{R}_M^-$ then remove (s,t) from $Repair(a,b)$;
   else if $\exists (u,v) \in \mathcal{WI} \cup \mathcal{WM} \cup \mathcal{R}_I^- \cup \mathcal{R}_M^-$: (s,t) is more informative than (u,v) in KB
      and $u \rightarrow s$ and $t \rightarrow v$ are derivable from validated to be correct only is-a relations and/or mappings
   then remove (s,t) from $Repair(a,b)$;
5. return $Repair(a,b)$;

Fig. 4. Algorithm for generating repairing actions for missing is-a relations and mappings

In the network in Figure 1 *(nasal bone, bone)* in AMA is validated as missing. After adding the missing is-a relations to the ontology, its Source set is {*nasal bone, viscerocranium bone*} and its Target set is {*bone, limb bone, forelimb bone, hindlimb bone, foot bone, metatarsal bone, tarsal bone, jaw, maxilla, lacrimal bone*}, i.e., Repair contains $2 \times 10 = 20$ possible repairing actions.

Phase 3.2: Rank Wrong and Missing Is-a Relations and Mappings. In general, there will be many is-a relations/mappings that need to be repaired and some of them may be easier to start with such as the ones with fewer repairing actions. We therefore rank them with respect to the number of possible repairing actions.

Phase 3.3: Recommend Repairing Actions. The recommendation algorithm for wrong is-a relations/mappings assigns a priority to each possible repairing action based on how often it occurs in the justifications and its importance in already repaired is-a relations and mappings. For a missing is-a relation/mapping (a,b) (as defined in [19]) it computes the most informative repairing actions from $Source(a,b) \times Target(a,b)$ that are supported by external domain knowledge (WordNet and UMLS).

Phase 3.4: Execute Repairing Actions. Depending on whether a wrong or missing is-a relation/mapping is repaired the chosen repairing actions are removed from or added to the relevant ontologies and alignments. The current sets of wrong ($\mathcal{WI}$/$\mathcal{WM}$) and missing ($\mathcal{MI}$/$\mathcal{MM}$) is-a relations and mappings need to be updated since one repairing action can repair more than one is-a relation/mapping or previously repaired relations/mappings may need to be repaired again. The sets of repairing actions for wrong ($\mathcal{R}_I^-$, $\mathcal{R}_M^-$) and missing ($\mathcal{R}_I^+$, $\mathcal{R}_M^+$) is-a relations/mappings need to be updated as well. Further, new CMIs and CMMs may appear. In other cases the possible repairing actions for wrong and missing is-a relations and mappings may change (update justifications and sets of possible repairing actions for missing is-a relations and mappings). We also need to update the KBs.

5 Alignment Component

The input for this component consists of two taxonomies. The output is an alignment.

Phase 1: Detect Candidate Missing Mappings. In ontology alignment mapping suggestions are generated which essentially are CMMs. While the generation of CMMs in the debugging component is a specific kind of ontology alignment using the knowledge inherent in the network, in the alignment component we use other types of alignment algorithms. Matchers are used to compute similarity values between concepts in different ontologies. The results of the matchers can be combined and filtered in different ways to obtain mapping suggestions. In our approach we have currently used the linguistic, WordNet-based and UMLS-based algorithms from the SAMBO system [21]. The matcher *n-gram* computes a similarity based on 3-grams. The matcher *TermBasic* uses a combination of n-gram, edit distance and an algorithm that compares the lists of words of which the terms are composed. The matcher *TermWN* extends TermBasic by using WordNet for looking up is-a relations. The matcher *UMLSM* uses the domain knowledge in UMLS to obtain similarity values. The results of the matchers can be combined using a weighted-sum approach in which each matcher is given a weight and the final similarity value between a pair of concepts is the weighted sum of the similarity values divided by the sum of the weights of the used matchers. Further, we use a threshold for filtering. A pair of concepts is a mapping suggestion if the similarity value is equal to or higher than a given threshold value.

We note that in the alignment component the search space is not restricted to the *mapped concepts* only - similarity values are calculated for all pairs of concepts. KBs are initialized, in the same way as in the debugging component, for the taxonomy network and the pairs of taxonomies and their alignments. We also note that no initial alignment is needed for this component. Therefore, if alignments do not exist in the network (at all or between specific ontologies) this component may be used before starting debugging.

Phase 2: Validate Candidate Missing Mappings. The CMMs (mapping suggestions) are presented to a domain expert for validation, which is performed in the same way as in the debugging component. The domain expert can use the recommendation algorithms during the validation as well. As before, the CMMs are partitioned into two sets - wrong mappings and missing mappings. The wrong mappings are not repaired since they are not in the alignments. However, we store this information in order to avoid recomputations and for conflict checking/prevention. The concepts in the missing mappings are added to the set of *mapped concepts* (if they are not already there), and they will be used the next time CMMs/CMIs are derived in the debugging component.

Phase 3: Repairing Missing Mappings. As mentioned, we only need to repair the missing mappings. Initially, the missing mappings are added to the KBs in the same way as in the debugging component and then we try to repair them using more informative repairing actions. For repairing a missing mapping the same algorithms as in the debugging component are used to generate the Source and Target sets and the repairing process continues with the same actions described for the debugging workflow. In Phase 3.4 the repairing actions are executed analogically to those in the debugging

component and their consequences are computed. Further, the concepts in the repairing actions are added to the set of *mapped concepts* (if not there yet).

6 Interaction between the Components

The alignment component generates CMMs that are validated in the same way as in the debugging component. The CMMs validated to be correct often are missing mappings that are not found by the debugging component. Further, they may lead to new mapped concepts that are used in the debugging component. The CMMs validated to be wrong are used to avoid unnecessary recomputations and validations.

The debugging component repairs the is-a structure and the mappings. This can be used by the alignment component. For instance, the performance of structure-based matchers (e.g., [21]) and partial-alignment-based preprocessing and filtering methods [17] heavily depends on the correctness and completeness of the is-a structure.

We also note that the different phases in the components can be interleaved. This allows for an iterative and modular approach, where, for instance, some parts of the ontologies can be fully debugged and aligned before proceeding to other parts.

7 Experiments

We performed three experiments to demonstrate the benefits of the integrated ontology alignment and debugging framework. As input for Experiment 1 and 2 we used the two ontologies from the Anatomy track of OAEI 2011 - AMA contains 2,737 concepts and 1,807 asserted is-a relations, and NCI-A contains 3,298 concepts and 3,761 asserted is-a relations. The input for the last experiment contained the reference alignment (1516 equivalence mappings between AMA and NCI-A) together with the two ontologies. The reference alignment was used indirectly as external knowledge during the validation phase in the first two experiments. The experiments were performed on an Intel Core i7-2620M Processor 2.7GHz with 4 GB memory underWindows 7 Professional operating system and Java 1.7 compiler. The first author performed the validation in the experiments with help of two domain experts.

Experiment 1 - Aligning and Debugging OAEI Anatomy. The first experiment demonstrates a complete debugging and aligning session where the input is a set with the two ontologies. After loading the ontologies mapping suggestions were computed using matchers TermWN and UMLSM, weight 1 for both and threshold 0.5. This resulted in 1384 mapping suggestions. The 1233 mapping suggestions that are also in the reference alignment were validated as missing equivalence mappings (although, as we will see, there are defects in the reference alignment) and repaired by adding them to the alignment. The others were validated manually and resulted in missing mappings (53 equivalence and 39 is-a) and wrong mappings (59 equivalence and 39 is-a). These missing mappings were repaired by adding 53 equivalence and 28 is-a mappings (5 of them more informative) and 5 is-a relations (3 to AMA and 2 to NCI-A). 6 of these missing mappings were repaired by repairing others. Among the wrong mappings there were 3 which were derivable in the network. These were repaired by removing 2 is-a relations from NCI-A. Figure 5 - part A summarizes the results.

part A	candidate missing mappings	missing $\equiv$/$\leftarrow$ or $\rightarrow$	wrong $\equiv$/$\leftarrow$ or $\rightarrow$	repair missing $\equiv$/$\leftarrow$/$\rightarrow$/derivable /more informative	repair missing is-relations
Alignment	1384	1286/39	59/39	1286/20/8/6/5	-
AMA	-	-	-	-	3
NCI-A	-	-	-	-	2

part B	candidate missing all/non-redundant	missing	wrong	repair missing self/more informative/other	repair wrong removed
AMA	410/263	224	39	144/57/23	30
NCI-A	355/183	166	17	127/13/26	17
Alignment	-	-	-	-	8 $\equiv$ and 1 $\rightarrow$

Fig. 5. Experiment 1 results: A - debugging of the alignment; B - debugging of the ontologies

The generated alignment was then used in the debugging of the network created by the ontologies and the alignment. Two iterations of the debugging workflow were performed, since the repairing of wrong and missing is-a relations in the first iteration led to the detection of new CMIs which had to be validated and repaired. Over 90% of the CMIs for both ontologies were detected during the first iteration, the detection of CMIs took less than 30 seconds per ontology. Figure 5 - part B summarizes the results.

The system detected 410 (263 non-redundant) CMIs for AMA and 355 (183 non-redundant) CMIs for NCI-A. The non-redundant CMIs were displayed in groups, 45 groups for AMA and 31 for NCI-A. Among the 263 non-redundant CMIs in AMA 224 were validated as missing and 39 as wrong. In NCI-A 166 were validated as missing and 17 as wrong. The 39 wrong is-a relations in AMA were repaired by removing 30 is-a relations from NCI-A, and 8 equivalence and 1 is-a mapping from the alignment. The 17 wrong is-a relations in NCI-A were repaired by removing 17 is-a relations in AMA. The missing is-a relations in AMA were repaired by adding 201 is-a relations - in 144 cases the missing is-a relation itself and in 57 cases a more informative is-a relation. 23 of the 224 missing is-a relations became derivable after repairing some of the others. To repair the missing is-a relations in NCI-A 140 is-a relations were added - in 127 cases the missing is-a relation itself and in 13 cases a more informative is-a relation. 26 out of the 166 missing is-a relations were repaired while other is-a relations were repaired.

We observe that for 57 missing is-a relations in AMA and 13 in NCI-A the repairing actions are more informative than the missing is-a relation itself, i.e., for each of these, knowledge, which was not derivable from the network before, was added to the network. Thus the knowledge represented by the ontologies and the network has increased.

Experiment 2. For this experiment the alignment process was run twice and at the end the alignments were compared. The same matchers, weights and threshold as in Experiment 1 were used. During both runs the CMMs (mapping suggestions) were computed and validated in the same manner. This step is as in Experiment 1 and the results are the ones in Figure 5 - part A. The difference between both runs is in the repairing phase. In the first run the missing mappings were repaired by directly adding them to the final alignment without benefiting from the repairing algorithms - in the same way most of

the alignment systems do. The final alignment contained 1286 equivalence and 39 is-a[5] mappings.

During the repairing phase in the second run the debugging component was used to provide alternative repairing actions than those available in the initial set of mapping suggestions. The final alignment then contained 1286 equivalence mappings from the mapping suggestions, 28 is-a mappings from the mapping suggestions where 5 of them are more informative, thus adding knowledge to the network. Further, 5 mapping suggestions were repaired adding is-a relations (3 in AMA and 2 in NCI-A) and thus adding more knowledge to each of the ontologies. 6 more mapping suggestions became derivable from the network as a result from the repairing actions for other CMMs.

Experiment 3. In this experiment the debugging process was run twice, CMIs were detected for both ontologies and compared between the runs. The input for the first run was the set of the two ontologies and their alignment from the Anatomy track in OAEI 2011. The network was loaded in the system and the CMIs were detected. 496 CMIs were detected for AMA, of which 280 were non-redundant. For NCI-A 365 CMIs were detected of which 193 were non-redundant. The same input was used in the second run. However, the alignment algorithms were used to extend the set with mappings prior to generating the CMIs. The set-up for the aligning was the same as in Experiment 1 and the mapping suggestions were computed, validated and repaired in the same way as well. Then CMIs were generated - 638 CMIs were detected for AMA (357 non-redundant), and 460 CMIs for NCI-A (234 non-redundant). In total 145 new CMIs were detected for AMA - 120 were validated as missing and 25 validated as wrong[6]. 103 new CMIs were detected for NCI-A - 53 were validated as missing and 50 as wrong.

Discussion. Experiment 1 shows the usefulness of the system through a complete session where an alignment was generated and many defects in the ontologies were repaired. Some of the repairs added new knowledge. As a side effect, we have shown that the ontologies that are used by the OAEI contain over 200 and 150 missing is-a relations, respectively and 39 and 17 wrong is-a relations, respectively. We have also shown that the alignment is not complete and contains wrong information. We also note that our system allows validation and allows a domain expert to distinguish between equivalence and is-a mappings. Most ontology alignment systems do not support this.

Experiment 2 shows the advantages for ontology alignment when also a debugging component is added. The debugging component allowed to add more informative mappings, reduce redundancy in the alignment as well as debug the ontologies leading to further reduced redundancy in the alignment. For the ontologies and alignment new knowledge not found when only aligning, was added. In general, the quality of the final alignment (and the ontologies) becomes higher.

Experiment 3 shows that the debugging process can take advantage of the alignment component even when an alignment is available. The alignment algorithms can provide additional mapping suggestions and thus extending the alignment. More mappings be-

[5] 5 of these are repaired in the second run by adding is-a relations in the ontologies.

[6] The sum of the newly generated CMIs and those in the first run is not equal to the number of the CMIs in the second run because some of the CMIs generated in the first run are derivable in the second run.

tween two ontologies means higher coverage and possibly more detected and repaired defects. In the experiment more than 100 CMIs (of which many correct) were detected for each ontology using the extended set of mappings. We also note that the initial alignment contained many mappings (1516). In the case that the alignment contains fewer mappings the benefit to the debugging process will be even more significant.

8 Related Work

To our knowledge there is no other system that integrates ontology debugging and ontology alignment in a uniform way and that allows for a strong interleaving of these tasks. There are some ontology alignment systems that do semantic verification and disallow mappings that lead to unsatisfiable concepts (e.g., [10,12]). Further, adding missing is-a relations to ontologies was a step in the alignment process in [17].

Regarding the debugging component, this work extends the work in [19,18] that dealt with debugging is-a structure in taxonomy networks. These were one of the few approaches dealing with repairing missing is-a structure and in the case of [18] debugging both missing and wrong is-a structure. The current work extends this by also including debugging of mappings in a uniform way as well as ontology alignment. The ontology alignment component also removed the restriction of [18] that required the existence of an initial alignment.

There are different ways to *detect* missing is-a relations. One way is by inspection of the ontologies by domain experts. Another way is to use external knowledge sources. For instance, there is much work on finding relationships between terms in the ontology learning area [1]. Regarding the detection of is-a relations, one paradigm is based on linguistics using lexico-syntactic patterns. The pioneering research conducted in this line is in [9], which defines a set of patterns indicating is-a relationships between words in the text. Another paradigm is based on machine learning and statistical methods. Further, guidelines based on logical patterns can be used [3]. These approaches are complementary to the approach used in this paper. There is, however, not much work on the *repairing* of missing is-a relations that goes beyond adding them to the ontologies except for [19] for taxonomies and [16] for $\mathcal{ALC}$ acyclic terminologies.

There is more work on the debugging of semantic defects. Most of it aims at identifying and removing logical contradictions from an ontology. Standard reasoners are used to identify the existence of a contradiction, and provide support for resolving and eliminating it [6]. In [24] minimal sets of axioms are identified which need to be removed to render an ontology coherent. In [15,14] strategies are described for repairing unsatisfiable concepts detected by reasoners, explanation of errors, ranking erroneous axioms, and generating repair plans. In [8] the focus is on maintaining the consistency as the ontology evolves through a formalization of the semantics of change for ontologies. [26] introduces a method for interactive ontology debugging. In [22] and [11] the setting is extended to repairing ontologies connected by mappings. In this case, semantic defects may be introduced by integrating ontologies. Both works assume that ontologies are more reliable than the mappings and try to remove some of the mappings to restore consistency. The solutions are often based on the computation of minimal unsatisfiability-preserving sets or minimal conflict sets. The work in [23] further characterizes the problem as mapping revision. Using belief revision theory, the authors

give an analysis for the logical properties of the revision algorithms. Another approach for debugging mappings is proposed in [28] where the authors focus on the detection of certain kinds of defects and redundancy. The approach in [13] deals with the inconsistencies introduced by the integration of ontologies, and unintended entailments validated by the user.

Regarding the alignment component there are some systems that allow validation of mappings such as SAMBO [21], COGZ [5] for PROMPT, and COMA++ [4]. [7] introduces an efficient algorithm for computing a minimal set with mappings which could reduce user interaction. Many matchers have been proposed (e.g., many papers on http://ontologymatching.org/), and most systems use similar combination and filtering strategies as in this paper. For an overview we refer to [25].

9 Conclusion

In this paper we presented a unified approach for aligning taxonomies and debugging taxonomies and their alignments. This is the first approach which integrates ontology alignment and ontology debugging and allows debugging of both the structure of the ontologies as well as their alignments. Further, we have shown the benefits of our approach through experiments. The interactions between ontology alignment and debugging significantly raise the quality of both taxonomies and their alignments. The ontology alignment provides or extends alignments that are used by the debugging. The debugging provides algorithms for repairing defects in alignments and possibly add new knowledge.

We will continue exploring the interactions between ontology alignment and debugging. We will include and investigate the benefits when using structure-based alignment algorithms and partial-alignment-based techniques. Further, we will investigate the debugging problem for ontologies represented in more expressive formalisms.

Acknowledgements. We thank the Swedish Research Council (Vetenskapsrådet) and the Swedish e-Science Research Centre (SeRC) for financial support.

References

1. Cimiano, P., Buitelaar, P., Magnini, B.: Ontology Learning from Text: Methods, Evaluation and Applications. IOS Press (2005)
2. Conroy, C., Brennan, R., O'Sullivan, D., Lewis, D.: User Evaluation Study of a Tagging Approach to Semantic Mapping. In: Aroyo, L., et al. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 623–637. Springer, Heidelberg (2009)
3. Corcho, O., Roussey, C., Vilches, L.M., Pérez, I.: Pattern-based OWL ontology debugging guidelines. In: Workshop on Ontology Patterns, pp. 68–82 (2009)
4. Do, H.-H., Rahm, E.: Matching large schemas: approaches and evaluation. Information Systems 32, 857–885 (2007)
5. Falconer, S.M., Storey, M.-A.: A cognitive support framework for ontology mapping. In: Aberer, K., et al. (eds.) ISWC/ASWC 2007. LNCS, vol. 4825, pp. 114–127. Springer, Heidelberg (2007)
6. Flouris, G., Manakanatas, D., Kondylakis, H., Plexousakis, D., Antoniou, G.: Ontology change: Classification and survey. Knowledge Engineering Review 23(2), 117–152 (2008)

7. Giunchiglia, F., Maltese, V., Autayeu, A.: Computing minimal mappings. In: Ontology Matching Workshop, pp. 37–48 (2009)
8. Haase, P., Stojanovic, L.: Consistent Evolution of OWL Ontologies. In: Gómez-Pérez, A., Euzenat, J. (eds.) ESWC 2005. LNCS, vol. 3532, pp. 182–197. Springer, Heidelberg (2005)
9. Hearst, M.: Automatic acquisition of hyponyms from large text corpora. In: 14th International Conference on Computational Linguistics, pp. 539–545 (1992)
10. Jean-Mary, Y.R., Shironoshita, E.P., Kabuka, M.R.: Ontology matching with semantic verification. Journal of Web Semantics 7(3), 235–251 (2009)
11. Ji, Q., Haase, P., Qi, G., Hitzler, P., Stadtmüller, S.: RaDON — repair and diagnosis in ontology networks. In: Aroyo, L., et al. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 863–867. Springer, Heidelberg (2009)
12. Jimenez-Ruiz, E., Cuenca-Grau, B., Zhou, Y., Horrocks, I.: Large-scale interactive ontology matching: Algorithms and implementation. In: 20th European Conference on Artificial Intelligence, pp. 444–449 (2012)
13. Jiménez-Ruiz, E., Cuenca Grau, B., Horrocks, I., Berlanga, R.: Ontology integration using mappings: Towards getting the right logical consequences. In: Aroyo, L., et al. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 173–187. Springer, Heidelberg (2009)
14. Kalyanpur, A., Parsia, B., Sirin, E., Cuenca-Grau, B.: Repairing Unsatisfiable Concepts in OWL Ontologies. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 170–184. Springer, Heidelberg (2006)
15. Kalyanpur, A., Parsia, B., Sirin, E., Hendler, J.: Debugging Unsatisfiable Classes in OWL Ontologies. Journal of Web Semantics 3(4), 268–293 (2006)
16. Lambrix, P., Dragisic, Z., Ivanova, V.: Get my pizza right: Repairing missing is-a relations in ALC ontologies. In: 2nd Joint International Semantic Technology Conference (2012)
17. Lambrix, P., Liu, Q.: Using partial reference alignments to align ontologies. In: Aroyo, L., et al. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 188–202. Springer, Heidelberg (2009)
18. Lambrix, P., Liu, Q.: Debugging is-a structure in networked taxonomies. In: 4th International Workshop on Semantic Web Applications and Tools for Life Sciences, pp. 58–65 (2011)
19. Lambrix, P., Liu, Q., Tan, H.: Repairing the missing is-a structure of ontologies. In: Gómez-Pérez, A., Yu, Y., Ding, Y. (eds.) ASWC 2009. LNCS, vol. 5926, pp. 76–90. Springer, Heidelberg (2009)
20. Lambrix, P., Qi, G., Horridge, M.: Proceedings of the 1st International Workshop on Debugging Ontologies and Ontology Mappings. LiU E-Press, LECP 79 (2012)
21. Lambrix, P., Tan, H.: SAMBO - a system for aligning and merging biomedical ontologies. Journal of Web Semantics 4(3), 196–206 (2006)
22. Meilicke, C., Stuckenschmidt, H., Tamilin, A.: Repairing Ontology Mappings. In: 22nd Conference on Artificial Intelligence, pp. 1408–1413 (2007)
23. Qi, G., Ji, Q., Haase, P.: A Conflict-Based Operator for Mapping Revision. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 521–536. Springer, Heidelberg (2009)
24. Schlobach, S.: Debugging and Semantic Clarification by Pinpointing. In: Gómez-Pérez, A., Euzenat, J. (eds.) ESWC 2005. LNCS, vol. 3532, pp. 226–240. Springer, Heidelberg (2005)
25. Schvaiko, P., Euzenat, J.: Ontology matching: state of the art and future challenges. IEEE Transactions on Knowledge and Data Engineering 25(1), 158–176 (2013)
26. Shchekotykhin, K., Friedrich, G., Fleiss, P., Rodler, P.: Interactive ontology debugging: Two query strategies for efficient fault localization. Journal of Web Semantics 12-13, 88–103 (2012)
27. UMLS. Unified medical language system, `http://www.nlm.nih.gov/research/umls/about_umls.html`
28. Wang, P., Xu, B.: Debugging ontology mappings: a static approach. Computing and Informatics 27, 21–36 (2008)
29. WordNet, `http://wordnet.princeton.edu/`

Opening the Black Box of Ontology Matching

DuyHoa Ngo, Zohra Bellahsene, and Konstantin Todorov

Université Montpellier 2, LIRMM,
161 rue Ada, 34095, Montpellier, France
firstname.lastname@lirmm.fr

Abstract. Due to the high heterogeneity of ontologies, a combination of many methods is necessary in order to discover correctly the semantic correspondences between their elements. An ontology matching tool can be seen as a collection of several matching components, each implementing a specific method dealing with a specific heterogeneity type (terminological, structural or semantic). In addition, a mapping selection module is introduced to filter out the most likely mapping candidates. This paper proposes an empirical study of the interaction between these components working together inside an ontology matching system. By the help of datasets from the Ontology Alignment Evaluation Initiative, we have carried out several experimental studies. In the first place, we have been interested in the impact of the mapping selection module on the performance of terminological and structural matchers revealing the advantage of using global methods vs. local ones. Further, we have carried an extensive study on the flaw of the performance of a structural matcher in the presence of noisy input coming from a terminological method. Finally, we have analyzed the behavior of a structural and a semantic component with respect to inputs taken from different terminological matchers.

1 Introduction

The field of ontology matching has matured considerably as a result of more than a decade of research and practice. Many ontology matching approaches and systems have been developed dealing with the semantic heterogeneity problem by taking into account various aspects of this problem [20]. Methodologically speaking, these approaches rely on techniques from fields as diverse as machine learning, graph matching, information retrieval, relational algebra, logics, – each of these fields providing a framework to deal with a certain heterogeneity type. In this respect, a standalone ontology matching system is a successful combination of several matching components. We consider that time has come to pay attention to the way these components connect to each other within a matching system and how these interactions impact the overall quality of the produced alignments.

Many challenges stand in front of the ontology matching community – a full picture is given in [20]. By this study, we contribute to the solution of matcher selection and combination problems, which are fundamental for the development of a stable system. A matching system can be seen as a combination of four main components: a terminological, a structure-based and a semantics-based matcher accompanied by a mapping selection module[1]. Although these components exploit different features of the entities

[1] Not each of these components is always and necessarily part of the system's architecture.

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 16–30, 2013.

of an ontology, they are not independent. A structure-based matcher takes as an input the mappings resulting from a terminological matcher [1,7,24]; a semantics-based matcher may take as an input the mappings resulting from either a terminological [5,9], or a structure-based matcher, or a combination of the mappings resulting from both [4,6]. Therefore, difficulties can arise not only inside each component but also on the interaction lines between them. We take into consideration several of these difficulties.

A ***mapping selection*** module is usually introduced in order to filter out the best mapping candidates, at each of the different matching levels. The interaction of this module with the matchers is, therefore, among the basic issues to be addressed.

A ***terminological matcher*** discovers mappings by comparing annotations (i.e., labels, comments) of entities. To this end, it may use many different similarity measures. The difficulty is, on the one hand how to select the most appropriate similarity measures and, on the other hand, how to effectively combine them.

A ***structure-based matcher*** discovers mappings between entities by analyzing the similarity of the structural patterns, which these entities are part of. However, according to [3], almost all methods of this type are not stable and do not improve the matching quality when the structures of the ontologies are different. Moreover, structural matchers are error-prone, since they strongly depend on initial mappings provided by a terminological matcher and on the specific settings of the mapping selection component.

A ***semantic matcher*** is mainly used to refine candidate mappings [5,6,9]. It exploits the semantic constraints between entities in the ontologies in order to discover conflicts between potential mappings and remove them from the list of candidate mappings. To do that, in some tools [5,9], the semantic module requires a confidence value for each mapping candidate. Then, it applies a global optimization method in order to find the minimal inconsistent set of mappings. Therefore, similarly to structural methods, semantic methods are error-prone because they also depend on the confidence values of the mappings obtained at previous steps.

This empirical study aims to investigate the interconnections between the different components in an ontology matching system. Our intention has been to make explicit the relations between these components by showing how one impacts the other and thus guide practitioners and researchers in the choice of the matchers with regard to the global quality of the matching system.

The rest of the paper is organized in the following manner. In the next section, we present a generic ontology matching system architecture together with an evaluation scenario for the ontology matching task. We continue by presenting two basic studies. With respect to different settings of the mapping selection module, we evaluate the performance of different terminological methods (Sections 3) and different structural methods (Section 4). Further, we go into a detailed study of the interaction between terminological, structural and semantic methods. We first study the performance of different structural matchers at the presence of noisy input (Section 5) and then the behavior of structural and semantic matchers with respect to mappings produced by different terminological methods that they take as an input (Section 6). Sections 3 to 6 present one independent study each and are structured in an uniform manner: the matching methods are presented first, followed by a presentation of the evaluation data and strategy and, finally, the results are given and supported by an in-depth discussion.

2 A Generic Framework for Ontology Matching and Evaluation

The main components of a standalone ontology matching system are depicted in the lower part of Fig. 1. As discussed in the introduction, the three core matching components are based on terminology, structure and semantics. The role of these matchers is to discover correct mappings or remove incorrect ones according to specific features extracted from the entities of the input ontologies. Additionally, a mapping selection module is introduced to act as a filter which selects the best candidate mappings. We define a matching strategy as the way the matcher and selection components work together in order to produce an alignment.

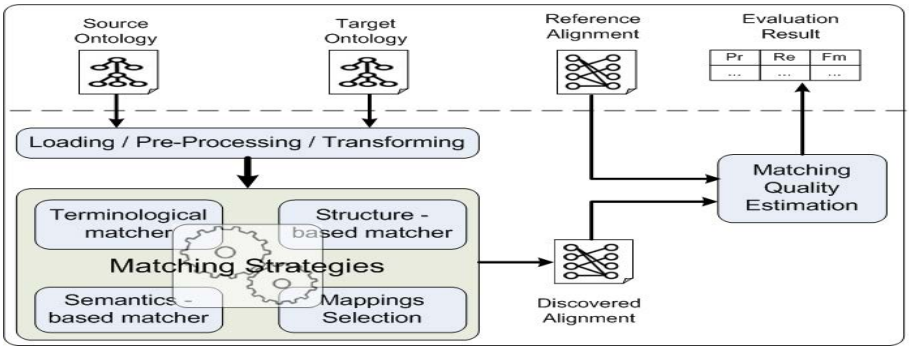


Fig. 1. Ontology Matching: System Architecture and Evaluation Scenario

To perform an evaluation of the quality of the different matching strategies, the ontology matching system requires matching scenarios as an input (upper part of Fig. 1). A matching scenario consists of a source and a target ontology and a reference alignment provided by a domain expert. Given a matching scenario, input ontologies are loaded, pre-processed and transformed into internal data structures (Loading/Pre-Processing/Transforming component). The Matching Quality Estimation module evaluates the quality of a given matching strategy by comparing discovered alignments with the reference alignment. It outputs three evaluation values corresponding to Precision (Pr), Recall (Re) and F-measure (Fm). In this study, we compute the harmonic means of precision, recall and F-measures on a set of n tests. These evaluation measures are used in the OAEI campaign and are given as follows.

$$H(p) = \frac{\sum_{i=1}^{n} |C_i|}{\sum_{i=1}^{n} |A_i|}; \quad H(r) = \frac{\sum_{i=1}^{n} |C_i|}{\sum_{i=1}^{n} |R_i|}; \quad H(fm) = \frac{2 * H(p) * H(r)}{H(p) + H(r)}.$$

For the *i*th test, $|A_i|$ denotes the total number of mappings discovered by a matching system, $|C_i|$ – the number of correct mappings, and $|R_i|$ – the number of reference mappings provided by a domain expert. In the sequel, all results will be given by considering F-measures only.

By following this generic architecture, we have developed the YAM++ system[2]. Various matching methods inside of the three matcher components and several filtering

[2] YAM++ - (not) Yet Another Matcher, published here: `http://www2.lirmm.fr/~dngo/`.

methods used in the mapping selection module have been implemented. The system is described in detail in [17]. Because of the broad scope and diversity of the techniques employed by YAM++, as well as its excellent results in the OAEI campaigns[3], we have used this system in order to evaluate the different matching strategies based on the interaction of the matchers. More detail about the setup of the matching and filtering methods for each matching strategy will be given in each experiment in the succeeding sections.

Note that required computation times for each technique have not been taken into account in this study. They could, however, be a useful decision factor when two techniques produce similar results in terms of precision and recall.

3 Terminological Matchers and Mapping Selection

In this evaluation, we focus on terminological matchers and we study their interaction with the mapping selection module. According to a classification found in [3], the terminological matching approaches are divided into ***local*** and ***global*** methods. Local methods focus on the similarity between individual entities, whereas global methods combine local ones, taking into account the semantic context that these entities belong to. We are particularly interested in the comparison between local and global methods.

3.1 Methods

We have considered several state-of-the-art local methods as well as advanced global methods, some of which have been proposed originally for YAM++.

Local Terminology-Based Methods. We have implemented more than fifty local methods used for terminology-based matching [16]. We divide them into three groups based on the algorithm for computing similarity between strings that they rely on. To economize space, the following representative methods will be used in this experiment:

Edit Distance-Based Methods. The similarity of two strings is computed based on the number of edit operations needed to transform one string to another. We have considered Levenstein and ISUB [21].

Token-Based Methods. These methods split strings into sets of tokens and then compare tokens by string-based methods. We have considered QGrams and TokLev (using Levestein to compare tokens).

Hybrid Methods. Methods in this group split strings into sets of tokens and then compare tokens by combining string-based and linguistic-based methods. We have taken as examples HybLinISUB and HybJCLev. HybLinISUB uses a combination of ISUB and Lin [11]; HybJCLev relies on the Levenstein and the Jiang-Corath [8] methods.

Global Terminology-Based Methods. In our experiments, we have implemented the following global methods:

[3] In OAEI 2012 YAM++ was first on the Conference, Multifarm, Benchmark and Bio-Medical track, and second on the Anatomy track.

Weighted Average with Local Confidence (LC). Each local method is assigned a local confidence value. These values are used as weights in a weighted average function to compute the final similarity score between entities. More details can be found in [1].

Harmony-based Adaptive Similarity Aggregation (HADAPT). Here, each local method is assigned a weight which is computed by the harmony estimation algorithm [12]. Then, a weighted sum aggregation method is used to produce a final similarity score between entities.

Machine Learning-Based Approach (ML). This method combines all local methods and constructs a classification function on the basis of given training data. In a machine learning setting, the training dataset consists of pairs of entities (seen as training examples) for which the confidence value of their similarity is known. Based on these training examples, a classifier learns a function which is able to predict the confidence value of an unseen pair of entities. In that way, the ontology matching task is transformed into a classification task. After testing the performance of over 15 machine learning techniques, we have seen that J48 decision tree is the most appropriate one for the ontology matching task [17]. This is the method that has been used in the following experiments.

Information Retrieval-Based Approach (IR). This method judges the similarity between two entities by the amount of overlap of the information content of their labels [18]. It splits all labels of entities into tokens and calculates the information content of each token in the whole ontology. Then, IR extends Tversky's similarity measure [23] with weight of tokens to compute a similarity score between labels of entities. The method compares similarity of two labels by using not only the sequence of characters themselves, but also their information content in an ontology. We will illustrate this idea by examples in this experiment.

3.2 Matching and Evaluation Strategy

To perform this experiment, we have chosen the Conference dataset from the OAEI including 21 test cases[4]. The reason for this choice is that this dataset consists of moderate-sized real-world ontologies describing the same domain. These ontologies are highly heterogeneous since they were developed by different people, hence, the same concept is often labeled differently. We assume that high matching quality of a system on these tests guarantees similar results of the system when applied to other real matching scenarios.

The matching evaluation strategy works as follows. For each matching method (including local and global ones) in the terminological matching module, we compute a similarity score for all pairs of entities of the input ontologies. The mapping selection module then selects candidate mappings according to the filter threshold value. At each level of this threshold, a H-mean F-measure is computed over all test cases in the dataset.

3.3 Results and Discussion

Fig. 2 shows the results of this comparison. As can be seen, almost all local methods (except for QGrams and ISUB) improve the F-measure when the threshold value of the

[4] `http://oaei.ontologymatching.org/2012/conference/index.html`

filter increases. When the filter threshold increases, it appears that only highly similar or identical labels are be passed. Therefore, as Fig. 2 shows, local methods closely converge to results of the Identical method (≈ 0.55) when the filter threshold reaches 0.95 and 0.97. The same trend is observed for HADAPT and LC because they are linear combinations of local methods.

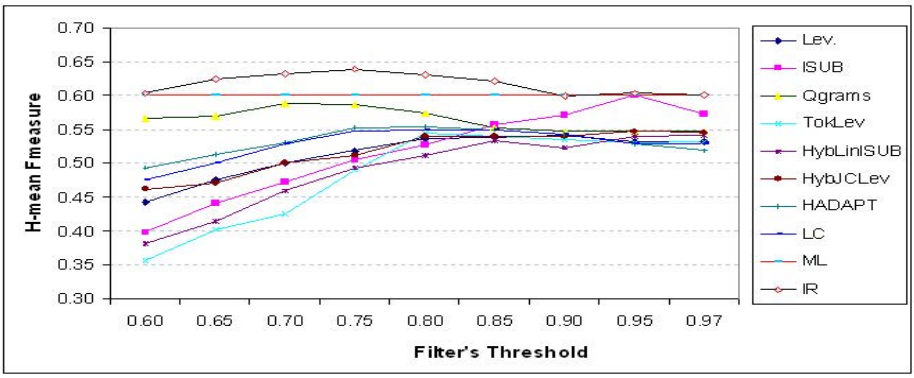


Fig. 2. Mapping Selection for the Terminological Matcher Module

The experiment shows that the two global methods, ML and IR outperform the other techniques within the terminological matcher module. Therefore, in what follows, we will discuss in more detail these two methods.

Performance of ML. A machine learning method requires training data on which to learn a classification function and test data on which to apply this function. To create training data independent on the Conference dataset on which the evaluation will be performed, we have used data from the OAEI Benchmark 2009[5] and I3CON[6] datasets. We have constructed 10 different training datasets by using these two sources and we have trained the decision tree ML method on each of these 10 training sets. At each time, the learned classification algorithm has been applied to the Conference dataset, providing 10 different results. The result given in Fig. 2 is obtained by taking the average over these 10 results.

We note that the ML method does not depend on the filter threshold since no candidate mapping selection takes place. As it is seen from the figure, ML returns a better matching quality than LC, HADAPT and all local methods. For example, the ML method discovers (cmt.owl#Co-author $\equiv$ conference.owl#Contribution_co-author) in the `cmt.owl` and `conference.owl` ontologies, whereas local methods return a low similarity score between these labels (Levenstein(Co-author,Contribution_co-author) = 0.4; QGrams(Co-author,Contribution_co-author) = 0.6). This is explained by the fact that ML does not use arithmetic combination functions like LC and HADAPT, instead, it extracts the combination rules on local methods from training data. ML is able to find many patterns in the training data similar to the current example

[5] `http://oaei.ontologymatching.org/2009/benchmarks`
[6] `http://www.atl.external.lmco.com/projects/ontology/i3con.html`

(e.g., (networkA.rdf#Office ≡ networkB#OfficeSoftware), (russia1#payment ≡ russia2# means_of_payment), etc.). However, the ML method strongly depends on the training data. With different training data, different machine learning models will be generated and, therefore different matching results will be produced. For instance, with some training data, ML can discover (cmt.owl#Co-author ≡ conference.owl#Contribution_co-author), but not with other. Moreover, for a given training data this mapping is discovered by ML, but (cmt.owl#Document ≡ conference.owl#Conference_document) is not, even though the latter seems similar to the former. To address this problem, we have designed the IR method, which is discussed in the sequel.

Performance of IR. The IR method proposed in YAM++ [18] outperforms all other methods in the experiment. We analyze this fact by giving an example with two entities: `cmt.owl`#Co-author and `conference.owl`#Contribution_co-author. After splitting and normalizing the labels, we have the following two sets of tokens: {`coauthor`} and {`coauthor`, `contribution`}. Token `coauthor` appears in each input ontology only once, whereas token `contribution` appears 10 times among 60 concepts in the `conference.owl` ontology. Therefore, the information content of the token `contribution` is lower than that of token `coauthor`. In particular, the normalized *tf-idf* weights of each token inside the input ontologies are equal: $\{w_{coauthor} = 1.0\}$, $\{w_{coauthor} = 1.0, w_{contribution} = 0.34\}$. The two sets of tokens share only the token `coauthor`, hence the similarity computed by Tversky's method is $\frac{1.0+1.0}{1.0+1.0+0.34} = 0.855$. Similarly, we have the similarity between (`Document`,`Conference_document`) equaling `0.91`. In this pair, the token `conference` appears 15 times in the `conference.owl` ontology. Therefore, this token brings little information for this ontology and, consequently, this pair of entities represents a likely match.

It is difficult to give a clear indication which of these two best performing methods to use – ML or IR. Clearly, in the absence of training data, the choice will go for IR. Even in the presence of training data, the IR method appears to be more suitable because it reaches an overall higher performance than ML for low values of the mapping selection threshold. However, an advantage of the ML method is that it does not depend on the setting of a filter threshold. Both methods can be combined within the architecture of an ontology matching tool.

4 Structural Matchers and Mapping Selection

In this evaluation, we are interested in the behavior of structural similarity methods with respect to the mapping selection module.

4.1 Methods

The following standard structural matching methods have been considered within this study: ***ANCESTORS*** (two entities are similar if all or most of their ancestor entities are already similar), ***DESCENDANTS*** (two entities are similar if all or most of their descendant entities are already similar), ***LEAVES*** (two entities are similar if all or most of their leaf entities are already similar [2]), ***ADJACENTS*** (two entities are similar if all or most of their adjacent entities (parents, children, siblings, domains, ranges)

are already similar), ***ASCOPATH*** (two entities are similar if all or most of entities in the paths from the root to the entities in question are already similar [10]). ***DSIPATH (Descendant's Similarity Inheritance)*** (two entities are similar if the total contribution of entities in the paths from the root to them is higher than a specific threshold [22]), and ***SSC (Sibling's Similarity Contribution)*** (two entities are similar if the total contribution of their sibling entities is higher than a specific threshold [22]).

Additionally, we have considered the ***SP (Similarity Propagation)*** method. This method is proposed in the system YAM++ as an extension of the well-known similarity flooding algorithm [15]. The basic idea of the method is as follows. Assume that the entities A_1 and A_2 in one ontology are related by a directed relation P and the entities B_1 and B_2 in another ontology are related by the same directed relation. Then, if we discover that (A_1, B_1) is a match, the SP method would imply that (A_2, B_2) is a match, too. The similarity values between the two pairs are propagated to each other at each iteration of algorithm. The approach is described in detail in [19].

4.2 Matching and Evaluation Strategy

To perform this experiment, we have used the Benchmark 2011 dataset from the OAEI campaign including 103 test cases. These datasets are acquired by taking an original ontology and altering the names of some of its entities by using random strings (no variation by naming convention or synonym words). The entities whose labels have not been altered are kept as in the original ontology. Therefore, a matching scenario which takes as an input the original and the altered ontologies is appropriate for evaluating the performance of structural methods. As an input to these structural methods, we use the alignment produced by an identical metric (defined as one which returns a correct mapping only for identical strings), since the non-altered string names are identical in both ontologies. An additional characteristics of this dataset is that in some tests, not only the names of entities are altered but also the ontology structure by flattening, extension and other structure modifying operations.

The matching and evaluation strategy used in the experiment is given as follows. Only three modules will be used: a terminological matcher, a structure-based matcher and a mapping selection module.

The ***terminology-based*** matcher provides input mappings to the structural matcher. They are provided by the *identical metric*, denoted by INIT_MAPPINGS.

Each ***structure-based*** matcher corresponding to each of the selected structural measures above produces a similarity matrix for all pairs of entities from the two input ontologies.

In the ***mapping selection*** module, we vary the threshold $(0.01 – 0.9)$ to filter out the mappings discovered by this matcher. The mappings obtained by the structural matcher are combined with mappings obtained by the terminological matcher to produce the set of candidate mappings. Then, a greedy selection method [14] is used to extract the final alignment.

4.3 Results and Discussion

As can be seen from Fig. 3, when the threshold varies from 0.6 to 0.9, the structural methods converge to the INIT-MAPPINGS line where H-mean Fmeasure = 0.68 (4463 correct

mappings, 27 incorrect mappings, 4342 unfound). This means that the structural methods did not discover additional correct mappings or they discovered correct mappings, which already exist in the init mappings. This is natural, because most of the structural methods compute similarity between two entities based on the overlap of their structural patterns (i.e., adjacent, ancestor, etc.) by using, for instance, the Jaccard measure. Therefore, the higher the filter threshold, the lower the possibility of discovering new mappings.

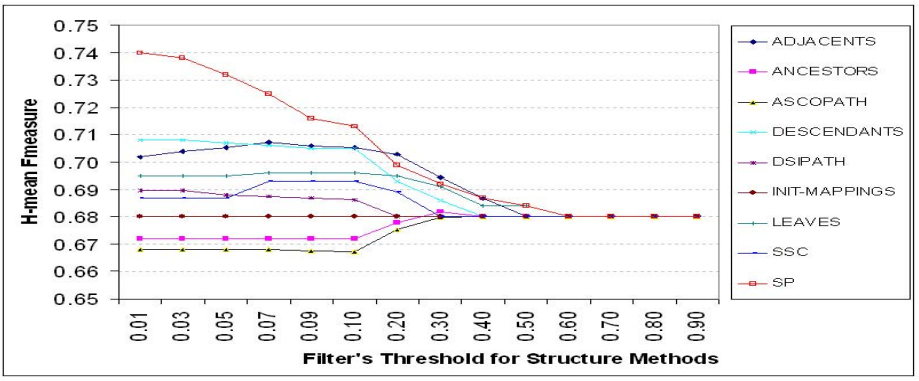


Fig. 3. Mapping Selection for Structural Methods

We note that the corresponding matching qualities of the structural methods differ significantly when the filter threshold is set to small values. We will consider methods that perform poorly and such that perform well.

For threshold values between 0.01 and 0.09, ASCOPATH and ANCESTORS discover many incorrect mappings. For example, when the threshold is equal to 0.01, ACSOPATH discovers 90 $(= 4733 - 4643)$ additional correct mappings but 453 $(= 480 - 27)$ incorrect mappings in comparison to the init mappings. This can be explained as follows. After observing the ontologies in the Benchmark 2011 dataset, we see that the maximum depth and also maximum number of ancestors of an entity in the ontology hierarchy is 5. Assume that two entities have only one common entity among their ancestors. Then their similarity score is equal to $1/10 = 0.1$ at least. If two entities do not have any common entities, then their similarity is equal to 0. Therefore, with a threshold in the range of 0.01 to 0.09, any pair of entities having at least one common ancestor will be considered as a match. Since sibling entities have the same ancestors and paths to these ancestors, they will have the same structural patterns. Therefore, many pairs of entities will have the same similarity scores. Moreover, one entity may have many descendant entities so many pairs of entities can be coupled, consequently, many incorrect mappings will be produced.

In contrast, other methods such as DESCENDANTS, LEAVES, DSIPATH and SSC provide better results with small thresholds than the methods discussed above. They discover more additional correct mappings and, consequently, improve the overall quality of the matching. For example, with a threshold equal to 0.01, DESCENDANTS discovers 494 $(= 5137 - 4643)$ additional correct mappings and 175 $(= 202 - 27)$ incorrect mappings in comparison to the init mappings. Similarly to the ASCOPATH and

ANCESTORS methods, with low threshold filter, many pairs of entities are passed. However, these methods clearly distinguish the structural patterns of entities. For instance, in DESCENDANTS and LEAVES, different entities have different sets of leaves / descendants; in DSIPATH and SSC, they use different contribution percentage of entities according to how much one entity is important to another [22]. Therefore, by running greedy selection, which always selects the pair of entities having high similarity score with 1:1 cardinality, most of the selected mappings are correct.

Performance of SP. The similarity propagation (SP) method that we propose differs from the other structural methods discussed above in several aspects. Note that the similarity scores produced by SP are not absolute but relative values due to the normalization process at the end of each running iteration. SP propagates similarity values from one pair of entities to another, hence, if two entities have a similarity score higher than 0, then they are considered as similar to a certain degree. Thus, with a low threshold filter, SP discovers more correct mappings than with a high threshold value. Moreover, the similarity score of a pair of entities depends not only on their current status but also on the status of other related (neighboring) pairs. The more neighbors with high similarity a pair of entities has, the likelier that they are matched. For example, when the threshold is set to 0.01, SP discovers 1298 $(= 5941 - 4643)$ additional correct mappings and 247 $(= 274 - 27)$ incorrect ones in comparison with the init mappings. Therefore, SP distinguishes well correct and incorrect mappings by ranking the similarity scores which is the main reason why this method outperforms the other local structural methods discussed above when the filter threshold is low.

5 Impact of Noisy Input on Structural Matchers

In this experiment, we evaluate the behavior of different structural matchers when we add noise into the mappings that these methods take as an input from a terminological matcher. Here, we call "noise" a pair of dissimilar entities labeled as similar by the terminological matcher. Indeed, in real matching scenarios, a terminological method rarely produces 100% precision, consequently, it rarely provides input mappings without noise to the structural methods. We will study the impact of this noise on the mappings discovered by several structural methods with the aim to outline the most stable among these methods with respect to the presence of noise.

5.1 Methods and Evaluation Strategy

For these experiments, we have used the Benchmark 2011 dataset from the OAEI campaign. The reasons for this choice given in Section 4 are valid here, as well, since we are dealing with structural methods.

At ***terminological level***, we use the identical metric. To produce noise, we add a number of random incorrect mappings, which correspond to $N\%$ of the size of the original init mappings, with $N \in \{0, 10, .., 100\}$.

At ***structural level***, the matcher takes input mappings from the terminological matcher. According to the experiments in Section 4, we select the best threshold filter for each structural method. For example, $\theta_{SP} = 0.01, \theta_{DESCENDANTS} = 0.01, \theta_{ADJACENTS} = 0.07$, etc.

At each iteration, we count the total number of correct mappings and the total number of incorrect mappings that a structure method produces over all 103 test cases contained in the Benchmark 2011 dataset.

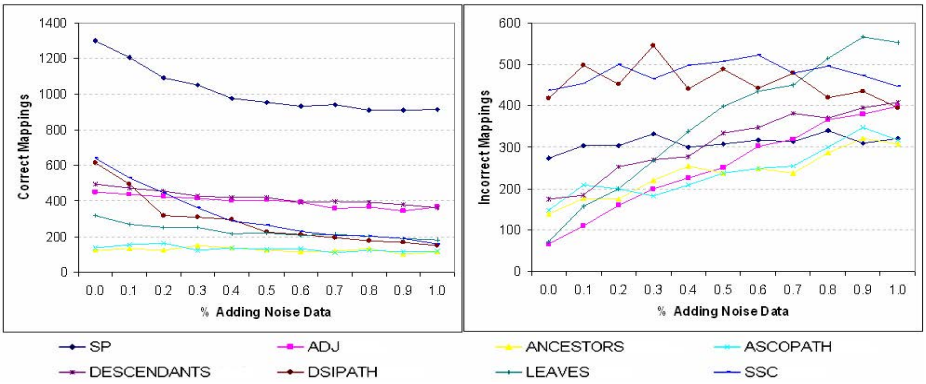


Fig. 4. Impact of input noise on structural matchers

5.2 Results and Discussion

Fig. 4 shows the total number of correct and incorrect mappings produced by the structural methods at each time when noisy data are added to the input. When noisy data are added, the number of correct mappings discovered by all the methods decreases. Regarding the number of incorrect mappings, it increases for all methods, except for DSIPATH and SSC. Note that DSIPATH and SSC differ from the other local structural methods in terms of the interaction between the entities in an ontology. For example, the similarity of two entities computed by DSIPATH strongly depends on the similarity provided by input mappings and decreasingly depends on the similarity of their parents, grandparents, etc. Consider two entities of two input ontologies. If noise appears at the same level in their paths to the root, their similarity will be impacted by this noise, otherwise, it will not. Therefore, the impact of noise in discovering further mappings depends on the position of the entities in the hierarchies of the input ontologies. Because noise is generated randomly, its impact is hard to predict for these methods. Other structural methods use set operations (i.e., intersection, union) with no hierarchical consideration for the elements. When noise appears in the set of ancestors or descendants of two entities, the noise will directly propagate errors to them. Therefore, as seen in Fig. 4, the number of incorrect mappings increases in almost all structural methods of this type.

This experiment shows the dominance of similarity propagation (SP) over other structural methods in terms of stability. When noisy data reaches 100%, SP still discovers 913 additional correct mappings in comparison to the init mappings. Note that the maximum number of correct mappings discovered by the other methods is only 612 mappings with no noise added. Moreover, from 0% to 100% of the noisy data, SP produces only 57 $(321 - 274)$ additional incorrect mappings. In contrast, for example, the LEAVES method produces 481 $(553 - 72)$. This is explained by the fact that SP takes into account all kinds of semantic relations of entities such as concept-concept, concept-property and property-property, which reduces the impact of noise.

6 Interaction of Terminological Matchers with Structural and Semantic Matchers

In this evaluation, we are going to study the impact of the quality of the input mappings provided by several terminological methods on the matching quality of structural and semantic matchers. More precisely, we are interested in discovering which are the terminological methods that provide best performance of the structural and semantic matchers for a given mapping selection threshold.

6.1 Methods and Evaluation Strategy

To carry out this experiment, we have used the Conference dataset from the OAEI campaign, which is a real world dataset from the domain of scientific publishing. Our evaluation strategy is described as follows.

At ***terminological level***, we have used three different methods to produce initial mappings. The choice of these matchers has been motivated by the study described in Section 3 and the results shown in Fig. 2. We have chosen QGrams representing token-based methods and ISUB representing edit-based methods because they show different behaviors when the terminology-based filter threshold changes as compared to the other methods. In addition, we have included IR, representing global methods, which is the best performing among these methods.

At ***structural level***, we have considered SP which takes input from the terminological matchers and performs similarity propagation. This choice is justified by the fact that this method has shown to perform best in the experiments in Section 4.

At ***semantic level***, we use the global diagnosis optimization method proposed in [13] which refines input terminological mappings in order to remove inconsistent ones.

We have studied the performance of each of the terminological methods when used alone and when used as an input for the structural and the semantic methods, respectively. At each iteration, the matching quality is evaluated by comparing the discovered alignment to a reference alignment.

6.2 Results and Discussion

Fig. 5 shows the performance of the terminological methods used alone and in combination with a structural matcher (SP) again as a function of the mapping selection threshold. Fig. 6 shows the behavior of the same terminological methods, this time taken as an input by a semantic matcher. The first observation is that the structural and the semantic methods combined with terminological matchers have similar behaviors, therefore the following analysis will encompass both.

Globally, the combined methods outperform the single terminological methods. Similarly to the previous experiments, the overall performance increases by increasing the threshold value. Quite straightforwardly, the quality of the combined methods increases simultaneously with the quality of the single terminological methods.

Further, we notice that the methods based on QGrams tend to be more stable over the variations of the filter threshold and provide high quality results already at low filter values. This is explained by the fact that the QGrams measure is based on Jaccard similarity computation and as soon as the threshold value reaches 0.6, the matcher already

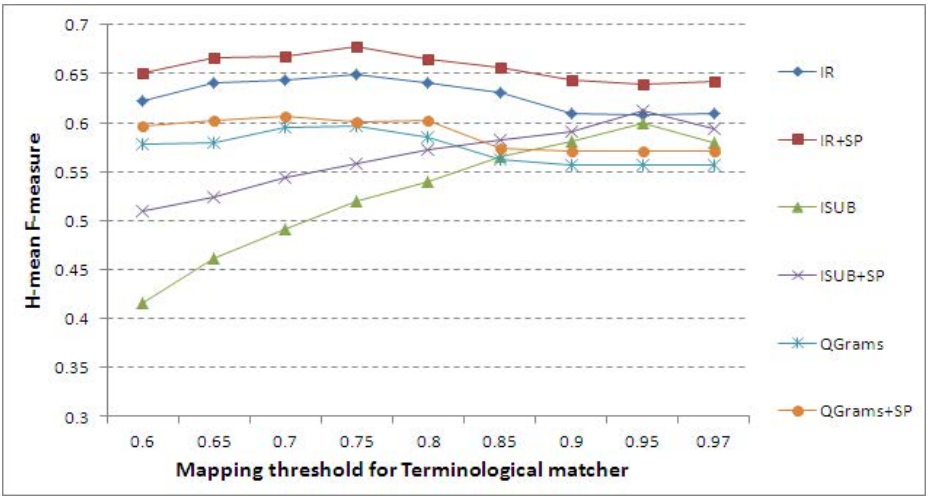


Fig. 5. Interaction of terminological methods with a structural matcher (SP) w.r.t. different values of the mapping selection filters

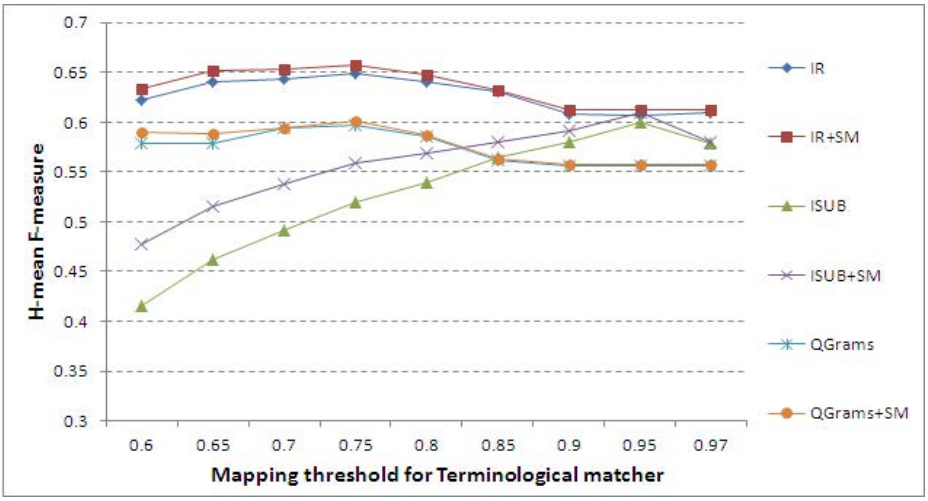


Fig. 6. Interaction of terminological methods with a semantic matcher (SM) w.r.t. different values of the mapping selection filters

accounts for two third of the overlapping tokens. The methods based on ISUB have a different behavior – they have an almost linear growth of the performance as a function of the filter threshold, reaching higher values of the F-measure than the ones of the QGrams methods for thresholds above 0.9, both for structural and semantic approaches.

We explain that by the fact that at a certain level of the threshold value, the number of incorrect mappings is always higher than the number of correct mappings, especially due to the 1:1 cardinality. Therefore, when the threshold value increases, the number of removed incorrect mappings will get higher than the number of removed correct

mappings. Thus, the overall quality increases. However, after surpassing the threshold of 0.95 the quality decreases again. This is due to the fact that when the threshold is that high, only identical or nearly identical strings are passed (i.e. the overall number of passed entities decreases).

Finally, we note that the mapping selection component is a very important intermediate level between the terminology matchers and the structural or semantic ones in order to select output of each matching component. Indeed, the quality of the produced alignments is much worse if no mapping selection is performed. In the experiments, the role of mapping selection is shown by varying the value of the filter threshold.

As a general conclusion, we outline the fact that both the structural and the semantic matchers boost the performance of both local and global terminological methods, but perform best by taking input from the global IR method.

7 Conclusion

In this empirical study, we have presented an analysis of the interaction between the components of an ontology matching system, seen as a chain in which the resulting mapping of a given module is the input to another. We have used evaluation data from the OAEI campaign. In the first place, we were interested in the impact of the mapping selection module on the performance of terminological and structural methods revealing the advantage of using global methods vs. local ones. Further, we have carried an extensive study on the flaw of the performance of a structural method in the presence of noisy input coming from a terminological method. Finally, we have analyzed the behavior of a structural and a semantic matcher with respect to different inputs taken from different terminological methods at different values of the mapping selection filter.

The results of this study are oriented towards researchers and practitioners and are meant to serve as a guide in the design and the use of a matching tool. Our outcomes provide a support on the choice of matchers and the effects that can be expected in their selection and combination. The ultimate goal has been to give the user control and understanding of the mechanism behind an ontology matching system.

References

[1] Cruz, I.F., Stroe, C., Caci, M., Caimi, F., Palmonari, M., Antonelli, F.P., Keles, U.C.: Using agreementmaker to align ontologies for oaei 2010. In: OM (2010)
[2] Dieng, R., Hug, S.: Comparison of personal ontologies represented through conceptual graphs. In: ECAI, pp. 341–345 (1998)
[3] Euzenat, J., Shvaiko, P.: Ontology matching. Springer, Heidelberg (2007)
[4] Giunchiglia, F., Shvaiko, P., Yatskevich, M.: S-match: an algorithm and an implementation of semantic matching. In: Bussler, C.J., Davies, J., Fensel, D., Studer, R. (eds.) ESWS 2004. LNCS, vol. 3053, pp. 61–75. Springer, Heidelberg (2004)
[5] Huber, J., Sztyler, T., Nößner, J., Meilicke, C.: Codi: Combinatorial optimization for data integration: results for oaei 2011. In: OM (2011)
[6] Jean-Mary, Y.R., Shironoshita, E.P., Kabuka, M.R.: Ontology matching with semantic verification. Web Semantics: Science, Services and Agents on the World Wide Web 7(3), 235–251 (2009)

[7] Jian, N., Hu, W., Cheng, G., Qu, Y.: Falcon-ao: Aligning ontologies with falcon. In: Proceedings of K-CAP Workshop on Integrating Ontologies, pp. 85–91 (2005)
[8] Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. CoRR (1997)
[9] Jiménez-Ruiz, E., Cuenca Grau, B.: LogMap: Logic-based and scalable ontology matching. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 273–288. Springer, Heidelberg (2011)
[10] Thanh Le, B., Dieng-Kuntz, R., Gandon, F.: On ontology matching problems. In: ICEIS (4), pp. 236–243 (2004)
[11] Lin, D.: An information-theoretic definition of similarity. In: ICML, pp. 296–304 (1998)
[12] Mao, M., Peng, Y., Spring, M.: A harmony based adaptive ontology mapping approach. In: SWWS, pp. 336–342 (2008)
[13] Meilicke, C.: Alignment incoherence in ontology matching. Thesis (2011)
[14] Meilicke, C., Stuckenschmidt, H.: Analyzing mapping extraction approaches. In: OM (2007)
[15] Melnik, S., Garcia-Molina, H., Rahm, E.: Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In: ICDE, pp. 117–128 (2002)
[16] Ngo, D., Bellahsene, Z., Coletta, R.: A generic approach for combining linguistic and context profile metrics in ontology matching. In: Meersman, R., et al. (eds.) OTM 2011, Part II. LNCS, vol. 7045, pp. 800–807. Springer, Heidelberg (2011)
[17] Ngo, D., Bellahsene, Z., Coletta, R.: A flexible system for ontology matching. In: Nurcan, S. (ed.) CAiSE Forum 2011. LNBIP, vol. 107, pp. 79–94. Springer, Heidelberg (2012)
[18] Ngo, D.H.: Enhancing Ontology Matching by Using Machine Learning, Graph Matching and Information Retrieval Techniques. PhD thesis, University of Montpellier 2 (2012) (in print)
[19] Ngo, D.H., Bellasene, Z., Coletta, R.: Yam++ results for oaei 2011. In: OM (2011)
[20] Shvaiko, P., Euzenat, J.: Ontology matching: State of the art and future challenges. IEEE TKDE 99 (2011)
[21] Stoilos, G., Stamou, G., Kollias, S.: A string metric for ontology alignment. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 624–637. Springer, Heidelberg (2005)
[22] Sunna, W., Cruz, I.F.: Structure-based methods to enhance geospatial ontology alignment. In: Fonseca, F., Rodríguez, M.A., Levashkin, S. (eds.) GeoS 2007. LNCS, vol. 4853, pp. 82–97. Springer, Heidelberg (2007)
[23] Tversky, A.: Features of similarity. Psychological Review 84, 327–352 (1977)
[24] Wang, P., Xu, B.: Lily: Ontology alignment results for oaei 2009. In: OM (2009)

Towards Evaluating Interactive Ontology Matching Tools

Heiko Paulheim[1], Sven Hertling[2], and Dominique Ritze[1]

[1] University of Mannheim, Germany
Research Group Data and Web Science
{heiko,dominique}@informatik.uni-mannheim.de
[2] Technische Universität Darmstadt, Germany
Knowledge Engineering Group
hertling@ke.tu-darmstadt.de

Abstract. With a growing number of ontologies used in the semantic web, agents can fully make sense of different datasets only if correspondences between those ontologies are known. Ontology matching tools have been proposed to find such correspondences. While the current research focus is mainly on fully automatic matching tools, some approaches have been proposed that involve the user in the matching process. However, there are currently no benchmarks and test methods to compare such tools. In this paper, we introduce a number of quality measures for interactive ontology matching tools, and we discuss means to automatically run benchmark tests for such tools. To demonstrate how those evaluation can be designed, we show examples on assessing the quality of interactive matching tools which involve the user in matcher selection and matcher parametrization.

1 Introduction

Ontologies are used for describing information in the semantic web as well as for assigning meaning to data. Until now, there has been no commonly agreed upon a universal ontology, and it is unlikely that such an ontology will ever exist. On the contrary, there is a wide spectrum of ontologies used in the semantic web. For example, in the Linked Open Data cloud, more than half of the 295 datasets use their own ontologies.[1]

To use information from those ontologies in a reasonable way, ontology alignments, i.e., links between those ontologies, are necessary. Ontology matching tools are capable of finding such alignments. In the past, research on ontology matching tools has been focused on developing fully automatic ontology matching tools to a large extent.

Performing ontology matching fully automatically in high quality is hard. For many real-world datasets, fully automatic state of the art tools still yield results at a quality level that is unsatisfying for many use cases. At the recent ontology alignment evaluation initiative (OAEI),[2] the best fully automatic ontology matching tools have yielded a result quality[3] of about 70% for single-language and 40% for multi-lingual matching tasks from the conference domain [1].

[1] http://wifo5-03.informatik.uni-mannheim.de/lodcloud/state/
[2] http://oaei.ontologymatching.org/
[3] In terms of F-measure, i.e., the harmonic mean of recall and precision.

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 31–45, 2013.

Our hypothesis implies that there is an upper bound to the quality of the alignment which is hard to exceed by fully automatic ontology matching tools. As stated by Falconer and Noy, ontology matching is "a very challenging problem for both man and machine" [10], which calls for semi-automatic approaches combining the strengths of automatic matching algorithms and the expertise of domain experts in the matching process. On the other hand, domain experts are a scarce and expensive resource. This makes it necessary to a) design tools that draw maximum benefits from as little user interaction as possible, b) define suitable evaluation measures that capture those benefits and interactions, as well as the trade-off between them, and c) provide automatic evaluation approaches for such tools.

Incorporating user interaction in ontology matching tools is still a major challenge in ontology matching today [28]. Furthermore, unlike the OAEI benchmarks that measure the quality of fully automatic ontology matching tools in terms of recall, precision, and F-measure, there are no commonly agreed upon quality measures for interactive ontology matching tools, let alone comparative evaluations [10].

This paper introduces a number of quality measures for interactive ontology matching tools, inspired by a similar field of research in the area of machine learning, i.e., *active learning* [25]. Furthermore, we discuss how that quality can be measured in fully automatic test settings.These test settings will form the basis of a new branch of tests in future OAEI campaigns.

The rest of this paper is structured as follows. In Section 2, we discuss previous approaches to interactive ontology matching and evaluations. Section 3 introduces a general framework for describing interactive ontology matching tools. Based on that framework, we discuss a number of measures in Section 4. To illustrate how interactive ontology matching tools can be evaluated, we show two experiments for interactive matching in Section 5, that deal with matcher selection and parameterization. We conclude with a summary and an outlook on future work.

2 Related Work

Interactive ontology matching is closely related to active learning, a problem class in machine learning where the algorithm actively presents instances to label to a user [25]. The idea behind active learning is that a learning algorithm can minimize the workload to label examples if it is able to choose examples that are "interesting" for learning, e.g., borderline cases. Active learning has been successfully applied to other fields, related to ontology matching. Isele et al. discuss an active learning approach to generate linkage rules for the Web of Data [14], de Freitas et al. provide a method for detecting data duplication in databases [5] and Rodler et al. use it for ontology debugging [26].

In general, there are several possibilities how and at which point of time to involve the user in the matching process. This can be either before, during or after the matching process. Common examples to improve the matching by involving the user include defining system configurations, creating anchor mappings, correcting suggested correspondences, or evaluating the created alignment.

Several matching systems provide a configuration which can be adapted by the user according to the actual matching task. Since defining a configuration is a difficult task

for domain experts, some approaches ask users for example mappings or validation of generated correspondences instead [23,27].

Most ontology matching systems combine different matching strategies. Some of the systems implementing machine learning for selecting strategies and fully automatically learn how to best combine the matching methods, e.g. *GLUE* [6]. Other approaches, e.g. [7], even combine several different matching systems. Beside automatic combination, some systems involve the user by asking for validation of created correspondences [6,8,16] or request an initial list of correspondences and non-correspondences [32]. Such approaches are capable of outperforming conventional systems (in terms of quality), but it is not clear how the additional effort of the training or even user interaction pays off and to which amount, nor how to measure that trade-off between user efforts and improvement in alignment quality.

Besides the configuration and combination of matching systems, users can help to improve the alignment, or the systems can support the users to generate an alignment. The improvement can either be done a posteriori or (interactively) during the process.

One idea to perform an improvement after the matching is to let the user rate the correspondences such that other users can take advantage of this rating [22]. This strategy is independent of the applied matching system, however, cannot help to improve future mappings since the rating is not fed back into the matching systems. Approaches such as *PROMPT* perform the alignment generation interactively, ask the user for feedback, indicate conflicts [20], and/or provide a proper visualization to support the decision [3]. Moreover, they usually try to reduce the number of user interactions.Whenever a domain expert is asked to manually create mappings, several tools can be taken into account to support the process, e.g., by showing partial results when certain matching rules are applied [2,19] or to point the user to places where attention is required [18].

Since collaborative strategies are getting very popular, some matching systems even apply concepts like crowd sourcing [24] or gamification [30] to generate ontology alignments. Obviously, these approaches require a lot of user interactions.

The proposed systems clearly differ in the kind and amount of user involvement. Most of the corresponding papers provide some evaluation, but they are rarely comparable since they often apply different data sets or focus on various measures, e.g. quality, runtime, and/or amount of user interaction.

Evaluation of ontology matching tools, such as the ontology alignment evaluation initiative (OAEI), have focused on non-interactive aspects of the systems so far, and do not take any user interaction into account. In contrast, Lambrix and Edberg [17] performed an evaluation to compare tools with user involvement. They compared two interactive matching systems with respect to availability, stability, representation language, functionality, assistance, precision and recall of the mapping suggestions and time, and also evaluated user satisfaction with a questionnaire. However, an evaluation campaign involving many tools and including a certain amount of users would result in an enormous effort and is thus hardly feasible.

Falconer and Storey [11] proposed a theoretical framework with several principles and corresponding software requirements for ontology matching systems. They introduce several functionalities each a tool should provide. These functionalities are concerned with user analysis and decision making, interaction, analysis and generation as

well as representation. Their work only shows a theoretical framework but does not provide any benchmarks or evaluation techniques.

So far, several methods for interactive ontology matching have been proposed. However, as stated in [10], "evaluation of such [semi-automatic ontology matching] tools is still very much in its infancy." With this paper, we aim at closing that gap by providing a set of measures and a toolkit for fully automatic evaluation of interactive ontology matching tools.

3 Generic Framework

Non-interactive ontology matching tools do not provide any point of interaction between their invocation and the delivery of the final alignment. In the standard model introduced by Euzenat and Shvaiko [9], ontology matching tools take two ontologies and an (optional) alignment as input, and, optionally based on some parameters and external resources, deliver a final alignment. We extend that model to include interaction with a user.

To describe those interactions, we use the following convention: an alignment is a set of triples, also called correspondences

$$< o_1 \# e_1, o_2 \# e_2, r >, \tag{1}$$

where $o_1 \# e_1$ is an element from ontology one, $o_2 \# e_2$ is an element from ontology two, and r is a relation that holds between the two, such as equality or subsumption. While most ontology matching tools also deliver a confidence score for each triple, we consider that confidence score as meta-information on the alignment rather than part of the alignment itself. Furthermore, we only consider *simple* mappings that relate single elements, and not *complex* mappings, which might call for different forms of interaction. Examples for possible interactions include, but are not limited to:

Asking for Validation of a Candidate Alignment. The tool provides a mapping element $< o_1 \# e_1, o_2 \# e_2, r >$ and asks the user if that mapping is correct or not.

Asking for Definition of the Relation in a Candidate Alignment. The tool provides a mapping element $< o_1 \# e_1, o_2 \# e_2, X >$ and asks the user for filling in the variable X for the relation that holds between e_1 and e_2, e.g. *broader than* or *equivalent to*.

Asking for Completion of an Element in a Candidate Alignment. The tool provides a mapping element with a variable, e.g., $< o_1 \# e_1, X, r >$ and asks the user to fill in the variable X, if there is a sensible substitution.

Figure 1 depicts our generic framework. The tool may issue a hypothesized partial alignment (which may contain variables, as discussed above) to a domain expert, and asks for carrying out a certain action, such as validation or completion. The domain expert returns a correct partial alignment (which may be empty, in case a hypothesized alignment is completely discarded by the user). After a certain number of interactions (which may be after a fixed a number of interactions or when the tool has derived an alignment it considers to be stable), the tool delivers a final alignment.

Note that the list of interactions above is not fixed, and that the interactions may go beyond purely validating the system's output. For example, users may also be asked for

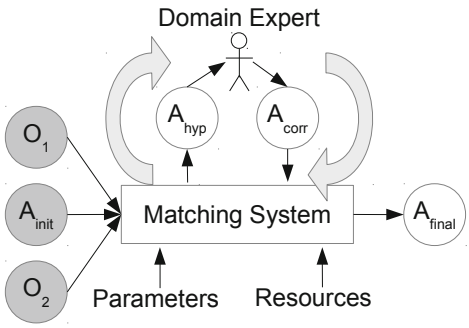


Fig. 1. A generic framework for interactive ontology matching, as an extension of [9] and [10]. Unlike fully automatic matching systems, interactive ontology matching systems support (an arbitrary number of) interactions between the system and a domain expert, triggered by the matching system. Typically, the ontology matching system will provide a hypothesized, partial alignment, possibly containing variables, which is validated and completed by the domain expert.

arbitrary example mappings (using the third type of interaction with two variables), or for the confirmation of completeness of a partial alignment.

4 Evaluation Measures for Interactive Ontology Matching Tools

Non-interactive ontology matching tools are evaluated using recall, precision, and in particular F-measure [9]. Interactive ontology matching tools require different measures which take into account the achieved result quality (i.e., the F-measure), as well as the economic use of the domain expert's workload, e.g., the amount of mappings a user has to validate.

To compute a domain expert's workload, we assign costs c_i to each action a_i performed by the domain expert. This is a *generic cost*, which can be filled by different *actual measures* when implementing our model, such as the time consumed, the money paid to an expert, or the money spent on a crowdsourcing platform. With those values, we can compare two matchers both by the F-measure they achieve, as well as by the cost of interaction they have caused.

For an *automatic evaluation*, it is often necessary to further *simplify the cost model*. For example, an automatic evaluation scenario may only allow one type of interaction. In that case, it is possible to assign a constant weight (e.g., 1) to each interaction. In the case that different interactions are allowed, the weights have to be fixed in a more sophisticated manner (e.g., letting users perform sample tasks of each type, and computing average times for those tasks). In scenarios where different interactions are possible, those may impose different cognitive loads on the domain expert (e.g., confirming or discarding an element is less demanding than completing an incomplete mapping). Thus, those different loads have to be reflected in different costs, which of course are only an approximation of the actual costs.

While it is easy to optimize F-measure (let the domain expert do everything) and cost (do not involve the domain expert at all) on their own, achieving a reasonable trade-off

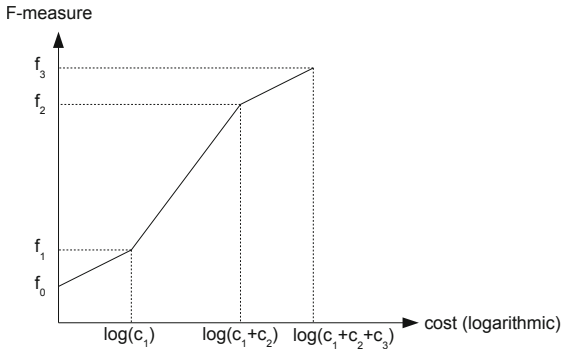


Fig. 2. An example learning curve for an ontology matcher. After each interaction step, the F-measure of the preliminary alignment is graphed against the cost consumed so far. The cost axis uses a logarithmic scale to reward fast convergence towards an optimal F-measure.

between the two is more challenging. Furthermore, depending on the actual interactive ontology matching algorithm, it may be difficult to determine when to stop, as better results may still be achieved through more user involvement. In order to account for those differences, we demand that each matching tool is capable of delivering a *preliminary alignment* A_{prelim} at any given point in time. This alignment represents the *best alignment found so far*.[4] Having preliminary alignments allows for plotting a *learning curve* of F-measure relative to the cost consumed, as shown in Fig. 2.

As discussed above, interactive ontology matching follows a similar task setting as active learning. Active learning tools are often evaluated by drawing a graph depicting the quality of the learning algorithm (e.g., its ROC value) plotted against the number of examples presented to the user. The normalized area under that curve (referred to as AUL, the area under the learning curve) is then used as a measure for comparing active learning tools [12]. A high AUL is achieved if a tool reaches a high overall F-measure and converges towards that value with few user interactions. To reward quick convergence (and, hence, efficient use of the domain expert's workload), the cost axis uses a logarithmic scale.

Given that the F-measure after the i-th user interaction is measured as f_i using the preliminary alignment A_{prelim}, and the cost of the i-th interaction is measured as c_i, the normalized AUL, using a logarithmic scale for the cost axis, can be computed as

$$AUL = \frac{1}{log \sum_{i=1}^{n} c_i} \sum_{i=1}^{n} \left(log \sum_{k=1}^{i} c_k - log \sum_{k=1}^{i-1} c_k \right) \frac{f_i + f_{i-1}}{2} \tag{2}$$

In that case, f_0 is the F-measure that the tool achieves without any user interaction, such as a default initial mapping determined with a simple string-based similarity measure. When using the AUL measure for comparing two matchers performing user interactions at a different overall cost, the final F-measure of the matcher consuming lower cost is

[4] Tools that do not maintain any intermediate results may simply return an empty alignment.

used for the remaining interactions steps of the matcher consuming higher cost, so that the AUL values can be compared in a meaningful manner. Usually, the largest cost consumed is normalized to 1, so that AUL is a value between 0 and 1.

Next to AUL, the maximum F-measure value f_{max} reached during the interactive matching process, as well as the final F-measure value f_{final} are of interest. For an ideal interactive ontology matching tool, the learning curve is monotonously increasing, thus, $f_{max} = f_{final}$ holds in that case. However, real interactive ontology matching tools will probably not always expose that behavior.

Besides new quality measures, interactive ontology matching introduces a new baseline as well. After each interaction, the human expert may have generated a partial mapping. For example, after correcting n mappings, there are is a partial mapping up to size n. The F-measure achieved with that mapping, depicted as f_{human}, serves a baseline for interactive matching tools: a matching tool that makes use of a number of interactions with a domain expert should provide a better alignment than that created by the domain expert alone.

For automatically evaluating interactive matching systems, we use the architecture depicted in Fig. 3.[5] An evaluation system which holds the reference alignment creates an oracle that answers the queries posed by the matching system. It informs the evaluation system whenever the oracle is called. The evaluation system can then ask the matching system for a preliminary alignment, as discussed above, whenever the oracle is called, in order to plot the learning curve and compute the final AUL value once the matching system has returned its final alignment.

5 Experiments

To illustrate how interactive ontology matching tools are evaluated, as well as providing some reasonable use cases for interactive ontology matching, we have conducted two experiments: interactively selecting a matcher for a given problem, and interactively tuning a matcher's parameters. These experiments use existing matchers and results from previous OAEI challenges and an illustration of how to implement our evaluation framework, rather than an in-depth study in matcher combination and parameterization, as the results strongly depend on the matchers and data used for the experiments.

5.1 Experiment 1: Matcher Selection

Not all ontology matching tools perform equally well on each ontology matching task. Thus, automatically selecting a matching tool for a given pair of ontologies is desirable to allow for optimal matching results. However, selecting a matching tool is difficult for an end user, who may be a domain expert, but not an expert for ontology matching tools [29].

As a possible solution, matchers can be selected indirectly by a domain expert in an interactive matching setting. By letting users rate individual mappings generated by

[5] An implementation of that framework, based on the Alignment API [4], is available from `http://www.ke.tu-darmstadt.de/resources/ontology-matching/interactivematching/`

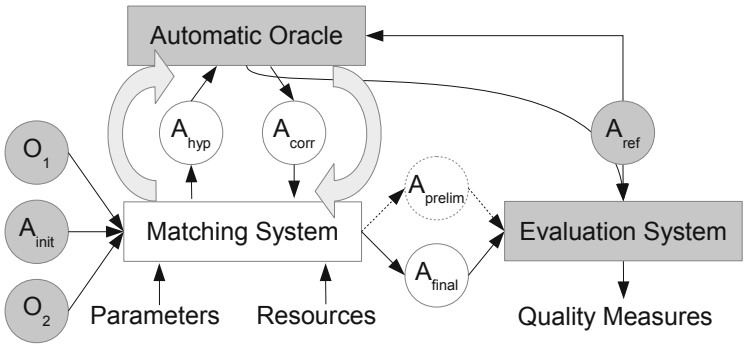


Fig. 3. Framework for evaluating interactive ontology matching tools. In order to facilitate automatic evaluation, the domain expert in Fig. 1 is replaced by an automatic oracle. The oracle is observed by the evaluation system in order to track the cost consumed. The evaluation measures are calculated based on the cost consumed and the quality of the preliminary (i.e., the best alignment found so far at a given point in time) and final alignments generated by the matching tool.

different matching tools, we can select the matching tool that receives the best ratings for a given pair of ontologies. In that case, we assume that the performance on a rated partial alignment correlates with the performance on the complete alignment, which is a valid assumption, as shown in [23].

For this experiment, we have used the dataset from the OAEI 2012 conference track.[6] The public part of this dataset consists of seven ontologies and pairwise reference alignments, resulting in a total of 21 matching problems. In order to test the matcher selection, we have used three of the currently best performing matchers on this dataset: CODI [13], LogMap [15], and Optima [31].[7]

As shown in Table 1, they all have their strengths and weaknesses: Each tool is superior on a number of individual problems (CODI: four, LogMap: seven, Optima: seven, tied: three). This shows that not every matching problem is best addressed by the same strategy: While CODI reformulates the matching problem as an optimization problem, Optima is a combination of "classic" matching techniques, such as string similarities and structural measures, and LogMap uses reasoning for computing a mapping. LogMap, the best tool among the three, achieves an average F-measure of .69, however, if we were able to always select the best tool, that selection would yield an average F-measure of .73 (and hence even beat the best tool in the competition, YAM++, which reaches an overall F-measure of .71).

The algorithm for selecting matchers is designed as follows: each matcher is run on the dataset. From the results, we collect all mappings that are found by at least one, but not by all matchers. Those mappings are put in a list in random order, and presented to the user for validation. We use one basic kind of interaction, i.e., validating if a

[6] `http://oaei.ontologymatching.org/2012/conference/`

[7] We did not take the best performing tool YAM++ into account, because it was the best tool in the majority of all cases, which makes it uninteresting for demonstrating matcher selection.

Table 1. Setup and results of the matcher selection experiment. The table depicts the individual results of the matchers included in the experiment, with the best results marked in bold. For both selection strategies, the final and the maximum F-measure, the total cost of interactions and the AUL value are depicted.

	Matcher			Selection by F-measure				Selection by scoring			
Data	CODI	LogMap	Optima	f_{final}	f_{max}	cost	AUL	f_{final}	f_{max}	cost	AUL
cmt-conf	0.56	0.54	**0.63**	0.63	0.63	14	0.59	0.63	0.63	14	0.61
cmt-confOf	0.56	0.45	**0.61**	0.61	0.61	7	0.56	0.61	0.61	7	0.55
cmt-edas	0.73	**0.76**	0.67	0.73	0.76	8	0.71	0.67	0.76	8	0.70
cmt-ekaw	**0.67**	**0.67**	0.5	0.67	0.67	11	0.67	0.67	0.67	11	0.67
cmt-iasted	**0.89**	**0.89**	0.72	0.89	0.89	3	0.89	0.89	0.89	3	0.89
cmt-sigkdd	0.75	**0.91**	0.87	0.91	0.91	7	0.91	0.91	0.91	7	0.91
conf-confOf	0.67	0.76	**0.80**	0.80	0.80	20	0.80	0.80	0.80	20	0.69
conf-edas	0.60	**0.73**	0.63	0.73	0.73	18	0.70	0.63	0.73	18	0.67
conf-ekaw	0.5	**0.56**	0.45	0.56	0.56	17	0.56	0.56	0.56	17	0.56
conf-iasted	0.4	**0.61**	0.44	0.61	0.61	17	0.61	0.61	0.61	17	0.61
conf-sigkdd	0.71	0.71	**0.77**	0.77	0.77	11	0.73	0.77	0.77	11	0.74
confOf-edas	0.51	0.67	**0.69**	0.69	0.69	13	0.67	0.69	0.69	13	0.68
confOf-ekaw	0.74	0.80	**0.83**	0.83	0.83	10	0.81	0.83	0.83	10	0.82
confOf-iasted	**0.67**	0.62	0.64	0.67	0.67	11	0.64	0.64	0.67	11	0.64
confOf-sigkdd	**0.92**	0.73	0.88	0.92	0.92	6	0.86	0.88	0.92	6	0.83
edas-ekaw	0.62	0.58	**0.64**	0.64	0.64	12	0.63	0.64	0.64	12	0.61
edas-iasted	**0.58**	0.52	0.54	0.58	0.58	13	0.55	0.54	0.58	13	0.54
edas-sigkdd	0.61	**0.64**	**0.64**	0.64	0.64	2	0.64	0.64	0.64	2	0.64
ekaw-iasted	0.67	**0.70**	0.54	0.70	0.70	15	0.70	0.70	0.70	15	0.70
ekaw-sigkdd	**0.86**	0.74	0.80	0.86	0.86	9	0.83	0.80	0.86	9	0.81
iasted-sigkdd	0.75	**0.85**	0.63	0.85	0.85	16	0.83	0.85	0.85	16	0.85
average	0.67	**0.69**	0.66	0.73	0.73	11.4	0.71	0.71	0.73	11.4	0.70

candidate mapping element is correct or not. Based on the user's response, a score for each matcher is computed. We use two different variants for scoring matchers:

F-measure. Based on all true positives and false positives gathered from the user so far, we compute a partial F-measure, as described in [23].

Scoring. For a true positive, all matchers that have found the element increase their score by 1, for a false positive, all matchers that have not found the element increase their score by 1.

For both variants, in case of ties, the tool with the higher a priori quality (i.e., with the better overall performance in OAEI 2012) is returned (in that case, a random selection would also be possible). Since scores are attributed after each interaction step, we can always return a preliminary alignment (i.e., the one produced by the matcher which is currently the best).

The results are shown in Fig. 4 and Table 1. It can be observed that the baseline (i.e., only using the matcher which is best on average) is constantly exceeded by both approaches after only two interactions. The value for f_{human} is always worse than the

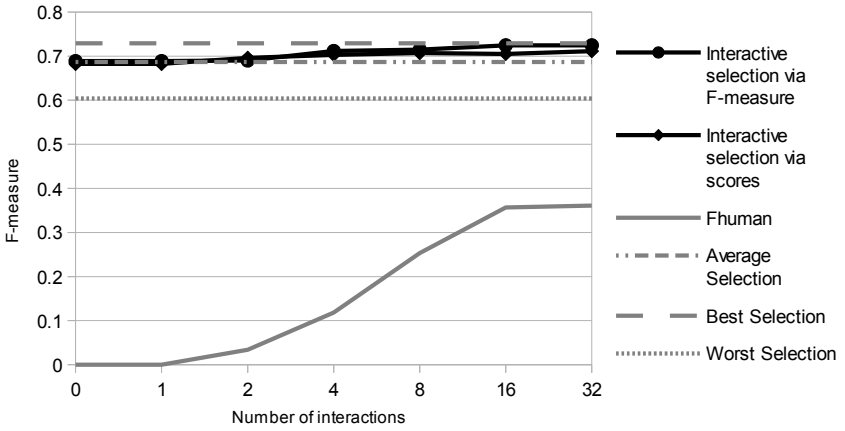


Fig. 4. Resulting learning curves for the matcher selection experiment on the OAEI conference dataset, using three matchers. The best results are achieved by selection via F-Measure. After four interactions, the default selection is outperformed, and the algorithm quickly approaches the theoretical best selection.

baseline, which shows that the approach actually makes significant use of the information gathered from the domain expert.

When comparing both variants, it can be observed that selection by F-measure is slightly superior to selection by scores, as it is less likely to diverge again once it has found an optimal matcher, which can be seen by comparing f_{max} to f_{final}. For selection based on F-measure, the best possible selection is achieved in all but one case (after a maximum of 14 interactions), i.e., $f_{max} \approx f_{final}$; the selection based on scores misses the best possible selection in six out of 21 cases., i.e., $f_{max} > f_{final}$. In both cases, f_{max} equals the best possible selection, i.e., the best matcher is found at least once in the process, but only the approach using selection by F-measure actually sticks to the best selection.

To analyze how the approach can deal with a larger number of matchers, we have repeated the experiment above with all matchers participating in OAEI 2012, again using the conference dataset. The results are depicted in Fig. 5. Here, the superiority of selection via F-measure over selection via scores is even more strongly visible: only the selection via F-measure converges towards the optimum selection, while selection via scores is not capable of outperforming the average selection baseline.

The reason why selection via scores is not optimal is that each true negative (among the false positives found by any other matcher) and each true positive equally increase the score. This can lead to skewed results in some situations. For example, a "defensive" matcher with low recall and high precision can become over-rated in the presence of a matcher producing a large number of false positives. Such effects are avoided using selection via F-measure, which provides a more accurate approximation to the final F-measure achieved in the selection process.

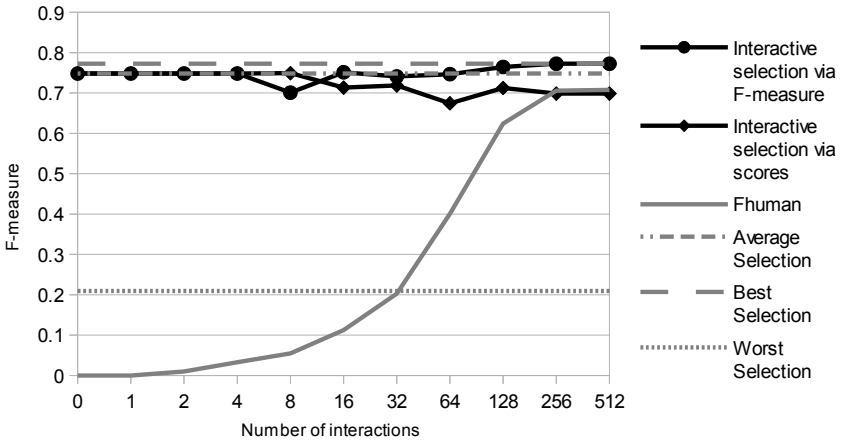


Fig. 5. Resulting learning curves for the matcher selection experiment with all matchers on the OAEI conference dataset. Only selection via F-Measure is capable of finding the best possible selection, but the number of user interactions required is fairly large.

5.2 Experiment 2: Matcher Parametrization

Like selecting a matcher that performs optimally on a dataset, setting good parameters for that matcher is a task that is hard to perform for a person who does not know about the internals of that matcher (it may even be hard for the developer of the matcher, since some parameters are hard to determine without experimentation) [23,29]. In our second experiment, we try to let users determine parameters indirectly via rating candidate alignments instead of direct parameter manipulation.

In this experiment, we use the matching tool WeSeE [21], which, like many tools, requires a parameter for the cutoff threshold above which mappings are returned. As shown in Table 2 for the conference dataset, always selecting an optimal threshold would yield an F-measure of 0.70, while the best global threshold only yields 0.65.

We determine the threshold to select by presenting mappings to the user and collecting the feedback. The presented mappings are selected by using a search window of size w containing mappings around a given threshold.

To find the threshold, we use the following algorithm: starting with thresholds of 0, 0.5, and 1, we initially pick each w mapping element which has a confidence score as close as possible to those values. After rating the $3 \cdot w$ elements, the approximate F-measure on each of those thresholds is calculated based on all the elements rated so far, like in experiment 1. For the best threshold value, we divide the intervals left and right of that value in half and select to mapping elements at those split points for the next round of interaction. The algorithm ends if there are no more mappings elements between two split points.

All costs are set as in experiment 1. The results are shown in Fig. 6 and Table 2 for window sizes w of 1, 3, and 5. It can be seen that the baseline is exceeded in all three variants, and that the results are close to the optimum. For window sizes 1 and 3, the search algorithm is sometimes distracted in a wrong direction, thus, the F-measure does not grow monotonously. For a window size of 5, where a lot of mapping elements

Table 2. Setup and results of the matcher parametrization experiment with WeSeE-Match. The table depicts the matcher's results with a default threshold parameter and with the best possible threshold parameter. For the three variants, the final and the maximum F-measure achieved, as well as the total cost of interactions and the AUL value are depicted.

	Threshold		$w = 1$				$w = 3$				$w = 5$			
Data	def.	best	f_{final}	f_{max}	cost	AUL	f_{final}	f_{max}	cost	AUL	f_{final}	f_{max}	cost	AUL
cmt-conf	0.58	0.62	0.58	0.58	11	0.44	0.58	0.58	16	0.45	0.58	0.58	22	0.43
cmt-confOf	0.43	0.48	0.38	0.38	9	0.34	0.48	0.48	14	0.37	0.48	0.48	22	0.37
cmt-edas	0.76	0.76	0.76	0.76	11	0.51	0.76	0.76	16	0.47	0.76	0.76	22	0.41
cmt-ekaw	0.49	0.67	0.67	0.67	9	0.48	0.67	0.67	14	0.45	0.67	0.67	19	0.42
cmt-iasted	0.89	0.89	0.89	0.89	9	0.57	0.89	0.89	17	0.45	0.89	0.89	22	0.36
cmt-sigkdd	0.82	0.92	0.92	0.92	10	0.65	0.92	0.92	19	0.59	0.92	0.92	20	0.55
conf-confOf	0.71	0.71	0.69	0.69	10	0.49	0.71	0.71	16	0.48	0.71	0.71	22	0.53
conf-edas	0.64	0.69	0.67	0.67	11	0.48	0.67	0.67	17	0.45	0.62	0.67	22	0.40
conf-ekaw	0.45	0.54	0.55	0.55	12	0.53	0.46	0.48	17	0.38	0.52	0.55	29	0.50
conf-iasted	0.38	0.42	0.44	0.44	10	0.28	0.44	0.44	18	0.23	0.35	0.35	22	0.21
conf-sigkdd	0.69	0.69	0.67	0.67	11	0.48	0.69	0.69	18	0.47	0.69	0.69	25	0.44
confOf-edas	0.65	0.69	0.69	0.69	12	0.48	0.69	0.69	16	0.47	0.69	0.69	30	0.44
confOf-ekaw	0.78	0.82	0.76	0.76	10	0.61	0.79	0.79	14	0.72	0.82	0.82	21	0.70
confOf-iasted	0.67	0.71	0.67	0.67	10	0.45	0.71	0.71	16	0.43	0.71	0.71	21	0.37
confOf-sigkdd	0.83	0.86	0.83	0.83	9	0.58	0.86	0.86	16	0.54	0.86	0.86	20	0.48
edas-ekaw	0.50	0.59	0.48	0.48	10	0.39	0.51	0.51	18	0.38	0.56	0.56	22	0.39
edas-iasted	0.56	0.63	0.62	0.62	11	0.41	0.62	0.62	22	0.37	0.62	0.62	30	0.32
edas-sigkdd	0.61	0.77	0.77	0.77	12	0.50	0.72	0.72	18	0.43	0.72	0.72	22	0.40
ekaw-iasted	0.67	0.75	0.75	0.75	20	0.48	0.67	0.67	17	0.36	0.75	0.75	22	0.38
ekaw-sigkdd	0.78	0.78	0.78	0.78	10	0.53	0.78	0.78	14	0.52	0.78	0.78	20	0.49
iasted-sigkdd	0.73	0.81	0.85	0.85	10	0.57	0.85	0.85	22	0.51	0.85	0.85	22	0.45
average	0.65	0.70	0.69	0.69	10.8	0.49	0.69	0.69	16.9	0.45	0.69	0.70	22.7	0.43

are presented to the user, the curve does not grow significantly faster than F_{human}, although it starts from a higher level (the F-measure achieved using a default parameter setting). The results do not differ much in terms of the final F-measure that is reached, but larger window sizes consume more interaction cost and take longer to converge to the optimum, as reflected in the cost and AUL values.

6 Conclusion and Outlook

In this paper, we have discussed the problem of interactive ontology matching and its evaluation. We have introduced a number of evaluation measures, which are generic enough to be applied in an evaluation scenario involving end users or crowd sourcing, as well as allow for automatic evaluation.

To support fully automatic testing and comparison of interactive ontology matching tools, we have proposed a framework which can make use of existing benchmark data and emulate user behavior. We have shown how the framework can be applied to address common problems in ontology matching, such as matcher selection and parameter tuning, by interactive techniques, to compare different variants of interactive matchers, and to draw useful insights from the results provided by our evaluation measures.

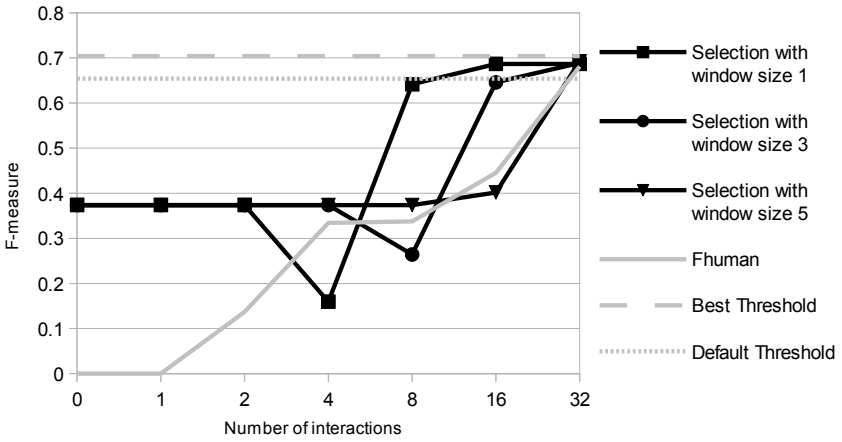


Fig. 6. Resulting learning curves for the matcher parametrization experiment with WeSeE-Match on the OAEI conference dataset. Larger window sizes result in better results, but slower convergences (i.e., more user interaction).

In this paper, we have introduced a set of example interactions. Those may range from accepting or rejecting a candidate mapping to completing mappings. This set of examples is not an exhaustive list. For the future, we envision a full catalog of possible interactions during ontology matching. In particular, when turning to complex mappings involving more than one element on each side, the set of interactions may encompass more interactions.

Such a catalog of interactions would be even more informative with a throrough evaluation of the costs that typically come with those interactions. These could be obtained, for example, through user studies measuring the average time spent on different types of interactions. A more fine-grained weighting of interactions than simply assigning a weight to a class of interactions may also be beneficial. For example, users might be faster in rejecting many false positives (such as $Researcher \equiv Publication$), while true positives may require a closer look (such as $Researcher \equiv Scientist$).

Measuring user experience of interactive ontology matching tools has been out of scope of our work so far. The reason is that we aim at test procedures that can be fully automatized, which is difficult (if not impossible) for measuring user experience. However, for interactive ontology matching tools providing a user interface, measuring user experience is a useful complement to the measures discussed in this paper.

Based on the work presented in this paper, we are planning to introduce a new track to the next OAEI campaign, which will explicitly focus on interactive ontology matching tools. By supplying a test environment, we hope to gather insights in the qualitative comparison of existing interactive matching approaches, as well as encourage researchers in the community to develop novel approaches and algorithms for interactive ontology matching.

References

1. Aguirre, J.L., Eckert, K., Euzenat, J., Ferrara, A., van Hage, W.R., Hollink, L., Meilicke, C., Nikolov, A., Ritze, D., Scharffe, F., Shvaiko, P., Šváb Zamazal, O., Trojahn, C., Jiménez-Ruiz, E., Grau, B.C., Zapilko, B.: Results of the Ontology Alignment Evaluation Initiative 2012. In: Proc. of the 7th Int. Workshop on Ontology Matching (2012)
2. Chalupksy, H.: OntoMorph: A Translation System for Symbolic Knowledge. In: Proc. of the 17th Int. Conference on Knowledge Representation and Reasoning (2000)
3. Cruz, I.F., Stroe, C., Palmonari, M.: Interactive User Feedback in Ontology Matching Using Signature Vectors. In: Proc. of the 28th Int. Conference on Data Engineering, pp. 1321–1324 (2012)
4. David, J., Euzenat, J., Scharffe, F., Trojahn dos Santos, C.: The Alignment API 4.0. Semantic Web 2(1), 3–10 (2011)
5. de Freitas, J., Pappa, G.L., da Silva, A.S., Gonçalves, M.A., de Moura, E.S., Veloso, A., Laender, A.H.F., de Carvalho, M.G.: Active Learning Genetic Programming for Record Deduplication. In: Proc. of IEEE Congress on Evolutionary Computation, pp. 1–8 (2010)
6. Duan, S., Fokoue, A., Srinivas, K.: One Size Does Not Fit All: Customizing Ontology Alignment Using User Feedback. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010, Part I. LNCS, vol. 6496, pp. 177–192. Springer, Heidelberg (2010)
7. Eckert, K., Meilicke, C., Stuckenschmidt, H.: Improving ontology matching using meta-level learning. In: Aroyo, L., et al. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 158–172. Springer, Heidelberg (2009)
8. Ehrig, M., Staab, S., Sure, Y.: Bootstrapping ontology alignment methods with APFEL. In: Gil, Y., Motta, E., Richard Benjamins, V., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 186–200. Springer, Heidelberg (2005)
9. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer (2007)
10. Falconer, S.M., Noy, N.F.: Interactive Techniques to Support Ontology Matching. In: Bellahsene, Z., Bonifati, A., Rahm, E. (eds.) Schema Matching and Mapping, pp. 29–51. Springer (2011)
11. Falconer, S.M., Storey, M.-A.: A cognitive support framework for ontology mapping. In: Aberer, K., et al. (eds.) ISWC/ASWC 2007. LNCS, vol. 4825, pp. 114–127. Springer, Heidelberg (2007)
12. Guyon, I., Cawley, G.C., Dror, G., Lemaire, V.: Results of the Active Learning Challenge. Journal of Machine Learning Research - Proceedings Track 16, 19–45 (2011)
13. Huber, J., Sztyler, T., Noessner, J., Meilicke, C.: CODI: Combinatorial Optimization for Data Integration - Results for OAEI 2011. In: Proc. of the 6th Int. Workshop on Ontology Matching (2011)
14. Isele, R., Jentzsch, A., Bizer, C.: Active learning of expressive linkage rules for the web of data. In: Brambilla, M., Tokuda, T., Tolksdorf, R. (eds.) ICWE 2012. LNCS, vol. 7387, pp. 411–418. Springer, Heidelberg (2012)
15. Jiménez-Ruiz, E., Grau, B.C., Horrocks, I.: LogMap and LogMapLt Results for OAEI 2012. In: Proc. of the 7th Int. Workshop on Ontology Matching (2012)
16. Jiménez-Ruiz, E., Grau, B.C., Zhou, Y., Horrocks, I.: Large-scale interactive ontology matching: Algorithms and implementation. In: Proc. of the 20th European Conference on Artificial Intelligence (2012)
17. Lambrix, P., Edberg, A.: Evaluation of Ontology Merging Tools in Bioinformatics. In: Proc. of the Pacific Symposium on Biocomputing, pp. 589–600 (2003)
18. McGuinness, D., Fikes, R., Rice, J., Wilder, S.: An environment for merging and testing large ontologies. In: Proc. of the 17th Int. Conference on Principles of Knowledge Representation and Reasoning (2000)

19. Miller, R., Haas, L., Hernandez, M.: Schema mapping as query discovery. In: Proc. of the 26th Int. Conference on Very Large Databases (2000)
20. Noy, N.F., Musen, M.A.: The PROMPT suite: interactive tools for ontology merging and mapping. Int. Journal of Human-Computer Studies 59(6), 983–1024 (2003)
21. Paulheim, H.: WeSeE-Match results for OEAI 2012. In: Proc. of the 7th Int. Workshop on Ontology Matching (2012)
22. Paulheim, H., Rebstock, M., Fengel, J.: Context-Sensitive Referencing for Ontology Mapping Disambiguation. In: Proc. of the 2007 Workshop on Context and Ontologies Representation and Reasoning, pp. 47–56 (2007)
23. Ritze, D., Paulheim, H.: Towards an automatic parameterization of ontology matching tools based on example mappings. In: Proc. of the 6th Int. Workshop on Ontology Matching (2011)
24. Sarasua, C., Simperl, E., Noy, N.F.: CROWDMAP: Crowdsourcing ontology alignment with microtasks. In: Cudré-Mauroux, P., et al. (eds.) ISWC 2012, Part I. LNCS, vol. 7649, pp. 525–541. Springer, Heidelberg (2012)
25. Settles, B.: Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison (2009)
26. Shchekotykhin, K., Friedrich, G., Fleiss, P., Rodler, P.: Interactive ontology debugging: two query strategies for efficient fault localization. Web Semantics: Science, Services and Agents on the World Wide Web 12(13), 88–103 (2012)
27. Shi, F., Li, J., Tang, J., Xie, G., Li, H.: Actively learning ontology matching via user interaction. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 585–600. Springer, Heidelberg (2009)
28. Shvaiko, P., Euzenat, J.: Ontology Matching: State of the Art and Future Challenges. IEEE Transactions on Knowledge and Data Engineering 25(1), 158–176 (2013)
29. Shvaiko, P., Euzenat, J.: Ten challenges for ontology matching. In: Meersman, R., Tari, Z. (eds.) OTM 2008, Part II. LNCS, vol. 5332, pp. 1164–1182. Springer, Heidelberg (2008)
30. Siorpaes, K., Thaler, S., Simperl, E.: SpotTheLink: A Game for Ontology Alignment. In: Proc. 6th Conference for Professional Knowledge Management (2011)
31. Thayasivam, U., Chaudhari, T., Doshi, P.: Optima+ Results for OAEI 2012. In: Proc. of the 7th Int. Workshop on Ontology Matching (2012)
32. To, H.-V., Ichise, R., Le, H.-B.: An Adaptive Machine Learning Framework with User Interaction for Ontology Matching. In: Proc. of the IJCAI 2009 Workshop on Information Integration on the Web, pp. 35–40 (2009)

A Session-Based Approach for Aligning Large Ontologies

Patrick Lambrix[1,2] and Rajaram Kaliyaperumal[1]

[1] Department of Computer and Information Science
[2] Swedish e-Science Research Centre
Linköping University, 581 83 Linköping, Sweden

Abstract. There are a number of challenges that need to be addressed when aligning large ontologies. Previous work has pointed out scalability and efficiency of matching techniques, matching with background knowledge, support for matcher selection, combination and tuning, and user involvement as major requirements. In this paper we address these challenges. Our first contribution is an ontology alignment framework that enables solutions to each of the challenges. This is achieved by introducing different kinds of interruptable sessions. The framework allows partial computations for generating mapping suggestions, partial validations of mappings suggestions and use of validation decisions in (re)computation of mapping suggestions and the recommendation of alignment strategies to use. Further, we describe an implemented system providing solutions to each of the challenges and show through experiments the advantages of the session-based approach.

1 Introduction

In recent years many ontologies have been developed and many of those contain overlapping information. Often we want to use multiple ontologies. For instance, companies may want to use community standard ontologies and use them together with company-specific ontologies. Applications may need to use ontologies from different areas or from different views on one area. In each of these cases it is important to know the relationships between the terms in the different ontologies. Further, the data in different data sources in the same domain may have been annotated with different but similar ontologies. Knowledge of the inter-ontology relationships would in this case lead to improvements in search, integration and analysis of data. It has been realized that this is a major issue and much research has recently been done on ontology alignment, i.e. finding mappings between terms in different ontologies (e.g. [5]).

The existing frameworks for ontology alignment systems (e.g. [3,13]) describe different components and steps in the ontology alignment process such as preprocessing, matching, filtering and combining match results, and user validation of the mapping suggestions generated by the ontology alignment system. Systems based on the existing frameworks function well when dealing with small ontologies, but there are a number of limitations when dealing with larger ontologies. Some recent work (e.g. [15,8]) has defined challenges that need to be addressed when dealing with large ontologies. According to [8] interactivity, scalability, and reasoning-based error diagnosis are required

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 46–60, 2013.

to deal with large ontologies. [15] defines the following challenges related to aligning large ontologies. Regarding scalability [15] discusses efficiency of matching techniques. This is important as many participants in the Ontology Alignment Evaluation Initiative (OAEI, a yearly event that focuses on evaluating systems that automatically generate mapping suggestions) have perfomance problems when dealing with large ontologies. Further, matching with background knowledge should be used (which could include in the [15] interpretation of background knowledge the error diagnosis of [8]). Based on OAEI experience it is also clear that there is a need for support for matcher selection, combination and tuning. There is also a need for user involvement in the matching process. The user can be involved during the mapping generation. Further, as stated by the OAEI organizers [4], automatic generation of mappings is only a first step towards a final alignment and a validation by a domain expert is needed.

In this paper we address these challenges. Our first contribution is an ontology alignment framework that enables scalability, user involvement, use of background knowledge and matcher selection, combination and tuning (Section 2). This is achieved by introducing different kinds of interruptable sessions (computation, validation and recommendation). It is the first framework that allows partial computations for generating mapping suggestions. Currently, to our knowledge, no system allows to start validating mapping suggestions before every suggestion is computed. It also is the first framework that allows a domain expert to validate a sub-set of the mapping suggestions, and continue later on. Further, it supports the use of validation results in the (re)computation of mapping suggestions and the recommendation of alignment strategies to use.

Our second contribution is the first implemented system that integrates solutions for these challenges in one system (Section 3). It is based on our session-based framework. It deals with efficiency of matching techniques by, in addition to the sessions, avoiding exhaustive pair-wise comparisons between the terms in the different ontologies. It provides solutions to matching with background knowledge by using previous decisions on mapping suggestions as well as using thesauri and domain-specific corpora. Matcher selection, combination and tuning is achieved by using an approach for recommending matchers, combinations and filters. Further, user involvement is supported in the validation phase through user interfaces that have taken into account earlier experiments with ontology engineering systems user interfaces. Also, user decisions are taken into account in the matching and recommendation steps.

Our third contribution are experiments (Section 4) that show the advantages of the session-based approach. They show alignment quality improvements based on the new functionality and show how such a system can be used for evaluating strategies that could not (easily) be evaluated before.

2 Framework

Our framework is presented in Figure 1. The input are the ontologies that need to be aligned. The output is an alignment between the ontologies which consists of a set of mappings that are accepted after validation. When starting an alignment process the user starts a computation session. When a user returns to an alignment process, she can choose to start or continue a computation session or a validation session.

During the *computation sessions* mapping suggestions are computed. The computation may involve preprocessing of the ontologies, matching, and combination and filtering of matching results. Auxiliary resources such as domain knowledge and dictionaries may be used. A reasoner may be used to check consistency of the proposed mapping suggestions in connection with the ontologies as well as among each other. Users may be involved in the choice of algorithms. This is similar to what most ontology alignment systems do. However, in this case the algorithms may also take into account the results of previous validation and recommendation sessions. Further, we allow that computation sessions can be stopped and partial results can be delivered. It is therefore possible for a domain expert to start validation of results before all suggestions are computed. The output of a computation session is a set of mapping suggestions.

During the *validation sessions* the user validates the mapping suggestions generated by the computation sessions. A reasoner may be used to check consistency of the validations. The output of a validation session is a set of mapping decisions (accepted and rejected mapping suggestions). The accepted mapping suggestions form a partial alignment (PA) and are part of the final alignment. The mapping decisions (regarding acceptance as well as rejection of mapping suggestions) can be used in future computation sessions as well as in recommendation sessions. Validation sessions can be stopped and resumed at any time. It is therefore not neccesary for a domain expert to validate all mapping suggestions in one session. The user may also decide not to resume the validation but start a new computation session, possibly based on the results of a recommendation session.

The input for the *recommendation sessions* consists of a database of algorithms for the preprocessing, matching, combination and filtering in the computation sessions. During the recommendation sessions the system computes recommendations for which (combination) of those algorithms may perform best for aligning the given ontologies. When validation results are available these may be used to evaluate the different algorithms, otherwise an oracle may be used. The output of this session is a recommendation for the settings of a future computation session. These sessions are normally run when a user is not validating and results are given when the user logs in into the system again.

Most existing systems can be seen as an instantiation of the framework with one or more computation sessions. Some systems also include one validation session.

3 Implemented System

We have implemented a prototype based on the framework described above. The system includes components from the SAMBO system and newly developed components.

Session Framework. When starting an alignment process for the first time, the user starts a computation session. However, if the user has previously stored sessions, then a screen is shown where the user can start a new session or resume a previous session.

The information about sessions is stored in the session management database. This includes information about the user, the ontologies, the list of already validated mappings suggestions, the list of not yet validated mappings suggestions, and last access date. In the current implementation only validation sessions can be saved. When a computation session is interrupted, a new validation session is created and this can be stored.

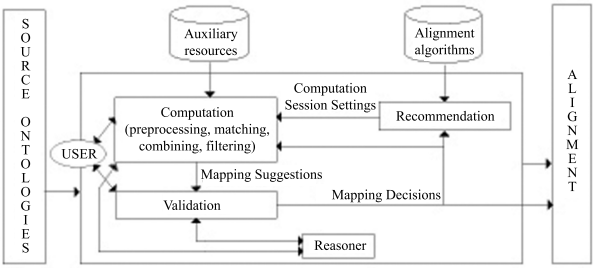


Fig. 1. Framework

Fig. 2. Screenshot: start computation session

Computation. Figure 2 shows a screen shot of the system at the start of a computation session. It allows for the setting of the session parameters. The computation of mapping suggestions uses the following steps. During the *settings selection* the user selects which algorithms to use for the preprocessing, matching, combining and filtering steps. An experienced user may choose her own settings. Otherwise, the suggestion of a recommendation session (by clicking the 'Use recommendations from predefined strategies' button) or a default setting may be used. This information is stored in the session information database.

When a PA is available, the *preprocessing* step partitions the ontologies into corresponding mappable parts that make sense with respect to the structure of the ontologies (details in [11]). Therefore, the matchers will not compute similarity values between all pairs of concepts, but only between concepts in mappable parts, thereby considerably reducing the search space. The user may choose to use this preprocessing step by checking the 'use preprocessed data' check box (Figure 2).

Matchers compute similarity values between terms in different ontologies. Whenever a similarity value for a term pair using a matcher is computed, it is stored in the similarity values database. This can be done during the computation sessions, but also during the recommendation sessions. In the current implementation we have used string matching for matching relations. Regarding concepts, the matchers compute similarity values between pairs of concepts as received from the preprocessing step (all pairs or pairs of concepts in mappable parts). We use the linguistic, WordNet-based, UMLS-based

and instance-based algorithms from the SAMBO system [13]. The matcher *n-gram* computes a similarity based on 3-grams. The matcher *TermBasic* uses a combination of n-gram, edit distance and an algorithm that compares the lists of words of which the terms are composed. The matcher *TermWN* extends TermBasic by using WordNet [20] for looking up is-a relations. The matcher *UMLSM* uses the domain knowledge in the Unified Medical Language System (UMLS, [17]) to obtain mappings. Finally, the instance-based matcher *NaiveBayes* makes use of research literature that is related to the concepts in the ontologies. It is based on the intuition that a similarity measure between concepts in different ontologies can be defined based on the probability that documents about one concept are also about the other concept and vice versa [18].

The user can define which matchers to use in the computation session by checking the check boxes in front of the matchers' names (Figure 2). To guarantee partial results as soon as possible the similarity values for all currently used matchers are computed for one pair of terms at a time and stored in the similarity values database. When the similarity values for each currently used matcher for a pair of terms are computed, they can be combined and filtered (see below) immediately. As ontology alignment is an iterative process, it may be the case that the similarity values for some pairs and some matchers were computed in a previous round. In this case these values are already in the similarity values database and do not need to be re-computed.

Results from different matchers can be *combined*. In our implemented system we allow the choice of a weighted-sum approach or a maximum-based approach. In the first approach each matcher is given a weight and the final similarity value between a pair of terms is the weighted sum of the similarity values divided by the sum of the weights of the used matchers. The maximum-based approach returns as final similarity value between a pair of terms, the maximum of the similarity values from different matchers. The user can choose which combination strategy to use by checking radio buttons (Figure 2), and weights can be added in front of the matchers' names.

Most systems use a threshold *filter* on the similarity values to decide which pairs of terms become mapping suggestions. In this case a pair of terms is a mapping suggestion if the similarity value is equal to or higher than a given threshold value. Another approach that we implemented is the double threshold filtering approach [1] where two thresholds are introduced. Pairs with similarity values equal to or higher than the upper threshold are retained as mapping suggestions. These pairs are also used to partition the ontologies in a similar way as in the preprocessing step. The pairs with similarity values between the lower and upper thresholds are filtered using the partitions. Only pairs of which the elements belong to corresponding elements in the partitions are retained as suggestions. Pairs with similarity values lower than the lower threshold are rejected as mapping suggestions. When a PA is available, a variant of double threshold filtering can be used, where the PA is used for partitioning the ontologies [11]. The user can choose single or double threshold filtering and define the thresholds (Figure 2). Further, to obtain higher quality mappings, we always remove mapping suggestions that conflict with already validated correct mappings [11].

The computation session is started using the 'Start Computation' button. The session can be interrupted using the 'Interrupt Computation' button. The user may also specify beforehand a number of concept pairs to be processed and when this number is reached,

Fig. 3. Screenshot: mapping suggestion

the computation session is interrupted and validation can start. This setting is done using the 'interrupt at' in Figure 2. The output of the computation session is a set of mapping suggestions where the computation is based on the settings of the session. Additionally, similarity values are stored in the similarity values database that can be used in future computation sessions as well as in recommendation sessions. In case the user decides to stop a computation session, partial results are available, and the session may be resumed later on. The 'Finish Computation' button allows a user to finalize the alignment process. (A similar button is available in validation sessions.)

Validation. The validation session allows a domain expert to validate mapping suggestions. The mapping suggestions can come from a computation session (complete or partial results) or be the remaining part of the mapping suggestions of a previous validation session. For the validation we extended the user interface of SAMBO [13] which took into account lessons learned from experiments [9,10] with ontology engineering systems' user interfaces. As stated in [6] our user interface evaluations are one of the few existing evaluations and our system is one of the few systems based on such evaluation. Through the interface, the system presents mapping suggestions (Figure 3) with available information about the terms in the mapping suggestions. When a term appears in multiple mapping suggestions, these will be shown at the same time. The user can accept a mapping suggestion as an equivalence or is-a mapping, or reject the mapping suggestion by clicking the appropriate buttons. Further, the user can give a preferred name to equivalent terms as well as annotate the decisions. The user can also review the previous decisions ('History') as well as receive a summary of the mapping suggestions still to validate ('Remaining Suggestions'). After validation a reasoner is used to detect conflicts in the decisions and the user is notified if any such occur.

The mapping decisions are stored in the mapping decisions database. The accepted mapping suggestions constitute a PA and are partial results for the final output of the ontology alignment system. The mapping decisions (both accepted and rejected) can also be used in future computation and recommendation sessions.

Validation sessions can be stopped at any time and resumed later on (or if so desired - the user may also start a new computation session).

Recommendation. We implemented several recommendation strategies. The first approach (an extension of [19]) requires the user or an oracle to validate all pairs in small

segments of the ontologies. To generate these segments we first use a string-based approach to detect concepts with similar names. The sub-graphs of the two ontologies with the matched concepts as roots are then candidate segments. Among the candidate segments a number of elements (15) of small enough size (60 concepts) are retained as segments. As a domain expert or oracle has validated all pairs in the segments, full knowledge is available for these small parts of the ontologies. The recommendation algorithm then proposes a particular setting for which matchers to use, which combination strategy and which thresholds, based on the performance of the strategies on the validated segments. The advantage of the approach is that it is based on full knowledge of the mappings of parts of the ontologies. An objection may be that good performance on parts of the ontologies may not lead to good performance on the whole ontologies. The disadvantage of the approach is that a domain expert or an oracle needs to provide full knowledge about the mappings of the segments.

The second and third approach can be used when the results of a validation are available. In the second approach the recommendation algorithm proposes a particular setting based on the performance of the alignment strategies on all the already validated mapping suggestions. In the third approach we use the segment pairs (as in the first approach) and the results of earlier validation to compute a recommendation. The advantages of these approaches are that decisions from different parts of the ontologies can be used, and that no domain expert or oracle is needed during the computation of the recommendation. However, no full knowledge may be available for any parts of the ontologies (e.g. for some pairs in the segment pairs, we may not know whether the mapping is correct or not), and validation decisions need to be available.

We note that in all approaches, when similarity values for concepts for certain matchers that are needed for computing the performance, are not yet available, these will be computed and added to the similarity values database.

To define the performance of the alignment algorithms several measures can be used. We define the measures that are used in our implementation. We assume there is a set of pairs of terms for which full knowledge is available about the correctness of the mappings between the terms in the pair. For the first approach this set is the set of pairs in the segments. In the other approaches this set is the set of pairs in the mappings decisions (accepted and rejected). For a given alignment algorithm, let then A be the number of pairs that are correct mappings and that are identified as mapping suggestions, B the number of pairs that are wrong mappings but were suggested, C the number of pairs that are correct mappings but that were not suggested, and D the number of pairs that are wrong mappings and that were not suggested (see Table 1(a)). In A + D cases the algorithm made a correct decision and in B + C cases the algorithm made a wrong decision. In our system we use then the following measures (see Table 1(b)). P^c, R^c and F^c are the common measures of precision, recall and their harmonic mean f-measure. These focus on correct decisions for correct mappings. P^w, R^w and F^w are counterparts that focus on correct decisions regarding wrong mappings. Sim1 is a similarity measure that computes the ratio of correct decisions over the total number of decisions. Sim2 is the Jaccard-similarity where the case of non-suggested wrong mappings is not taken into account (assumed to be a common and non-interesting case).

Table 1. Performance measures

(a) Number of correct/wrong mappings that are suggested/not suggested.

	Suggested	Not suggested
Correct	A	C
Wrong	B	D

(b) Measures.

$P^c = A/(A+B)$, $R^c = A/(A+C)$, $F^c = 2P^cR^c/(P^c+R^c)$
$P^w = D/(C+D)$, $R^w = D/(B+D)$, $F^w = 2P^wR^w/(P^w+R^w)$
$Sim1 = (A+D)/(A+B+C+D)$, $Sim2 = A/(A+B+C)$

The results of the recommendation algorithms are stored in the recommendation database. For each of the alignment algorithms (e.g. matchers, combinations, and filters) the recommendation approach and the performance measure are stored. A user can use the recommendations when starting or continuing a computation session.

4 Experiments

In this Section we discuss experiments that show the advantages of using a session-based system regarding performance of computation of similarity values, filtering and recommendation. Further, the experiments in Sections 4.2-4.3 also show how a session-based system can be used for evaluating PA-based and recommendation algorithms.

Experiments Set-Up. We use the OAEI 2011 Anatomy track for our experiments which contains the ontologies Adult Mouse Anatomy (AMA) and the anatomy part of the NCI Thesaurus (NCI-A). (Removing empty nodes in the files) AMA contains 2737 concepts and NCI-A contains 3298 concepts. This gives 9,026,626 pairs of concepts. Further, a reference alignment containing 1516 equivalence mappings is available.

Regarding the alignment strategies, we used the following. As matchers we used *n-gram*, *TermBasic*, *TermWN*, *UMLSM* and *NaiveBayes*.[1] As combination strategies we used weighted sum with possible weights 1 and 2 as well as the maximum-based approach. Further, we used the single and double threshold strategies with threshold values 0.3, 0.4, 0.5, 0.6, 0.7 and 0.8. In total this gives us 4872 alignment strategies. For each of these strategies we computed P^c, R^c, F^c, P^w, R^w, F^w, Sim1 and Sim2 based on the OAEI reference alignment. For instance, Table 2, shows the top 10 strategies with respect to Sim2. All these 10 strategies use a weighted-sum combination, double threshold and include *UMLSM* and at least one string matching-based matcher. These strategies have also a very high F^w of over 0.99. The top 10 strategies with respect to R^c all include *UMLSM* and at least one of *n-gram* or *TermWN*. All these strategies use a maximum-based combination approach, single threshold and, as expected, a low threshold (0.3). The best strategies find 1497 correct mapping suggestions. The highest P^c for these strategies is, however, less than 0.016. When sorting strategies based on P^c,

[1] For *NaiveBayes* we generated a corpus of PubMed abstracts. We used a maximum of 100 abstracts per concept. For AMA the total number of documents was 30,854. There were 2413 concepts for which no abstract was found. For NCI-A the total number of documents was 40,081. There were 2886 concepts for which no abstract was found.

Table 2. Top 10 strategies for F^c and Sim2

matchers	weights	threshold	correct suggestions	wrong suggestions	F^c	Sim2
TermBasic;*UMLSM*	1;1	0.4;0.7	1223	101	0.8612	0.7563
TermWN;*UMLSM*;*NaiveBayes*;*n-gram*	1;2;2;1	0.3;0.5	1223	101	0.8612	0.7563
n-gram;*TermBasic*;*UMLSM*	1;1;2	0.5;0.8	1192	63	0.8603	0.7549
n-gram;*UMLSM*	1;1	0.5;0.8	1195	67	0.8603	0.7548
UMLSM;*NaiveBayes*;*TermWN*	2;1;2	0.4;0.6	1203	78	0.8602	0.7547
UMLSM;*NaiveBayes*;*n-gram*;*TermBasic*	2;1;1;1	0.4;0.6	1199	73	0.8601	0.7545
n-gram;*TermBasic*;*UMLSM*	1;2;2	0.5;0.8	1181	50	0.8598	0.7541
UMLSM;*NaiveBayes*;*TermBasic*	2;1;2	0.4;0.6	1194	68	0.8596	0.7537
UMLSM;*NaiveBayes*;*n-gram*;*TermBasic*	2;2;1;1	0.3;0.5	1221	104	0.8595	0.7537
UMLSM;*NaiveBayes*;*TermBasic*	2;1;1	0.5;0.6	1187	60	0.8592	0.7531

Table 3. Three alignment strategies

strategy	matchers	weights	threshold	suggestions	F^c	Sim2
AS1	*TermBasic*;*UMLSM*	1;1	0.4;0.7	1324	0.86	0.75
AS2	*TermWN*;*n-gram*;*NaiveBayes*	2;1;1	0.5	1824	0.65	0.48
AS3	*n-gram*;*TermBasic*;*UMLSM*	1;1;2	0.3	4061	0.48	0.32

528 strategies had maximum P^c value of 1. All of these strategies include *NaiveBayes*. Six of the strategies are single matcher strategies (*NaiveBayes* with thresholds 0.6, 0.7, 0.8, 0.6;07, 0.6;0.8 and 0.7;0.8). No strategy has threshold 0.3. Among those strategies the maximum amount of correct mapping suggestions that are found is 259. All 528 strategies have $R^w = 1$ and $P^w > 0.99$. They have high Sim1 values and low Sim2 values. With respect to the other measures, i.e. R^w, P^w, F^w and Sim1, the strategies do not show much variation. Therefore, in the remainder of this paper, we mainly discuss results with respect to F^c and Sim2. F^c is a standard measure; Sim2 has a high correlation to F^c, but has a higher degree of differentiation in our experiments.

For the experiments in Sections 4.2 and 4.3 we chose three alignment strategies (Table 3) as a basis for discussion. Strategy AS1 uses a weighted sum combination of *TermBasic* with weight 1 and *UMLSM* with weight 1, and as double thresholds 0.4;0.7. This information is presented in columns 2-4 in Table 3. AS1 generates 1324 mapping suggestions (column 5). AS1 is the strategy with best F^c (0.86) and with best Sim2 (0.75) values. AS2 is an average strategy regarding F^c (0.65) and Sim2 (0.48). It uses a weighted sum combination of *TermWN* with weight 2, *n-gram* with weight 1 and *NaiveBayes* with weight 1, and as threshold 0.5. It generates 1824 mapping suggestions. AS3 performs poorly for F^c (0.48) and Sim2 (0.32), but has a high R^c value (0.89). It uses a weighted sum combination of *n-gram* with weight 1, *TermBasic* with weight 1, and *UMLSM* with weight 2, and as threshold 0.3. It generates 4061 mapping suggestions.

4.1 Computation of Similarity Values

For each of the matchers we computed the similarity values for all pairs of concepts. When a similarity value is computed it is stored in the similarity values database.

Table 4. Matcher computation time (in mins)

	n-gram		*NaiveBayes*	
number of pairs	without previous values stored	with previous values stored	without previous values stored	with previous values stored
902,662	2.59		196.15	
1,805,324	5.08	3.98	149.95	84.05
4,513,310	12.73	10.78	418.49	265.87
6,769,965	19.19	13.83	645.71	212.35
9,026,626	25.85	17.32	790.74	207.64

Previous approaches could not take advantage of previously stored values. However, computation sessions in a session-based approach can take advantage of the fact that previous computation sessions already stored similarity values. In Table 4 we show for two of the matchers the computation times for when previous values were stored and for when no previous values were stored. We do this for the computation of 10%, 20% (of which 10% stored), 50% (of which 20% stored), 75% (of which 50% stored) and 100% (of which 75% stored) of the 9,026,626 pairs. For instance, for *n-gram* the computation and storage of 902,662 similarity values took 2.59 minutes. The computation and storage of 1,805,324 similarity values from scratch took 5.08 minutes. However, assuming 902,662 similarity values are already stored and checking the database, it will take 3.98 minutes. Using the database is advantageous for string matchers, and even more advantageous for more complex matchers for which the speed-up may be up to 25%. The session-based approach leads therefore to reduced computation times and reduced waiting times for the domain expert.

4.2 Using the Validation Decisions from Previous Sessions for Filtering

There are few approaches that can take into account already given mappings. Further, it is not common that such a set a pre-existing mappings exists. In a session-based approach, however, every validation session generates such sets, which can be used to improve the quality of the mapping suggestions and reduce unnecessary user interaction. Further, the knowledge of the domain expert is taken into account in an early stage.

Filtering Using Validated Correct Mappings. Table 5 shows for the strategies AS1, AS2 and AS3 the reduction of the number of suggestions by using the filter strategy that removes mapping suggestions that are in conflict with already validated correct mappings. It shows the number of removed mapping suggestions after 500, 1000 and 1300 processed mapping suggestions. The results show that AS1 does not produce many mapping suggestions that would conflict. They also suggest that the removal should be done as soon as possible. For instance, when we would process 1000 suggestions without removal, the 156 that would be removed after 500 processed suggestions may actually have been - unnecessarily - validated by the user. Therefore, in our system we perform the removal after every validation of a correct equivalence mapping and thereby reduce unnecessary user interaction. We also remind that the strategies AS1, AS2 and AS3 produce 1365, 1824 and 4061 mapping suggestions, respectively. There-

Table 5. Filter using validated correct mappings

processed	AS1	AS2	AS3
500	20	107	156
1000	26	58	288
1300	4	20	20

Table 6. Double threshold filter using validated correct mappings

processed	AS1 suggestions removed	AS2 suggestions removed	AS3 suggestions removed	AS1 correct removed	AS2 correct removed	AS3 correct removed
500	0/2	134/113	244/279	0/0	12/1	9/1
1000	1/0	52/47	532/470	1/0	1/0	22/4
1300	0/2	43/35	443/276	0/0	9/2	21/3

fore, having processed 1000 mapping suggestions means that 73%, 40% and 25% of the suggestions have been processed for AS1, AS2 and AS3, respectively.

Double Threshold Filtering Using Validated Correct Mappings. In our second experiment, once a session is locked, we use double threshold filtering with thresholds 0.3 (lowest considered threshold) and 0.6 on the remaining unvalidated mapping suggestions of that session. Table 6 shows for the strategies AS1, AS2 and AS3 the total number of mapping suggestions (columns 2-4) and the number of correct suggestions (columns 5-7) that are removed by this operation. There are two values separated by '/'. As double threshold filtering heavily relies on the structure of the ontologies and many is-a relations are actually missing in AMA and NCI-A [12], we experimented with the original ontologies (first value) and the repaired ontologies (second value). The results show that this filtering has a positive effect on F^c. Further, in most cases more mapping suggestions, but also more correct suggestions are removed in the original ontologies than in the repaired ontologies, and the quality in terms of F^c is higher for the repaired ontologies. We also note that for the best strategy the effect is not that high.

4.3 Recommendation Strategies with and without Sessions

For the recommendation experiments we used Sim2 as recommendation measure. For some of the experiments we also needed to generate segment pairs. The system generated 94 segment pair candidates of which 15 were randomly chosen as segment pairs. The maximum number of concepts in a segment is 12 and the minimum number is 3. The total number of concept pairs for all 15 segment pairs together is 424. According to the reference alignment of the OAEI, 46 of those are correct mappings. The maximum number of correct mappings within a segment pair is 7 and the minimum is 1.

Session-Based Recommendation Using Validation Decisions Only. In this experiment we use the recommendation algorithm that computes a performance measure for the alignment strategies based on how the strategies perform on the already validated mapping suggestions. Table 7, rows 'rec1', show the recommended strategies together with their F^c value on the current validation decisions and their actual F^c value, after

having processed 500/503[2], 1000, ..., 4000 suggestions for AS1 and AS3, respectively. For AS1, AS1 itself does not appear among the top 10 recommendations for all the sessions. The strategies that received the best score for 500, 1000 and 1300 processed suggestions have actual F^c values of 0.18, 0.85 and 0.23 respectively. The results are explained by the consistent group in the double threshold filtering. For AS3, the strategy that receives the best score after 1000, 2000 and 2500 processed suggestions is also the best strategy (AS1) in reality. Otherwise, AS1 is within the top 10 recommendations. In these cases AS1 is not recommended because it suggests 2, 1, 13, 6 and 48 wrong mapping suggestions for 503, 1500, 3000, 3500 and 4000 processed suggestions, respectively, which are not suggested by the recommended strategy. The reason for the better performance of the recommended strategy is due to the generated consistent group in the double threshold filtering. We note that the recommended strategy always has an actual $F^c \geq 0.85$ (with best 0.861 for AS1).

Further, we performed an experiment where a recommendation was computed after every 500 validations and every time the recommended strategy was used. We noted that usually the recommendations improved. For instance, when using the recommended strategy after 500 validations for AS1 for computing the next 500 suggestions, leads to an improved recommendation after the 500 new suggestions are validated.

Session-Based Recommendation Using Segment Pairs and Validation Decisions. In this experiment we use the recommendation algorithm that uses segment pairs and computes a performance measure for the alignment strategies based on how the strategies perform on the already validated parts of the segment pairs. Table 7, rows 'rec2', show the results for AS1 and AS3, respectively. For AS1, the recommended strategy after 500, 1000 and 1300 processed suggestions has actual $F^c = 0.07$. The reason for this result is that AS1 has very high precision so the oracle (validated suggestions) has very little information about wrong mapping suggestions. However, it has much information about correct mapping suggestions. The strategy that is recommended in the three sessions is one that has very high recall but that also suggests many wrong mapping which the algorithm cannot detect. For AS3, the strategies that are recommended after 503, 1000, 1500, 2000 and 2500 processed suggestions have actual $F^c = 0.53$, after 3000 actual $F^c = 0.76$, and after 3500 and 4000 actual $F^c = 0.82$. This result shows that as the number of processed suggestions increases, the recommended strategy becomes better. This is because the quality of the oracle increases.

Session-Independent Recommendation Using Segment Pairs and Oracle. In this experiment we use the recommendation algorithm that uses segment pairs and computes a performance measure for the alignment strategies based on how the strategies perform on the segment pairs. This requires an oracle that has full knowledge about the mappings in the segment pairs and for this we use the reference alignment as provided by the OAEI. As this recommendation strategy is independent from the actual validation decisions, the recommendation does not change during the alignment process. It can therefore be performed in the beginning. Based on the performance on the 15 small segments pairs (with a reference alignment of only 46 mappings), the recommendation

[2] 503, because the validation decision for suggestion 500 removes other suggestions.

Table 7. Recommendations for AS1 and AS3

	processed suggestions	rec	matchers	weights	threshold	rec F^c	actual F^c
AS1	500	rec1	*NaiveBayes*;*n-gram*;*TermBasic*;*TermWN*	1;1;2;1	0.3;0.6	0.993	0.186
		rec2	*NaiveBayes*;*n-gram*	1;1	0.3;0.8	1	0.070
	1000	rec1	*TermBasic*;*TermWN*;*UMLSM*;*NaiveBayes*	2;1;2;1	0.5;0.7	0.992	0.850
		rec2	*NaiveBayes*;*n-gram*	1;1	0.3;0.8	1	0.070
	1300	rec1	*n-gram*;*TermBasic*;*TermWN*;*UMLSM*	1;1;2;1	0.3;0.7	0.972	0.235
		rec2	*NaiveBayes*;*n-gram*	1;1	0.3;0.8	1	0.070
AS3	503	rec1	*n-gram* ;*TermBasic*;*UMLSM*	1;1;2	0.4;0.8	0.920	0.850
		rec2	*n-gram* ;*TermBasic*;*TermWN*;*UMLSM*	1;1;1;2	0.3;0.5	1	0.530
	1000	rec1	*TermBasic*;*UMLSM*	1;1	0.4;0.7	0.950	0.861
		rec2	*n-gram* ;*TermBasic*;*TermWN*;*UMLSM*	1;1;1;2	0.3;0.5	1	0.530
	1500	rec1	*TermBasic*;*UMLSM*;*TermWN*	1;2;1	0.4;0.7	0.940	0.860
		rec2	*n-gram* ;*TermBasic*;*TermWN*;*UMLSM*	1;1;1;2	0.3;0.5	1	0.530
	2000	rec1	*TermBasic*;*UMLSM*	1;1	0.4;0.7	0.920	0.861
		rec2	*n-gram* ;*TermBasic*;*TermWN*;*UMLSM*	1;1;1;2	0.3;0.5	1	0.530
	2500	rec1	*TermBasic*;*UMLSM*	1;1	0.4;0.7	0.920	0.861
		rec2	*n-gram* ;*TermBasic*;*TermWN*;*UMLSM*	1;1;1;2	0.3;0.5	1	0.530
	3000	rec1	*UMLSM*;*TermWN*	1;1	0.4;0.7	0.920	0.860
		rec2	*n-gram* ;*TermBasic*;*TermWN*;*UMLSM*; *NaiveBayes*	1;1;1;2;1	0.3;0.7	1	0.760
	3500	rec1	*UMLSM*;*NaiveBayes*;*n-gram* ;*TermBasic*	2;2;1;1	0.3;0.5	0.920	0.860
		rec2	*TermBasic*;*TermWN*;*UMLSM*;*NaiveBayes*	1;2;2;1	0.3;0.6	1	0.820
	4000	rec1	*n-gram* ;*TermBasic*;*UMLSM*	1;1;2	0.5;0.8	0.920	0.860
		rec2	*TermBasic*;*TermWN*;*UMLSM*;*NaiveBayes*	1;2;2;1	0.3;0.6	0.990	0.820

algorithm gives Sim2 = 0.87 and F^c = 0.93 for AS1, Sim2 = 0.52 and F^c = 0.68 for AS2, and Sim2 = 0.47 and F^c = 0.64 for AS3.

However, there are also 145 strategies that have a higher Sim2 value than AS1. The top 8 recommended strategies all use double threshold filtering and have Sim2 = 0.98 and F^c = 0.99 for the segment pairs, and an actual F^c between 0.8 and 0.84. They suggest 45 correct mappings and 0 wrong mappings, whereas AS1 suggests 42 correct mappings and 2 wrong mappings. We also note that that there are 81 strategies which have Sim2 $>$0.9 and F^c $>$0.95 on the segment pairs.

5 Related Work

To our knowledge there is no other framework or system that deals with all the challenges for aligning large ontologies that our approach deals with. Many systems generate mapping suggestions and can be seen as covering a computation session. This is also what is evaluated at the OAEI. There are some systems that allow validation of mappings such as SAMBO [13], COGZ [7] for PROMPT, and COMA++ [2]. None of these systems allow, however, interruptable sessions. LogMap2 [8] allows user interaction although it does not have graphical user interfaces yet. Interrupting user interaction in this case means using heuristics to deal with remaining mapping suggestions.

Regarding the computation session components of our system, many matchers have been proposed (e.g. many papers on http://ontologymatching.org/). There are some approaches that reduce the search space by segmenting or partitioning the ontologies [2,16]. The main difference with our approach is that we use validation decisions to partition the ontologies. Our combination strategies are standard strategies. Most systems use single threshold filtering, while we also allow double threshold filtering. There are very few recommendation approaches. The approach in [3] proposes a machine learning approach to optimize alignment strategies and is complementary to our approach. Further, there are approaches that try to minimize user interaction (e.g. [14]).

6 Conclusion

In this paper we presented to our knowledge the first framework and implemented system that allows a user to interrupt and resume the different stages of the ontology alignment task. Our work addressed several of the challenges in ontology alignment [15].

Further, we showed the usefulness of the system and its components through experiments with many alignment strategies on the OAEI 2011 Anatomy track ontologies. We showed that we obtain better quality suggestions using the session-based approach. For instance, one of the lessons learned from the experiments is that filtering after the locking of sessions is useful and the worse the initial strategy, the more useful this is. Better quality suggestions are also achieved through the use of validated mappings in the preprocessing phase. In all these cases domain expert knowledge is taken into account through the validated mappings. We also showed that the use of the session-based approach reduces unnecessary user interaction. Further, the recommendation is important, especially when the initial strategy is not good. It is also clear that the approaches using validation decisions, become better the more suggestions are validated. For the approaches using segment pairs, the choice of the segment pairs influences the recommendation results (which is different from the conclusions of experiments in [19]).

We note that the session-based framework enabled experimentation and evaluation of new alignment approaches (both in computation and recommendation) that are based on validation decisions. These evaluations were not possible or cumbersome before.

In future work we will continue to develop and evaluate computation strategies and recommendation strategies. Especially interesting are strategies that reuse validation results to e.g. reduce the search space or guide the computation. Further, we will investigate new strategies for recommendations using validation decisions.

Acknowledgements. We thank Qiang Liu, Muzammil Zareen Khan and Shahab Qadeer for their implementation work on earlier versions of the system.

References

1. Chen, B., Lambrix, P., Tan, H.: Structure-based filtering for ontology alignment. In: IEEE WETICE Workshop on Semantic Technologies in Collaborative Applications, pp. 364–369 (2006)
2. Do, H.-H., Rahm, E.: Matching large schemas: approaches and evaluation. Information Systems 32, 857–885 (2007)

3. Ehrig, M., Staab, S., Sure, Y.: Bootstrapping ontology alignment methods with APFEL. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 186–200. Springer, Heidelberg (2005)
4. Euzenat, J., Meilicke, C., Stuckenschmidt, H., Shvaiko, P., Trojahn, C.: Ontology alignment evaluation initiative: Six years of experience. In: Spaccapietra, S. (ed.) Journal on Data Semantics XV. LNCS, vol. 6720, pp. 158–192. Springer, Heidelberg (2011)
5. Euzenat, J., Schvaiko, P.: Ontology Matching. Springer (2007)
6. Falconer, S., Noy, N.: Interactive techniques to support ontology matching. In: Schema Matching and Mapping, pp. 29–52 (2011)
7. Falconer, S.M., Storey, M.-A.: A cognitive support framework for ontology mapping. In: Aberer, K., et al. (eds.) ISWC/ASWC 2007. LNCS, vol. 4825, pp. 114–127. Springer, Heidelberg (2007)
8. Jimenez-Ruiz, E., Cuenca-Grau, B., Zhou, Y., Horrocks, I.: Large-scale interactive ontology matching: Algorithms and implementation. In: 20th European Conference on Artificial Intelligence, pp. 444–449 (2012)
9. Lambrix, P., Edberg, A.: Evaluation of ontology merging tools in bioinformatics. In: Pacific Symposium on Biocomputing, pp. 589–600 (2003)
10. Lambrix, P., Habbouche, M., Perez, M.: Evaluation of ontology development tools for bioinformatics. Bioinformatics 19(12), 1564–1571 (2003)
11. Lambrix, P., Liu, Q.: Using partial reference alignments to align ontologies. In: Aroyo, L., et al. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 188–202. Springer, Heidelberg (2009)
12. Lambrix, P., Liu, Q., Tan, H.: Repairing the missing is-a structure of ontologies. In: Gómez-Pérez, A., Yu, Y., Ding, Y. (eds.) ASWC 2009. LNCS, vol. 5926, pp. 76–90. Springer, Heidelberg (2009)
13. Lambrix, P., Tan, H.: SAMBO - a system for aligning and merging biomedical ontologies. Journal of Web Semantics 4(3), 196–206 (2006)
14. Rodler, P., Shchekotykhin, K., Fleiss, P., Friedrich, G.: Rio: Minimizing user interaction in debugging of aligned ontologies. In: 7th International Workshop on Ontology Matching, pp. 49–60 (2012)
15. Schvaiko, P., Euzenat, J.: Ontology matching: state of the art and future challenges. IEEE Transactions on Knowledge and Data Engineering 25(1), 158–176 (2013)
16. Hanif Seddiqui, M., Aono, M.: An efficient and scalable algorithm for segmented alignment of ontologies of arbitrary size. Journal of Web Semantics 7(4), 344–356 (2008)
17. Unified Medical Language System, `http://www.nlm.nih.gov/research/umls/about_umls.html`
18. Tan, H., Jakonienė, V., Lambrix, P., Aberg, J., Shahmehri, N.: Alignment of biomedical ontologies using life science literature. In: Bremer, E.G., Hakenberg, J., Han, E.-H(S.), Berrar, D., Dubitzky, W. (eds.) KDLL 2006. LNCS (LNBI), vol. 3886, pp. 1–17. Springer, Heidelberg (2006)
19. Tan, H., Lambrix, P.: A method for recommending ontology alignment strategies. In: Aberer, K., et al. (eds.) ISWC/ASWC 2007. LNCS, vol. 4825, pp. 494–507. Springer, Heidelberg (2007)
20. WordNet, `http://wordnet.princeton.edu/`

Organizing Ontology Design Patterns as Ontology Pattern Languages

Ricardo de Almeida Falbo, Monalessa Perini Barcellos,
Julio Cesar Nardi, and Giancarlo Guizzardi

Ontology and Conceptual Modeling Research Group (NEMO), Computer Science Department,
Federal University of Espírito Santo, Vitória, Brazil
{falbo,monalessa,jnardi,gguizzardi}@inf.ufes.br

Abstract. Ontology design patterns have been pointed out as a promising approach for ontology engineering. The goal of this paper is twofold. Firstly, based on well-established works in Software Engineering, we revisit the notion of ontology patterns in Ontology Engineering to introduce the notion of ontology pattern language as a way to organize related ontology patterns. Secondly, we present an overview of a software process ontology pattern language.

Keywords: ontology design patterns, ontology pattern language, software process ontology.

1 Introduction

Although nowadays ontology engineers are supported by a wide range of ontology engineering methods and tools, building ontologies is still a difficult task even for experts. In this context, reuse is pointed out as a promising approach for ontology engineering. Ontology reuse allows speeding up the ontology development process, saving time and money, and promoting the application of good practices [1]. However, ontology reuse in general is a hard research issue, and one of the most challenging and neglected areas of ontology engineering [2]. The problems of selecting the right ontologies for reuse, specializing them, and composing several ontology fragments are not properly addressed yet [3].

Ontology Design Patterns (ODPs) are an emerging approach that favors the reuse of encoded experiences and good practices. ODPs are modeling solutions to solve recurrent ontology development problems [4]. Experiments, such as the ones conducted by Blomqvist et al. [3], show that ontology engineers perceive ODPs as useful, and that the quality and usability of the resulting ontologies are improved. However, compared with Software Engineering, where patterns have been used for a long period, patterns in Ontology Engineering are still in infancy. The earliest works addressing the issue of patterns in Ontology Engineering are from the beginning of the 2000s (e.g. [5]), and only recently this approach has gained more attention in this area [1, 2, 3, 4] and in the Semantic Web area [6].

A striking feature of the current use of patterns in Ontology Engineering is that they are generally being applied as stand-alone entities. However, as pointed out by

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 61–75, 2013.

Alexander and colleagues in their pioneering work [7], each pattern can exist only to the extent that it is supported by other patterns. This is especially important to ontology patterns that are related to a specific domain.

Although many ODPs in the literature refer to others, most of these references fail to give more complete guidelines on how the patterns can be combined to form solutions to larger problems. Contexts and problem descriptions are usually stated as general as possible, so that each pattern can be applied in a wide variety of situations. In addition, solution descriptions tend to focus on applying the patterns in isolation, and do not properly address issues that arise when multiple patterns are applied in overlapping ways, such as the order in which they can be applied. This situation is problematic, since the features introduced by applying one pattern may be required by the next. A larger context is therefore needed to describe the larger problems that can be solved by combining patterns, and to address issues that arise when patterns are used in combination. This context can be provided by what in Software Engineering has been termed a *Pattern Language* [8].

It is important to highlight that we borrowed the term "pattern language" from Software Engineering (SE), where patterns have been studied and applied for a long time. A pattern language, in a SE view, *is a network of interrelated patterns that defines a process for systematically solving coarse-grained software development problems* [8, 9]. Thus, we are not actually talking about a language properly speaking. In "pattern language", the use of the term "language" is, in fact, a misnomer, given that a pattern language does not typically define *per se* a grammar with an explicit associated mapping to a semantic domain. However, if we focus on a more general concept of a *representation system*, we can consider the constituent patterns as an alphabet of higher-granularity primitives. Moreover, in this case, we can consider the procedural rules prescribing how these primitives can be lawfully combined as defining a set of valid possible instantiations for that representation system.

That all said, perhaps a more appropriate name would be a *"Pattern System"*. In any case, since we intend to reuse notions well-established in SE to apply them in Ontology Engineering as well as connect to the tradition in that area, we decided to keep here the term "pattern language". Thus, we define Ontology Pattern Language (OPL) as a network of interrelated domain-related ontology patterns that provides holistic support for solving ontology development problems for a specific domain.

An OPL contains a set of interrelated domain-related ontology patterns, plus a process providing explicit guidance on what problems can arise in that domain, informing the order to address these problems, and suggesting one or more patterns to solve each specific problem. It is worthwhile to point out that, although an OPL provides a process describing how to use the patterns to address problems related to a specific domain, an OPL is not a method for building ontologies. It only deals with reuse in ontology development, and its guidance can be followed by ontology engineers using whatever ontology development method that considers ontology reuse as one of its activities.

According to Schmidt et al. [10], the trend in the SE patterns community is towards defining pattern languages, rather than stand-alone patterns. We advocate this should also be taken into account in Ontology Engineering, mainly for a class of ontologies

called *Core Ontologies*. Core ontologies provide a precise definition of structural knowledge in a specific field that spans across different application domains in this field [11]. Thus, we argue that core ontologies are good candidates to be presented as ontology pattern languages.

In summary, the contribution of this paper is to incorporate ideas from patterns as used in Software Engineering to patterns in Ontology Engineering. Firstly, based on well-established works in Software Engineering, such as [9], we revisit the notion of ontology patterns in Ontology Engineering, and introduce the notion of Ontology Pattern Language as a way to organize domain-related ontology patterns. Secondly, we present a particular ontology pattern language in the Software Process domain.

This paper is organized as follows. In Section 2, we present pattern-related concepts, mainly as used in Software Engineering. In Section 3, we discuss ontologies focusing on their generality level. This discussion is important in the context of this paper to point out which is the generality level that we believe to be the most appropriate to build OPLs. In Section 4, we discuss ontology patterns and we introduce the notion of Ontology Pattern Language. In Section 5, we briefly present the Software Process Ontology Pattern Language (SP-OPL), and an example showing its use for building a fragment of a measurement process ontology. Section 6 discusses related works. Finally, in Section 7, we present the final considerations of the paper.

2 On Patterns and Pattern Languages

Patterns are vehicles for encapsulating knowledge. They are considered one of the most effective means for naming, organizing, and reasoning about design knowledge. "Design knowledge" here is employed in a general sense, meaning design in several different areas, such as Architecture and Software Engineering (SE). According to Buschmann et al. [9], "a pattern describes a particular recurring design problem that arises in specific design contexts and presents a well-proven solution for the problem. The solution is specified by describing the roles of its constituent participants, their responsibilities and relationships, and the ways in which they collaborate".

In SE, there are several types of patterns. The best known are analysis patterns, design patterns and idioms. An analysis pattern is a pattern that describes how to model a particular kind of problem in an application domain. A design pattern provides a scheme for refining elements of a software system or the relationships between them. An idiom is a pattern specific to a programming language or environment. An idiom describes how to implement particular behavior or structures in code using the features of the given language or environment [9].

Patterns are often considered and applied separately. However, no pattern is an island. Contrariwise, patterns are fond of company: sometimes with one pattern as an alternative to another, sometimes with one pattern as an adjunct to another, sometimes with a number of patterns bound together as a tightly-knit group. The manifold relationships that can exist between patterns help to strengthen and extend the power of an individual pattern beyond its specific focus [9].

A pattern language is a set of patterns and relationships among them that can be used to systematically solve coarse-grained problems [8]. A pattern language defines a process that aims to provide holistic support for using the patterns to address problems related to a specific technical or application domain. This holistic view should provide explicit guidance on what problems can arise in the domain, inform the order to address them, and suggest one or more patterns to solve each specific problem [9]. A pattern language should also provide guidelines showing how the patterns can be composed to form solutions to problems [8]. The patterns in a pattern language are usually designed to be used within the context of the language. Therefore, they tend to be tightly coupled, and it is difficult or even impossible to use them in isolation [8].

3 Ontologies and Their Generality Levels

There are different classifications of ontologies in the literature. In the context of this work, we are interested in the one that classifies ontologies according to their generality levels, discriminating between foundational, core and domain ontologies [11].

At the highest level of generality, there are the foundational ontologies. Foundational ontologies span across many fields and model the very basic and general concepts and relations that make up the world, such as object, event, parthood relation etc. [12, 13, 14]. Domain ontologies, in turn, describe the conceptualization related to a given domain, such as electrocardiogram in medicine [12]. With a level of generality between that of foundational and domain ontologies, there are core ontologies. Core ontologies provide a precise definition of structural knowledge in a specific field that spans across different application domains in this field. These ontologies are built based on foundational ontologies and provide a refinement to them by adding detailed concepts and relations in their specific field [11].

Guizzardi [15] makes an important distinction between ontologies as conceptual models, known as *reference ontologies*, and ontologies as coding artifacts, called here *operational ontologies*. A reference domain ontology is constructed with the goal of making the best possible description of the domain in reality. It is a special kind of conceptual model, an engineering artifact with the additional requirement of representing a model of consensus within a community [15]. On the other hand, once users have already agreed on a common conceptualization, operational versions of a reference ontology can be created. Contrary to reference ontologies, operational ontologies are designed with the focus on guaranteeing desirable computational properties.

Although we agree with Scherp et al.'s classification for ontologies [11], we perceive them as a continuum, ranging from pure foundational ontologies, such as DOLCE [13] and UFO (Parts A [14] and B [16]), to domain ontologies. In our view, there can be different levels of generality in ontologies that are classified as, for instance, core ontologies. In [11], for example, three core ontologies are presented: Event-Model-F provides a formal representation of the different aspects of events in which humans participate; The Core Ontology on Multimedia (COMM) describes arbitrary digital media data; The Cross-Context Semantic Information Management Ontology

(X-COSIMO) allows representing the communication taking place between different persons and systems and the information associated with it. Although all three are built based on DOLCE and classified as core ontologies, in our opinion, Event-Model-F is more general than COMM and X-COSIMO, since the last two address conceptualizations that are closer to a domain conceptualization (multimedia and personal information management, respectively) than the former (events for representing human experience).

We have experienced such situations when developing ontologies for the software process domain. Originally, we classified our Software Process Ontology (SPO) [16, 17] as a reference domain ontology. However, it has been used as basis for developing other reference domain ontologies related to specific software processes, such as the measurement process (Reference Software Measurement Ontology (RSMO) [18]). The latest version of SPO [17] is grounded in UFO-C, an ontology of social entities [16]. In [16], UFO-C is classified as a foundational ontology, but it builds on top of UFO-A (an ontology of endurants) and UFO-B (an ontology of events) to systematized social concepts such as action, goal, agent, commitment, among others.

In the light of the above, we see those categories of ontologies (foundational, core and domain ontologies) as regions in a spectrum with fuzzy boundaries between them. Figure 1 illustrates this continuous view using the aforementioned ontologies. DOLCE, UFO-A and UFO-B are genuine foundational ontologies. UFO-C and Event-Model-F are in the frontier between foundational and core ontologies. X-COSIMO, COMM and SPO are core ontologies, but the last is in the region closer to domain ontologies. Finally, RSMO is classified as a domain ontology.

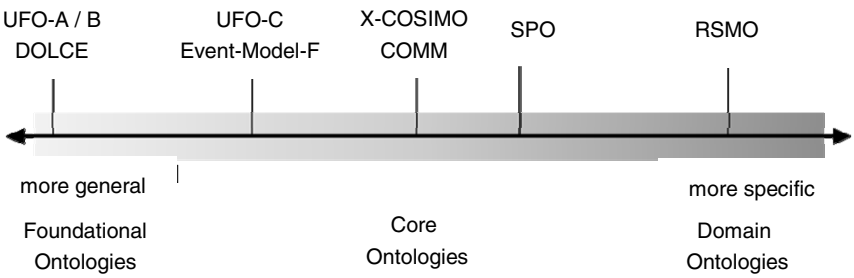


Fig. 1. Ontology level of generality as a continuum

In this paper we are interested in core ontologies, mainly those that are in a region closer to domain ontologies. Ontologies in this region, although general enough to be specialized when applied to more restrict domains, are still domain-related. We claim that these core ontologies should be presented as ontology pattern languages. Moreover, we are interested in patterns to support the development of reference domain ontologies [15], which are to be reused in the conceptualization phase. In the next section, we present a fuller argumentation defending our view that patterns defined in the level of Core Ontologies are the ones which can be most appropriately defined as a Pattern System or Pattern Language.

4 Ontology Design Patterns and Ontology Pattern Languages

According to Gangemi and Presutti [2], an Ontology Design Pattern (ODP) is a modeling solution to solve a recurrent ontology design problem. ODPs can be of different types, such as content, logical, architectural, and so on. Content Ontology Patterns (COPs) refer to small fragments of ontology conceptual models, and must be language-independent [2]. A COP can extract a fragment of either a foundational or a core ontology, which constitutes its background [19]. Thus, we consider two types of COPs: Foundational (FOPs) and Domain-related ontology patterns (DROPs).

Since FOPs are COPs extracted from foundational ontologies, they tend to be more generally applied. Although they certainly have dependencies with other patterns, these dependencies tend to be weaker, and the pattern is easily applied in isolation. Take the example discussed in [1] for the development of a context ontology network called mIO!. The reused patterns were selected among ODPs present in catalogues such as the one available in the `ontologydesignpatterns.org` portal. The reused patterns were related to general (formal) problems, such as taxonomical or part-whole relations, n-ary relations/participation. All the reused patterns are FOPs. None of the examples there are of DROPs.

In contrast, DROPs for a specific domain are very inter-related, and it is very difficult (if not impossible) to apply them in isolation. It is important to highlight, nonetheless, that as patterns move closer to a Domain ontology, they agglutinate to form a *stable model*, i.e., the constraints on how they can be inter-related become so strong that the very domain model is practically the only way they can appear together, thus, lacking the *potential for recurrence* which is part of the very definition of what a pattern is. That is why we advocate that DROPs occurring at the level of Core Ontologies are the best candidates for being organized as ontology pattern languages.

Regarding the way they are documented and communicated, COPs, in general, are comparable to design patterns in Software Engineering [2]. On the other hand, regarding their contents, DROPs are comparable to Software Engineering analysis patterns.

COPs should be encoded in a higher-order representation language [2]. OntoUML [14] is an example of an ontology representation language that is suitable for this purpose. OntoUML is a UML profile that enables modelers to make finer-grained modeling distinctions between different types of classes and relations according to ontological distinctions put forth by UFO-A. Thus, we advocate for the use of OntoUML as a modeling language for DROPs in an OPL. On the other hand, Gangemi and Presutti [2] state that "a (sample) representation in OWL is needed in order to (re)use the patterns as building blocks over the Semantic Web". We agree that an example in OWL could be useful, but it is not a requisite for DROPs. DROPs are to be reused in the conceptualization phase. If they have a counterpart implemented in some language (such as OWL), this operational version of the pattern can also be reused, amplifying the benefits of applying the pattern. However, we defend here that DROPs should be captured in a codification language independent manner. This allows for a modeling solution to be implemented in multiple codification languages.

A COP has to be small (typically two to ten classes with relations defined between them) [2]. Moreover, a COP can be an element in a partial order, where the ordering

relation requires that at least one of the classes or relations in the pattern is specialized [2]. These characteristics are essential for DROPs in an OPL. A user should be able to read the pattern, understand its applicability and decide if it is useful for the problem at hands or not. Once decided which DROPs to reuse, the user can specialize their concepts and relations.

A domain ontology typically results from the composition of several COPs, with appropriate dependencies between them, plus the necessary design expansion based on specific needs [3]. Making this knowledge explicit is essential for achieving the main benefits of reuse. Thus, organizing DROPs in catalogues is not a good choice. In a conventional catalog there is a lack of a strong sense of connection. We need something stronger than simply knowing that another pattern in the collection is related in some way. When collections are presented in conjunction with, for example, pattern sequences, we start to get a stronger sense of connection [9]. This is especially important for reusing DROPs.

An Ontology Pattern Language (OPL) aims to provide holistic support for using DROPs in ontology development for a specific application domain. It should provide explicit guidance on what problems can arise in that domain, inform the order to address these problems, and suggest one or more patterns to solve each specific problem. Thus, an OPL should support the explicit consideration of complementing or conflicting pattern combinations to solve a given problem, along with guidelines for integrating patterns into a concrete ontology conceptual model.

An OPL should indicate explicitly which referenced patterns address mandatory aspects and which ones address optional aspects. To ensure a stable and sound pattern application, referenced patterns should be presented in the suggested application order. Without this explicit procedural guidance, a representation that fits the basic network of the patterns might not provide a suitable process that helps to ensure a sufficiently complete and well-formed ontology.

OPLs are structured to support and encourage the application of one pattern at a time, in the order defined by the pattern sequences that result from the chosen paths through the language. This guideline ensures that the main property of piecemeal growth is preserved: the 'whole' always precedes its 'parts'. A pattern language is of little use if its audience loses the big picture. Conversely, the essential information of each individual pattern within the language must still be preserved [9].

In summary, an OPL should give concrete and thoughtful guidance for developing ontologies in a given domain, addressing at least the following issues: (i) What are the key problems to solve in the domain of interest? (ii) In what order should these problems be tackled? (iii) What alternatives exist for solving a given problem? (iv) How should dependencies between problems be handled? (v) How to resolve each individual problem most effectively in the presence of its surrounding problems?

Using the notion of OPLs, we can reorganize ontology pattern catalogues. We might provide an entry in a catalogue for each domain of interest. Each entry in the catalogue, in turn, can be viewed as a special purpose pattern language that advises developers how to construct a domain ontology with the help of DROPs.

For illustrating the ideas discussed above, in the next section, we present an OPL in the domain of Software Process. The patterns there were extracted from the Software

Process Ontology (SPO) presented in [17]. SPO has been developed since 1997, and it results from several revisions. The latest version was obtained as a result of a reengineering effort to ground it in UFO [17]. In the version presented here, we managed to advance further improvements, mainly regarding modularity, which directly affects reusability. For this reason, we decided to restructure SPO as an OPL.

5 An Ontology Pattern Language for the Software Process Domain (SP-OPL)

Figure 2 shows a UML activity diagram giving an overview of the SP-OPL. An activity diagram is one of the possible modeling notations comprising UML and it is the standard UML notation for representing temporal sequencing constraints between activity types and, hence, for specifying the possible order of execution between activities. In the model of Fig. 2, we use activities in an activity diagram (rounded rectangles) to represent specific patterns. Moreover, we use the activity ordering notation to represent the procedural rules governing the admissible sequences in which these patterns can be used. In that diagram, an extension to the original UML notation (dotted lines with arrows) was introduced to show variant patterns.

It is important to emphasize that we would have employed activity diagrams (or a language with similar representation capabilities) for that purpose regardless of the domain under study, i.e., the choice for using an activity-ordering language is related to the need for defining the permissible sequence of instantiation of the patterns. In particular, it bears no relation to the fact that, incidentally, the domain under study is about *Software Processes*.

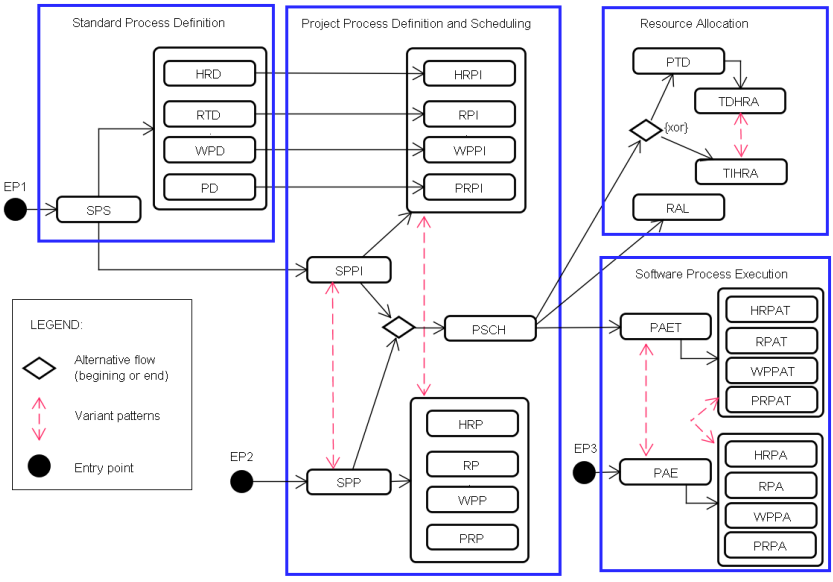


Fig. 2. Software Process Ontology Pattern Language (SP-OPL)

The main problem areas addressed by the SP-OPL are: *Standard Process Definition*, *Project Process Definition and Scheduling*, *Resource Allocation*, and *Software Process Execution*. Table 1 shows the patterns that compose the SP-OPL.

As shown in Fig. 2, SP-OPL has three entry points, depending on the focus of the ontology engineer. When the requirements for the domain ontology being developed include problems related to *Standard Process Definition*, the start point is EP1. In this case, first the ontology engineer should address problems related to how a standard process is structured in terms of standard sub-processes and activities (SPS). Following, he can optionally address problems related to the definition of human roles (HRD), types of resources (hardware and software) (RD), types of work products required (input) and produced (output) (WPD), and procedures (methods, techniques, guidelines etc.) (PD) that are required for performing each standard activity when it is instantiated in the scope of a project.

When the requirements for the ontology being developed include problems related to *Project Process Definition and Scheduling*, the start point is either EP2 or SPS. In this case, the ontology engineer has to first deal with problems related with the process planning in terms of project sub-processes and activities. If there is already defined a standard process, project process planning can be done by means of instantiating the standard process (SPI – Software Process Planning via Instantiation)[1], otherwise, the ontology engineer should consider planning the project process from scratch (SPP). Once defined the project processes and activities, he can treat modeling problems related to scheduling them (PSCH). Moreover, the ontology engineer can optionally treat modeling problems related to planning human roles (HRPI/HRP), types of resources (hardware and software) (RPI/RP), types of work products required (input) and produced (output) (WPPI/WPP), and procedures (methods, techniques, guidelines etc.) (PRPI/PRP) that are required for performing each project activity.

For dealing with problems related to *Resource Allocation*, it is necessary to have the project process planed and scheduled. Resource Allocation involves patterns regarding hardware and software resource allocation (RAL), project team definition (PTD), and human resource allocation. Human resource allocation problems can be solved considering constraints imposed by a project team (TDHRA) or not (TIHRA).

Finally, when there are requirements related to the *Software Process Execution*, the start point is either EP3 or PSCH. EP3 should be chosen when it is not a requirement for the ontology to address process planning and scheduling. In this case, the ontology engineer has to first deal with problems related to the execution of processes and activities (PAE). Then he can address problems related to resource (human and other) participation (HRPA and RPA), procedures adopted (PRPA), and work product inputs and outputs (WPPA). On the other hand, if the project process is already scheduled, it is possible to address problems related to process and activity execution and tracking, which involves the corresponding variant patterns PAET, HRPAT, RPAT, PRPAT

[1] The patterns SPPI, HRPI, RPI, WPPI, and PRPI shown in Fig. 2 are not listed in Table 1, due space limitations. Those patterns are variant patterns of SPP, HRP, RP, WPP and PRP, respectively, considering that they address the same problems, but considering the instantiation of a standard process or activity.

and WPPAT. These patterns, which are not shown in Table 1, address the same problems described above, but considering that it is possible to check if the execution of activities and processes conforms to their previous definition (process tracking).

Table 1. Domain-Related Ontology Patterns (DROPs) in the SP-OLP

Id	Name	Intent
Standard Process Definition		
SPS	Standard Process Structure	Represents how a standard software process is defined in terms of standard sub-processes and activities
HRD	Standard Activity Human Role Definition	Defines the human roles responsible for performing a standard activity in the projects that instantiate it
RTD	Standard Activity Resource Type Definition	Defines the types of resources (hardware and software) required for performing a standard activity
WPD	Standard Activity Work Product Definition	Defines the types of work products required (input) and produced (output) when performing a standard activity
PD	Standard Activity Procedure Definition	Defines the procedures (methods, techniques, guidelines etc.) to be applied when performing a standard activity
Project Process Definition and Scheduling		
SPP	Software Process Planning	Represents how a software process is planned in terms of sub-processes and activities
PSCH	Process Scheduling	Defines the time boundary for project processes and activities
HRP	Human Role Planning	Defines the human roles responsible for performing a project activity
RP	Resource Planning	Defines the types of resources (hardware and software) required for performing a project activity
WPP	Work Product Planning	Defines the types of work products required (input) and produced (output) when performing a project activity
PRP	Procedure Planning	Defines the procedures (methods, techniques, guidelines etc.) to be applied when performing a project activity
Resource Allocation		
PTD	Project Team Definition	Defines the human resources that are member of a project team
TDHRA	Team-dependent Human Resource Allocation	Allocates human resources to project activities, considering team allocation constraints
TIHRA	Team-independent Human Resource Allocation	Allocates human resources to project activities, when there is not a project team formally defined
RAL	Resource Allocation	Allocates resources (hardware equipments and software tools) to project activities
Software Process Execution		
PAE	Process and Activity Execution	Register the occurrences of processes and activities.
HRPA	Human Resource Participation	Registers the participation of Human Resources in an activity occurrence
RPA	Resource Participation	Registers the participation of Resources (hardware equipment or software tool) in an activity occurrence
WPPA	Work Product Participation	Register the participation of Work Products (as input or output) in an activity occurrence.
PRPA	Procedure Participation	Register the adoption of procedures by an activity occurrence

Figure 3 shows the conceptual model of the "Process and Activity Execution (PAE)" DROP. The intent of this pattern is to represent the occurrences of processes and activities in the context of a project, and their mereological structure. The following competency questions are addressed by this pattern: (CQ1) How is a process occurrence structured in terms of sub-processes and activities? (CQ2) When did a process/activity occurrence start and when did it end? (CQ3) From which activity occurrences does an activity occurrence depend on?

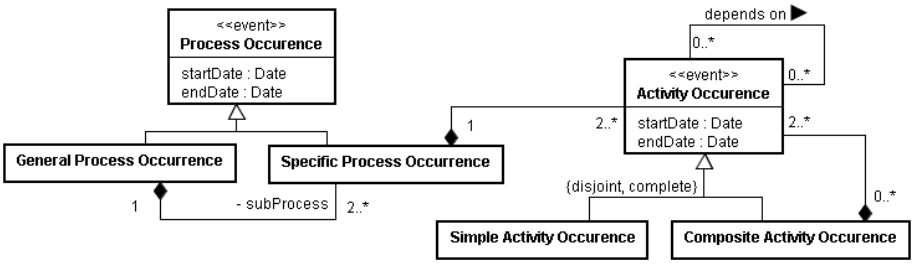


Fig. 3. The "Process and Activity Execution" (PAE) pattern

The foundations for the PAE pattern were given by UFO-B [16]. **Process Occurrences** and **Activity Occurrences** are complex events, and the whole-part relations between events are strict partial order. In the software process domain, there are two main kinds of **Process Occurrences**: **General Process Occurrence** and **Specific Process Occurrence**. A general process occurrence is the whole execution of a process. It is composed of specific process occurrences, allowing an organization to decompose a general process into sub-processes. A specific process occurrence, in turn, is decomposed into **Activity Occurrences**. Activity occurrences can be simple or composite. A composite activity occurrence is a complex event that is composed by other activity occurrences. A simple activity occurrence is not composed by other activity occurrences, but it is still a complex event in UFO-B, since it is composed by other events representing the participations of human resources, hardware and software resources, work products, and procedures in the activity occurrence.

The PAE pattern has some related patterns, with different types of relations holding between them. PAE has a variant pattern, the "Process and Activity Execution and Tracking (PAET)" pattern, which is an alternative to PAE when a project has a process previously defined and scheduled, allowing to track the execution against to what was previously planned. When PAE is used, its use can be followed by the use of patterns whose intent is to represent the participations of human resources (HRPA), software and hardware resources (RPA), procedures (PRPA), and work products (WPPA). Figure 4 presents the conceptual model of the WPPA pattern.

This pattern shows that an activity occurrence can have as its parts **Artifact Participations**, which are also events. An artifact participation is the participation of a single artifact. This is in line with UFO-B, which says that events are ontologically dependent entities in the sense that they existentially depend on objects in order to exist. **Artifact**, in turn, is a category in UFO-A [14], since it is a dispersive universal that aggregates essential properties (not shown in this pattern) that are common to

different subtypes of artifacts. Artifact participations can be of three types: (i) **Artifact Creation**, meaning that the artifact is created during the activity occurrence, and thus it is an output of this activity occurrence (the **/produces** derived relation); (ii) **Artifact Usage**, meaning that the artifact is only used during the activity occurrence, and thus it is only an input for the activity occurrence (the **/uses** derived relation); and (iii) **Artifact Change**, meaning that the artifact is changed during the activity occurrence, and thus it is both input and output of the activity occurrence. The foundations for this conceptualization are given by UFO-C [16], which defines four types of resource participations: creation, termination, usage and change. In the case of software processes, we consider that artifacts are not thrown away in activity occurrences, and thus there is not a case of termination participation in this domain.

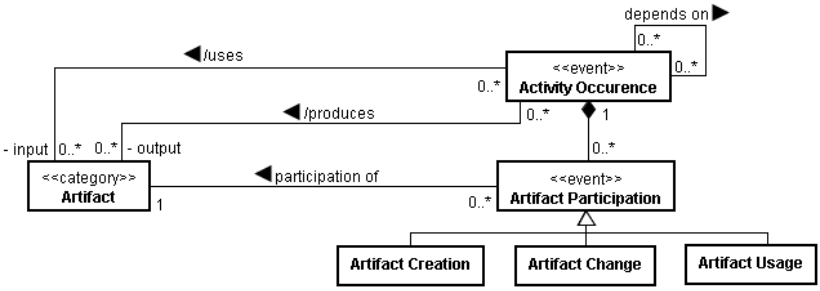


Fig. 4. The "Work Product Participation" (WPPA) pattern

SP-OPL was used for building a domain ontology about the software measurement process. Figure 5 shows a fragment of this domain ontology, considering the reuse of the two patterns presented before (PAE and WPPA). Concepts reused from the patterns are presented in grey.

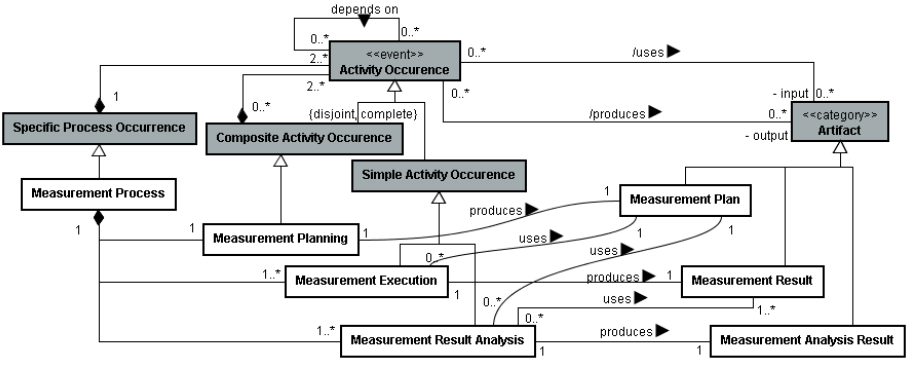


Fig. 5. A fragment of a Domain Ontology for the Software Measurement Process

As shown in Fig. 5, the **Measurement Process** is composed by activity occurrences of **Measurement Planning**, **Execution**, and **Result Analysis**. The first one is a composite activity occurrence, although, for simplicity, its parts are not shown in

the figure. The other two are simple activity occurrences. Measurement Planning produces a **Measurement Plan**, which is used by the other activity occurrences (Measurement Execution and Result Analysis occurrences). Measurement Execution produces **Measurement Results**, which are used by Measurement Result Analysis for producing **Measurement Analysis Results**.

6 Related Works

Our work is strongly inspired, on one side, by works on Ontology Design Patterns, especially those developed by Gangemi, Presutti and colleagues [2, 3, 19]; on the other side, by works on Pattern Languages in Software Engineering, especially those developed by Buschmann, Schmidt and colleagues [9, 10]. In fact, we believe that our main contribution in this paper is to introduce the idea of pattern languages, as used in Software Engineering, in the field of Ontology Design Patterns, which is especially important for Ontology Engineering and consequently for Semantic Web.

At the best of our knowledge, we are the first to organize domain-related ontology patterns as Ontology Patterns Languages (OPLs). However, it is important to reinforce that we borrowed the term "pattern language" from Software Engineering (SE), where it has a special meaning [8, 9]. A pattern language, in this context, is a network of interrelated patterns, plus a process for systematically solving software development problems [8, 9]. Highlighting the particular meaning that we associate to the term Pattern Language is particularly important in order to avoid confusion with existing literature. For instance, in [20], Noppens and Liebig seek to develop a language to encode OWL patterns in a declarative way. They did not use the term OPL in the sense we did.

Finally, although an OPL defines a process for traveling along the patterns, it is not a method for building ontologies. An OPL can be used jointly with several methods. For instance, the measurement process ontology partially presented in Section 5 was developed using the method SABiO [21], adapting one of its activities (Reusing Existing Ontologies) for using an OPL. In particular, the eXtreme Design (XD) method [4] is quite suitable to be used with an OPL, since it is a content pattern-oriented method. Tasks such as "Match Competency Questions to Generic Use Case", "Select Content Patterns (CPs) to Reuse", and "Reuse and Integrate Selected CPs" could be easily adapted to consider patterns in an OPL. In fact, an OPL has great potential to improve XD. Take the experiments done by Blomqvist et al. [3], which evaluate pattern-based ontology design using XD. As pointed by these authors, the participants of the experiments may be faster in using patterns if they are more familiar with them. Moreover, the particular set of CPs could have an impact on the time spent in the ontology development. In the reported experiments, most of the patterns were quite general. Regarding this, Blomqvist et al. suggest that more specific patterns could also improve this aspect. Based on those perceptions, we argue that an OPL could be used to improve XD. Firstly, the patterns in an OPL are domain-related patterns, and thus more specific ones. Secondly, the OPL gives a context for the patterns, and guides the ontology engineer in traveling along them.

7 Final Considerations

Nowadays, ontology design patterns are recognized as a beneficial approach for ontology development [2, 3]. Particularly in the case of Domain-related Ontology Patterns (DROPs), these benefits can increase if we organize them as a pattern language, as it has been shown to be the case in Software Engineering. In this paper we introduced the notion of Ontology Pattern Language (OPL) as a network of interrelated DROPs with procedural rules prescribing the order in which they can be combined. OPLs can then be used to systematically solve ontology modeling problems in a given (core) domain. We also briefly present an OPL for the Software Process domain (SP-OPL), which illustrates the approach.

We shall consider that, as pointed out by Buschmann et al. [9], useful pattern languages must be sufficiently complete and mature. In particular, they must be complete regarding the coverage of the problem and solution spaces for their subjects, and must be mature regarding the quality and interconnection of their constituent patterns. Quality and maturity cannot be produced casually and hastily, but require great care and much time to age gracefully. OPLs are not an exception. Moreover, we claim that OPLs must present some characteristics generally pointed as being present in "beautiful ontologies", such as [22]: satisfy relevant requirements, have a good coverage of the targeted domain, be often easily applicable in some context, be structurally well designed (either formally or according to desirable patterns), and their domains should introduce constraints that lead to modeling solutions that are non-trivial.

Finally, pattern languages should evolve in response to various events and insights. As new experiences are gained developing ontologies with reuse, it is certainly desirable to integrate these new experiences and patterns into related existing pattern languages to keep them up to date. Consequently, all pattern languages, from the rawest to the most mature, should always be considered as a work in progress that is subject to continuous revision, enhancement, refinement, completion, and sometimes even complete rewriting [9].

Acknowledgments. This research is funded by the Brazilian Research Funding Agencies FAPES (Process Number 52272362/11) and CNPq (Process Number 483383/2010-4).

References

1. Poveda-Villalón, M., Suárez-Figueroa, M.C., Gómez-Pérez, A.: Reusing Ontology Design Patterns in a Context Ontology Network. In: Proc. of the 2nd International Workshop on Ontology Patterns – WOP 2010, Shangai, China (2010)
2. Gangemi, A., Presutti, V.: Ontology Design Patterns. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies, 2nd edn., pp. 221–243. Springer (2009)
3. Blomqvist, E., Gangemi, A., Presutti, V.: Experiments on Pattern-based Ontology Design. In: Proc. of the Fifth International Conference on Knowledge Capture – K-CAP 2009, California, USA, pp. 41–48 (2009)

4. Presutti, V., Daga, E., Gangemi, A., Blomqvist, E.: eXtreme Design with Content Ontology Design Patterns. In: Proc. Workshop on Ontology Patterns, Washington, D.C., USA (2009)
5. Clark, P., Thompson, J., Porter, B.: Knowledge patterns. In: Proc. of the 7th International Conference on Principles of Knowledge Representation and Reasoning – KR 2000, San Francisco, USA, pp. 591–600 (2000)
6. Svatek, V.: Design Patterns for Semantic Web Ontologies: Motivation and Discussion. In: Proc. of the 7th Conference on Business Information Systems, Poznan, Poland (2004)
7. Alexander, C., Ishikawa, S., Silverstein, M.: A Pattern Language. Oxford University Press, New York (1977)
8. Deutsch, P.: Models and Patterns. In: Greenfield, J., Short, K., Cook, S., Kent, S. (eds.) Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools. Wiley Publishing Inc., Indianapolis (2004)
9. Buschmann, F., Henney, K., Schmidt, D.C.: Pattern-Oriented Software Architecture. On Patterns and Pattern Languages, vol. 5. John Wiley & Sons Ltd. (2007)
10. Schmidt, D., Stal, M., Rohnert, H., Buschmann, F.: Pattern-Oriented Software Architecture. Patterns for Concurrent and Networked Objects, vol. 2. Wiley Publishing (2000)
11. Scherp, A., Saathoff, C., Franz, T., Staab, S.: Designing core ontologies. Applied Ontology 6(3), 177–221 (2011)
12. Guarino, N.: Formal Ontology and Information Systems. In: Guarino, N. (ed.) Formal Ontology and Information Systems, pp. 3–15. IOS Press, Amsterdam (1998)
13. Borgo, S., Masolo, C.: Foundational Choices in DOLCE. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies, 2nd edn., pp. 361–381. Springer (2009)
14. Guizzardi, G.: Ontological Foundations for Structural Conceptual Models. Universal Press, The Netherlands (2005)
15. Guizzardi, G.: On Ontology, ontologies, Conceptualizations, Modeling Languages and (Meta)Models. In: Vasilecas, O., Edler, J., Caplinskas, A. (eds.) Databases and Information Systems IV, pp. 18–39. IOS Press, Amsterdam (2007)
16. Guizzardi, G., Falbo, R.A., Guizzardi, R.S.S.: Grounding software domain ontologies in the Unified Foundational Ontology (UFO): the case of the ODE software process ontology. In: Proc. of the XI Iberoamerican Workshop on Requirements Engineering and Software Environments – IDEAS 2008, Recife, Brazil, pp. 244–251 (2008)
17. Bringuente, A.C.O., Falbo, R.A., Guizzardi, G.: Using a Foundational Ontology for Reengineering a Software Process Ontology. Journal of Information and Data Management 2(3), 511–526 (2011)
18. Barcellos, M.P., Falbo, R.A., Dal Moro, R.: A Well-founded Software Measurement Ontology. In: Proc. of the 6th International Conference on Formal Ontology in Information Systems – FOIS 2010, Toronto, Canada, pp. 213–216 (2010)
19. Gangemi, A.: Ontology Design Patterns for Semantic Web Content. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 262–276. Springer, Heidelberg (2005)
20. Noppens, O., Liebig, T.: Ontology Patterns and Beyond - Towards a Universal Pattern Language. In: Proc. Workshop on Ontology Patterns, Washington, D.C., USA (2009)
21. Falbo, R.A.: Experiences in Using a Method for Building Domain Ontologies. In: Proc. of International Workshop on Ontology in Action, Banff, Canada (2004)
22. d'Aquin, M., Gangemi, A.: Is there beauty in ontologies? Applied Ontology 6(3), 165–175 (2011)

An Ontology Design Pattern for Cartographic Map Scaling*

David Carral[1], Simon Scheider[2], Krzysztof Janowicz[3], Charles Vardeman[4], Adila A. Krisnadhi[1], and Pascal Hitzler[1]

[1] Kno.e.sis Center, Wright State University, USA
{carral.2,pascal.hitzler,krisnadhi.2}@wright.edu
[2] Institute for Geoinformatics, University of Münster, Münster, Germany
simonscheider@web.de
[3] Department of Geography University of California, Santa Barbara, CA, USA
jano@geog.ucsb.edu
[4] Center for Research Computing, University of Notre Dame, Notre Dame, IN, USA
Charles.F.Vardeman.1@nd.edu

Abstract. The concepts of *scale* is at the core of cartographic abstraction and mapping. It defines which geographic phenomena should be displayed, which type of geometry and map symbol to use, which measures can be taken, as well as the degree to which features need to be exaggerated or spatially displaced. In this work, we present an ontology design pattern for map scaling using the Web Ontology Language (OWL) within a particular extension of the OWL RL profile. We explain how it can be used to describe scaling applications, to reason over scale levels, and geometric representations. We propose an axiomatization that allows us to impose meaningful constraints on the pattern, and, thus, to go beyond simple surface semantics. Interestingly, this includes several functional constraints currently not expressible in any of the OWL profiles. We show that for this specific scenario, the addition of such constraints does not increase the reasoning complexity which remains tractable.

1 Introduction and Motivation

The notion of scale is at the very core of cartography and essential for the visualization of geo-information in maps [14]. However, scale also plays a key role for knowledge representation and measurement [8]. In its simplest form, scale can be expressed as a representative fraction that specifies the relation between the distance measured on a map to the corresponding distance in the physical world [16]. For example, a *large* map scale of 1:25000 indicates that one unit of measure on this map corresponds to 25000 units of the same measure on the ground. In turn, a *small* scale map of 1:100000 covers a larger region.[1]

* An extended technical report with the appendix can be found at http://knoesis.wright.edu/pascal/resources/publications/odp-carto-scaling-TR.pdf

[1] The reason for this is that the terms *small* and *large* refer to the representative fraction, with 1/25000 being the larger fraction. Note that this usage of small and large differs from how other domains refer to scale, e.g., as in 'large scale study'.

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 76–93, 2013.

Dealing with scale-dependent representations of phenomena in maps in a seamless manner is called *map scaling*. It involves a lot of specialized knowledge. For instance, highway symbols may have to be exaggerated and displaced before they can be rendered on a small scale map. Thus, trying to read the street widths or pathway from a small scale map will yield meaningless results. Similarly, geographic features such as creeks, lakes, and ponds will have to be fused, simplified, or omitted. Buildings may be represented as polygons or point-like features at some scale, but may be fused to blocks and neighborhoods at a smaller scale. Generalizing further will collapse these blocks into a representation of a whole city. Finally, on a global map, only major capitals will be left while all other cities may disappear.

Map scaling requires choosing a certain data representation as well as a mode of display for every geographic phenomenon type at each scale level inside a map extent. Cartographic abstraction is, to a large degree, a sophisticated craft. It involves semantic as well as cartographic knowledge, including knowledge about the involved type of features, map generalization rules, and appropriate symbolism for layout as well as symbol placings [14].

However, currently, the knowledge about scale dependency of digital representations remains inaccessible. This makes the integration of digital information across scales and across applications challenging. While there is a rich body of work on how to address scale in cartography, most of the knowledge involved is not specified formally or is hidden in application source code. This contradicts with one of the major paradigm shifts underlying Semantic Web research, namely to enable the creation of smarter data instead of smarter applications. Rather than engineering increasingly complex software, the so-called business logic should be transferred to the level of (meta) data. The rationale behind this is that smarter data will enable more usable and flexible applications, while smarter applications alone fail to improve data along the same dimensions. So far, the notion of scale has barely been given any attention in the Semantic Web, even though most digital resources have an intrinsic scale level. In particular, we do not know of any published ontology patterns on scale.

In this paper, we propose a scale ontology design pattern (in the sense of [6]) which can be used to document and publish knowledge about map scaling applications on the web. It describes the scale dependent representation of geographic phenomena in such applications, and makes the underlying scaling decisions explicit and accessible on the Web. Hence, our work may be integrated with provenance ontologies such as Prov-O[2]. With respect to (semantic) Web services, our pattern can be used to link and track geo-features across scale levels. For example, one could query for a map service that serves base data in the scale required to visualize features from another service. Furthermore, the map scaling pattern allows to *reason on scaled geographic information*. For example, one can check whether two phenomena can be displayed together at a single scale level across scaling applications. One may also gather information about a certain geographic phenomenon at a high level of geographic detail across

[2] `http://www.w3.org/TR/prov-o/`

the Web. And one can check scaling applications for consistency of scaling and representation.

From the viewpoint of semantic technology, we will address two challenges for such a pattern: First, from a conceptual viewpoint, the pattern has to ensure that geographic features are traceable across scale levels, and that the basic logical constraints inherent in (rendering) applications are formally captured in the pattern. Second, from a computational point of view, reasoning with the pattern needs to be tractable. For this purpose, the application logic needs to captured in a tractable subset of first order logic (FOL). However, as we will show later on, current OWL fragments are not flexible enough to capture the required functional constraints. We will show that these constraints can be captured by a certain logical fragment that remains in polynomial order of complexity.

In the following, we will first discuss map scaling in order to motivate and help understand our axiomatization. Then, we will discuss a formal axiomatization on the level of a functional pattern as well as on the level of a DL fragment which allows tractable reasoning. We will evaluate the pattern by showing its use in an existing application that studies Malaria, before we discuss and conclude the paper.

2 Map Scaling in a Nutshell

Practical solutions of the map scaling problem draw on a number of core issues in Geographic Information Science (GIScience) as well as Computer Cartography [14,13]. In this section, we will give a very brief overview of key concepts and related work. We will also suggest a conceptual view on them, which helps put our ontology pattern into context.

2.1 Map Scale

With the notion *scale* we mean *cartographic scale*, which refers to the ratio of the depicted size of a feature on a cartographic map relative to its actual size [16]. There are other notions of scale. For example, the *scale of analysis* and the *scale of observation* are scales induced by analyzing or observing a phenomenon. The *scale of a phenomenon* is the scale at which a phenomenon appears or can best be studied [20]. The latter kinds of scale are not explicitly addressed in this paper, however, they may decide about whether certain kinds of entities *appear* at certain scales or not [16]. In this context it will be important to keep in mind that a unique map scale exists only for a *map image*, i.e., a map displayed as an image, and not for map data, which may be displayed at several different scales. Furthermore, (digital) zooming should not be confused with scaling, as no new information is added or removed while zooming. So far, we do not know of any published work addressing the issue of scale in the Semantic Web context[3]. However, the topic is central for GIScience [5].

[3] Preliminary unpublished work is avaible here: `http://vocamp.org/wiki/Scale-vocab`.

2.2 Geographic Phenomena

Geographic phenomena can be represented in a cartographic map. They come in different feature/object types, such as rivers, cities, roads, buildings, people, landparcels, or the earth's surface. They may also consist of conventionally established regions in a spatial reference system, such as the borders of Germany. Furthermore, they may consist of qualities, such as temperature or windspeed or building height. Geographic phenomena can be represented by geo-ontologies, ranging from *top-level ontologies* such as [2] to domain ontologies such as NASA's Sweet[4]. Most importantly, however, geographic phenomena can be measured in terms of *reference systems*. This means one can unambiguously *observe their extent* in at least some *spatial reference system* that allows to refer to geographic locations, such as WGS84. This makes them amenable to cartographic mapping. Externally, geo-ontologies can be aligned with the phenomenon class of our pattern to differentiate among these types of phenomena and subclass them further.

2.3 Map Data

Map data is any set of data which represents geographic phenomena and which can be cartographically mapped (i.e., displayed in a map). For this purpose, it needs to contain a *spatial geometry*, i.e., a type of data which specifies a subset of points in some spatial reference system. This subset may be a single point, a line, or some region. A frequent data structure are *geodata records* used in Geographic Information Systems (GIS), i.e., records of a spatial geometry and non-spatial attribute values. The latter can represent measurable qualities, e.g., temperature, as well as cartographic symbol types, e.g., a color symbol, which may be used to display the map data at a certain scale level. Depending on the kind of geometry, one can distinguish two kinds of map data: one is *raster data*, where the geometry forms a regular tessellation (a topological cover with regular polygons) of a subset of the reference space. An example is a satellite image. Another one is *vector data*, where geometries can be irregular and need not form a tesselation. Map data often comes in collections representing phenomena of similar type called *layers*. The different kinds of map data form the context but are not part of our ontology pattern.

2.4 Resolution

Resolution is a central notion for map scaling, however, its semantic specification is challenging [4]. In this section, we clarify our use of the term in an informal way, following ideas in [4]. Resolution can be regarded as a *property of map data* which allows to measure its *level of detail*. Note that map data has a resolution but not a map scale in our sense, since it is not necessarily displayed, and not necessarily at a single scale level. There are different proxy-measures for resolution, depending on the purpose. One proxy measure is based on *grain*

[4] `http://sweet.jpl.nasa.gov/2.2/sweetAll.owl`.

size, i.e., the extent of an atomic mapping entity in a spatial reference system. An example is the *instantaneaous field of view* of a satellite, i.e., the area on the ground surface that corresponds to a single remote detector element, or the *minimal mapping unit* [8]. Our pattern allows grain size resolution levels of map images interchangeably with their map scale levels in order to restrict potential display of map data to a scale level. However, we do not explicitly model scale and resolution of map data, which may be done in the future.

2.5 Map Image

A map image is the result of displaying map data in a map display, i.e., a medium (e.g., paper) used to visualize the map, according to a map scale. Therefore, a map image has a grain size resolution as well as a map scale, the latter because it is projected into a map display in which each pixel has a measurable size. A map scale can be computed from the image resolution by multiplying the latter with the pixel size. In a map scaling application, a new map image is generated every time one zooms in or out. Note that a *map image file* is a different beast. In contrast to the former, the latter is a form of map data in raster format. While the latter has a grain (pixel) size resolution, it does not have a map scale (since its pixels do not have fixed display sizes).

2.6 Scaling and Map Generalization

Scaling [20] refers to the seamless transfer of information between different scale levels. Even though a digital map display in principle allows zooming in and out regardless of map data or phenomena, a *visually graspable image* as well as a *semantically adequate and computationally efficient* data representation depends on the resolution as well as the design of the map data. For example, from a visual standpoint, representations can become congested, coalesced or imperceptible at inappropriate scales [14]. There is web technology available which allows to specify and serve scale dependent maps in the web[5]. The problem of map generalization has been addressed by Computer Cartography with operations that modify map data geometry and symbolization accordingly. These include simplification, smoothing, aggregation, amalgamation, merging, collapse, refinement, exaggeration, enhancement and displacement [14]. Since existing algorithms are not able to automatize generalization to a degree which corresponds in quality to manual cartographic techniques [15], map generalization in practice is still done manually or semi-automatically [15].

Furthermore, generalization has an aspect of *(semantic) modeling* [15], which relates to observation and intrinsic phenomenon scales mentioned above [20]. As a consequence of scaling, not only the visual representation of a phenomenon, but also geometry, semantic classes, properties and relations are affected. Furthermore, even the occurrence of individuals is not invariant across scales. Existing

[5] For example, the Feature Portrayal Service (FPS) and Styled Layer Descriptor (SLD) standards of the Open Geospatial Consortium (OGC) `http://www.opengeospatial.org`.

formalisms for scaling and generalization, such as *stratified map spaces* [19], simplify the problem to lifting or generalizing classes and spatial regions, not objects. However, as a matter of fact, individual geographic objects, such as cities and countries, appear at smaller scales, while streets, places and buildings only appear at larger scales. From an ontological viewpoint, generalization therefore affects the level of detail of an ontology and thus may change an ontological theory. This can be treated as a mereological problem on the side of phenomena, such as the problem to individuate road junctions from a road segment network [17]. For our pattern, we will therefore assume a generalization relation between map data sets, which reflects diverse ways of generalizing data sets on different scale levels.

3 Preliminaries

To define and implement the pattern in a tractable way, we introduce an extension of the existing description logics fragment DLP [9], the logic fragment underlying the OWL RL profile. We denote the extension as $\text{DLP}_{\exists}$ and show that, under certain appropriate syntactic restrictions, reasoning over this DL fragment remains tractable. Throughout the paper we will show how this extension allows us to express some useful constraints enhancing the usability of the map scaling design pattern.

We will make use of the DL notation along the paper, as we think it improves readability and understanding of the ideas presented. Furthermore, some of the new features included in the extended fragment $\text{DLP}_{\exists}$ are not even part of the OWL language, such as role conjunction, and therefore cannot be expressed in any of the existing OWL syntaxes. DL syntax allows us to express these constructors without having to introduce major changes in the existing notation. Henceforth, we only implement the part of the pattern that can be done making use of the current available constructors in OWL. If the reader is not familiar with the DL notation see [12] for a quick introduction and [1] for a lengthier one.

The set $\mathcal{C}$ of allowed concepts in $\text{DLP}_{\exists}$ is the set of all concepts that can be constructed making use of all constructors in Table 1. Like in other DL fragments, we can divide the axioms in a $\text{DLP}_{\exists}$ knowledge base into *ABox* $\mathcal{A}$, *TBox* $\mathcal{B}$, and *RBox* $\mathcal{R}$ statements.[6]

A $\text{DLP}_{\exists}$ *TBox* [*RBox*] is a finite set of general concept inclusions (GCIs) [role inclusion axioms (RIAs)] as described in Table 1. An *ABox* is a finite set of concept and role assertions also as described in Table 1. Furthermore, $\text{DLP}_{\exists}$ restricts the at-most one cardinal restriction constructor ($\leq 1R.C$) to only appear in the right hand side of GCIs. Note that $\text{DLP}_{\exists}$ allows for the use of unrestricted role conjunctions.

We do not impose any kind of role regularity restrictions in $\text{DLP}_{\exists}$ as defined for $\mathcal{SROIQ}$ [10], which is the logic fragment underlying the OWL DL profile. The regularity restrictions are applied to the tractable OWL profiles in order to define OWL DL as a superset of these. Otherwise some of the tractable fragments would allow some expressivity not available in OWL DL.

[6] Assertional, terminological, and role boxes respectively

Table 1. DLP$_\exists$ Constructors. $C, D \in \mathcal{C}$ are concepts, R and S are roles, and $\{t\}$ is a nominal.

Name	**Syntax**	**Semantics**
Concept Assertion	$C(a)$	$a^\mathcal{I} \in C^\mathcal{I}$
Role Assertion	$R(a, b)$	$\langle a, b\rangle \in R^\mathcal{I}$
GCI	$C \sqsubseteq D$	$C^\mathcal{I} \subseteq D^\mathcal{I}$
Existential Restriction	$\exists R.C$	$\{\delta \mid$ there is ϵ with $\langle\delta, \epsilon\rangle \in R^\mathcal{I}$ and $\epsilon \in C^\mathcal{I}\}$
≤ 1 Card. Restriction	$\leq 1 R.C$	$\{\delta \mid \sharp\{\langle\delta, \epsilon\rangle\} \in R^\mathcal{I} \mid \epsilon \in C^\mathcal{I}\} \leq 1$
Concept Conjunction	$C \sqcap D$	$C^\mathcal{I} \cap D^\mathcal{I}$
Top concept	$\top$	$\Delta^\mathcal{I}$
Bottom concept	$\bot$	$\emptyset$
RIA	$R \sqsubseteq S$	$R^\mathcal{I} \subseteq S^\mathcal{I}$
Role Inverse	R^-	$\{\langle\delta, \epsilon\rangle \mid \langle\epsilon, \delta\rangle \in V^\mathcal{I}\}$
Role Chain (RIA)	$R_1 \circ \ldots \circ R_n$	$R_1^\mathcal{I} \circ \ldots \circ R_n^\mathcal{I}$
Role Conjunction	$R \sqcap S$	$R^\mathcal{I} \cap S^\mathcal{I}$

To preserve tractability of DLP$_\exists$ we need to impose restrictions in the use of the existential constructor on the right hand side of GCIs, otherwise the fragment becomes undecidable as shown in [11]. The definition of necessary or at least sufficient conditions under which DLP$_\exists$ still retains tractability are out of the scope of this paper. Nevertheless as we show in the Appendix that reasoning in DLP$_\exists$ is not only decidable but tractable for the defined pattern and leave the definition of these restrictions as further work. We also elaborate on how simply we could devise a reasoning algorithm based on some of the existing reasoning procedures for DLP.

It may give the impression that the set of constructors represented in Table 1 do not cover the complete expressivity of the OWL RL profile. Indeed, we do not explicitly include some of the constructors that are part of the specifications of the RL profile. As shown Table 2, all of the original RL constructors can be constructed from the set of constructors in Table 1. A smaller set of constructors allows for more succinct definitions and theorems in further sections.

Table 2. DLP$_\exists$ Syntactic Sugar. C, D, and E are concepts, and R is a role.

RL Axiom	Equivalent Axioms
$C \sqsubseteq \forall R.D$	$\exists R^-.C \sqsubseteq D$
$C \sqcup D \sqsubseteq E$	$C \sqsubseteq E$
	$D \sqsubseteq E$
$C \sqsubseteq \neg D$	$C \sqcap D \sqsubseteq \bot$
$C \sqsubseteq \leq 0 R.D$	$C \sqcap \exists R.D \sqsubseteq \bot$

Note that we have also included the $\bot$ concept in our definition in Table 2, which does not appear as part of the RL constructors. This special concept is easy to simulate in RL adding axioms $C_\bot \sqcap D_\bot \sqsubseteq \bot$ and $C_\bot \sqsubseteq D_\top$ where $C_\bot$ and $D_\bot$ are fresh concepts. We have that if K contains these axioms then $K \models C_\bot \equiv \bot$.

4 Formal Description of the Pattern

In Section 2, we introduced map scaling as seamless transfer of information in maps from one level of detail to another. This idea can be understood in terms of a *binary scaling function*. In this section, we will describe this idea first as a simple functional pattern (see Figure 1), illustrate it with examples, and then formalize it in DL such that it can published as ontology vocabulary (see Figure 2).

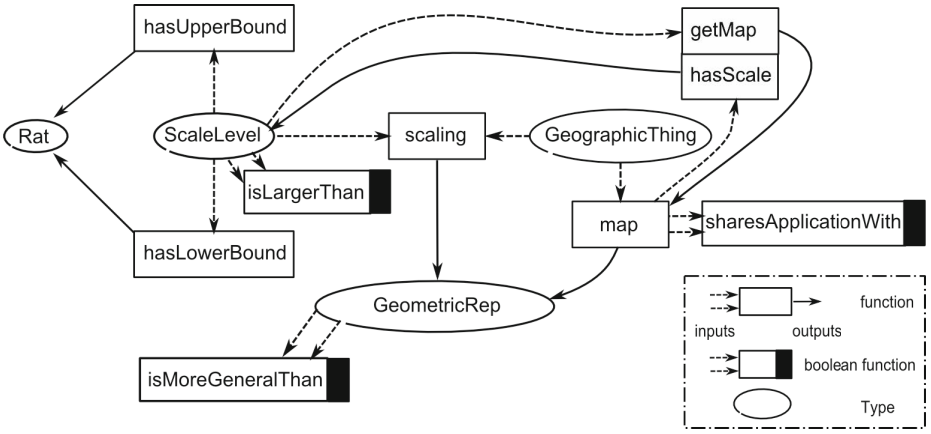


Fig. 1. The map scaling pattern in functional notation. Boxes denote functions, ellipses denote types of entities. Dotted arrows indicate input types and full arrows output types of functions. Relations are boolean functions.

The functional pattern in Figure 1 allows to quickly grasp the formal constraints in terms of functions (denoted as boxes) that map various input types to a single output (types denoted as ellipses). Along with the explanation of this pattern, the challenge addressed here is to translate these general functions into the previously described DL language $\mathrm{DLP}_{\exists}$, i.e., to translate Figure 1 into Figure 2[7].

Simply put, map scaling applications provide *more or less generalized geometric representations* of *geographic phenomena* for different *scale levels*. That is, we need to deal with the primitive types of things listed in Table 3. The formal relationship between these types, which is manifest in map scaling applications, can be captured by a *scaling function* and *two ordering relations*. The former can be understood as a binary function from geographic phenomena and scale levels into geometric representations. The latter account for scale dependent orderings of scale levels and data representations. The numeric scale/resolution boundaries which correspond to a single scale level are given by *hasUpperBound* and *hasLowerBound*, respectively. The scaling function is specific for a certain scaling application. Other relations (such as *hasScale*, *getMap* and *sharesApplicationWith*) and types (e.g. *MapData*)

[7] *General higher order functions*, which are used here as in functional programming, e.g. `http://www.cl.cam.ac.uk/research/hvg/Isabelle/`, are not formalizable in DL.

Table 3. Primitive types of entities involved in the functional pattern

Name	Formal Type	Explanation
Geographic phenomenon	*GeographicThing*	Type of mappable phenomenon, may be categorized by geo-ontologies and may be aggregated to *layers*
Geometric representation	*GeometricRep*	Type of data representation of a phenomenon which involves a geometric part (raster or vector representation) as well as a (map-) symbolization part
Scale level	*ScaleLevel*	Type of discrete level of detail
Scaling	*ScaleLevel* $\Rightarrow$ *GeographicThing* $\Rightarrow$ *GeometricRep*	Type of scaling function

can be based on this formal apparatus. For example, in this pattern, a *map* simply corresponds to a the projection of a scaling function to a fixed scale level.

The corresponding DL ontology pattern is described in Figure 2. Since DL does not support the specification of arbitrary functions, *scaling functions in our pattern are indirectly represented as subgraphs consisting of reified 3-tuples.* A *scaled representation* is a reified tuple of geographic thing, scale level and geometric representation. This work around requires the introduction of a new class *ScaledRep*, as well as its outgoing properties *representsObject*, *isScaled* and *isPresentedAs*, denoting the three slots in each 3-tuple. Maps need to be introduced as a primitive class, and we need to explicitly assert that they are constituted of scaled representations and have a single scale level. Furthermore, maps are assigned a single scale level which can only be shared within the same application. Our pattern also involves further formal constraints which will be discussed along the next paragraphs in terms of axioms.

Our pattern does not include concept hierarchies, because they are not primarily relevant for scaling. Scale/generalization orders and concept subsumption, even though related, are two separate things. However, the latter may be introduced through geo-ontologies of phenomena.

4.1 Maps and Applications

As mentioned above, we define maps as being constituted of scaled representations of phenomena at a fixed scale. This notion of a map captures the idea that maps are actually *semiotic signs*, i.e., they supply a unique cartographic representation of phenomena as referents.[8] In the ontology designed to fit the pattern a map can be seen as a set of scaled representation individuals.

[8] However, there are also other ways to capture the notion of cartographic maps. For example, from a GIS viewpoint, a map may be defined as a particular collection of map data.Or, from a cartographic point of view, maps may be seen as a kind of cartographic visualization or image.

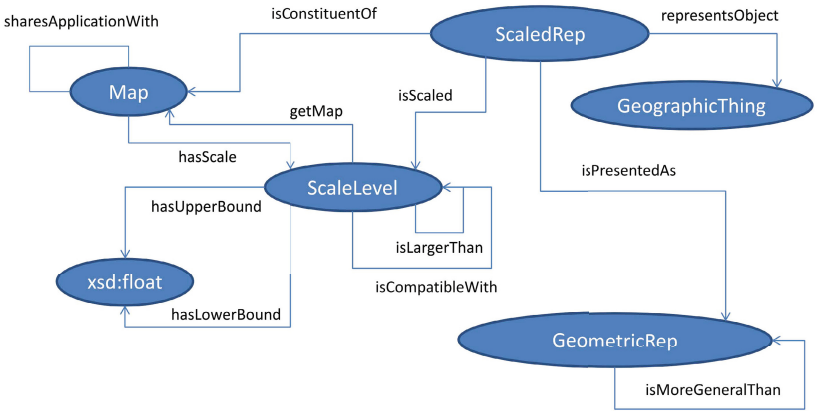


Fig. 2. The DL ontology design pattern corresponding to the functional map scale pattern. Ellipses stand for DL classes. Arrows represent DL properties that relate classes.

The maps that stem from a certain scaling function (the former being projections of the latter to fixed scale levels) are considered part of the same scaling application. This leaves open the possibility that there may be different scaling applications for the same set of geographic phenomena but for different purposes. We express this relation making use of the property *sharesApplicationWith* which relates those maps that are part of the same application based on whether they derive from the same scaling function. Property *sharesApplicationWith* is declared to be transitive (1), symmetric (2), and reflexive (3) with respect of the individuals within the class *Map*:

$$sharesApplicationWith \circ sharesApplicationWith \sqsubseteq sharesApplicationWith \quad (1)$$

$$sharesApplicationWith^{-} \sqsubseteq sharesApplicationWith \quad (2)$$

$$Map \sqsubseteq \exists sharesApplicationWith.Self \quad (3)$$

We define property *hasScale* as the function which delivers the unique scale level of a given map. This property is defined to be functional, since every map is associated with a single scale. We also define *getMap* as the inverse property of *hasScale* which allows us to retrieve the map associated with a given scale. These constraints are enforced using axioms (4) and (5).

$$\top \sqsubseteq \leq 1 hasScale.\top \quad (4)$$

$$hasScale^{-} \sqsubseteq getMap \quad (5)$$

We impose a less restrictive form of functionality over property *getMap*. We have that this property is functional over the set of maps that belong to the same application, which defined in our pattern as the set of maps that are connected through the *sharesApplicationWith*. Note that this property is declared to be both symmetric and transitive. I.e. a given scale s_1 cannot be shared by maps

m_1 and m_2 if we have that $sharesApplicationWith(m_1, m_2)$ is entailed by the ontology. We enforce this constraint with axiom 6 which automatically collapses into one single individual all maps associated to the same scale that are within the same application.

$$\top \sqsubseteq \leq 1(getMap \circ sharesApplicationWith).\top \tag{6}$$

Due to these constraints we have that for a given application, there is only one map at an specific scale level. The rationality behind this constraint is to eliminate all ambiguity at the time of representing map data at a given scale over the same application.

We have defined our ontology to allow for an easy retrieval of all the information pertaining to a single map. We can make use of the property connexion *isConstituentOf* to retrieve and query about all the existing scaled representations associated with a given map. Property *isScaled*, which links every scaled representation with the scale associated to the map this one belongs to, is automatically generated due to axiom (7).

$$isConstituentOf \circ hasScale \sqsubseteq isScaled \tag{7}$$

After imposing these restrictions over applications, maps and scales, we elaborate about *scale levels* and *geographic representations*, which are ordered in a chain-like manner.

4.2 Orders on Scale Levels and Geometric Representations

We enforce a strict partial order over properties *isLargerThan* and *isMoreGeneralThan* which respectively connect (and order) individuals over the classes *Scale* and *GeometricRep*.

$$isLargerThan \circ isLargerThan \sqsubseteq isLargerThan \tag{8}$$

$$\exists(isLargerThan \sqcap isLargerThan^{-}).\top \sqsubseteq \bot \tag{9}$$

$$isMoreGeneralThan \circ isMoreGeneralThan \sqsubseteq isMoreGeneralThan \tag{10}$$

$$\exists(isMoreGeneralThan \sqcap isMoreGeneralThan^{-}).\top \sqsubseteq \bot \tag{11}$$

As usual, a strict partial order is a binary relation that is irreflexive and transitive, and therefore antisymmetric. We enforce transitivity of properties *isLargerThan* and *isMoreGeneralThan* with axioms 8 and 10 respectively. Axioms 9 and 11 enforce irreflexivity of both properties[9] also enforcing antisymmetry.

Furthermore, we add a *similarity relation* among scale levels (i.e., one that is symmetric and reflexive), which allows us to connect compatible scales across different applications. Similarity among scale levels allows us to merge data from different applications, each having its separate scale level chain. Note that the computation of this similarity relation may be done in various ways based on their numerical scale boundaries, and we deliberately leave open in our pattern how this may be done. The similarity relation is represented in the pattern by the

[9] We have that $(\top \sqsubseteq \neg\exists R_1, Self) \equiv \top \sqsubseteq \exists(R_1 \sqcap R_1^{-}).\bot$ for any property R_1.

isCompatibleWith relation which is defined to be symmetric (12) and reflexive (13) connecting all individuals in class *ScaleLevel* with themselves.

$$isCompatibleWith^{-} \sqsubseteq isCompatibleWith \tag{12}$$

$$ScaleLevel \sqsubseteq \exists isCompatibleWith.Self \tag{13}$$

We assume that each scale level has one *upper and lower bound* in terms of numeric map scales or pixel resolutions. The latter are simply rational numbers. Every scale has at most one upper bound and one lower bound and therefore both properties *hasLowerBound* and *hasUpperBound* are declared functional. The defined constraint is enforced with axioms:

$$ScaleLevel \sqsubseteq \exists hasLowerBound.xsd{:}float \tag{14}$$

$$ScaleLevel \sqsubseteq \exists hasUpperBound.xsd{:}float \tag{15}$$

$$\top \sqsubseteq \leq 1 hasLowerBound.\top \tag{16}$$

$$\top \sqsubseteq \leq 1 hasUpperBound.\top \tag{17}$$

We skip the constraint that their order needs to comply with (or even defines) the scale level order. To improve the understanding of the publication we have not included datatypes within DLP$_{\exists}$, necessary to deal with algebraic operations over the *xsd:float* class. Once done it would not be difficult to verify than the existing *isLargerThan* relationships are valid, and to automatize the creation of this relationship between the existing scales.

Now we come to the most essential part of the pattern, namely representing the *scaling function*, compare Table 3 and Figure 1. The scaling function allows to switch to a new data representation for all phenomena by changing the scale level. This is done in a *monotonic* manner, i.e., such that the ordering of scale levels is preserved in generalization levels. Put differently, scaling to a larger scale level excludes the possibility that representations become more general. We include the possibility that a phenomenon is not represented at all at certain scale levels by creating an empty *GeometricRep* individual, as well as the possibility that it may be represented in a constant manner.

4.3 Monotonicity of Scaling

To enforce this constraint using OWL we make use of the class *ScaledRep*,and we add a logical equivalent to the following first order logic rule to the ontology:

$$\begin{aligned}
&sharesApplicationWith(m_x, m_y) \wedge hasScale(s_y, m_y) \wedge hasScale(s_x, m_x) \\
&\wedge isLargerThan(s_x, s_y) \wedge \\
&isConstituentOf(m_x, sr_x) \wedge isConstituentOf(m_y, sr_y) \wedge \\
&representsObject(sr_x, g) \wedge representsObject(sr_y, g) \wedge \\
&isPresentedAs(sr_x, grr_x) \wedge isPresentedAs(sr_y, grr_y) \wedge \\
&isMoreGeneralThan(grr_x, grr_y) \rightarrow \bot(m_x)
\end{aligned}$$

This rule enforces that the ontology becomes inconsistent if

- there exist maps m_1 and m_2 belonging to the same application with scales s_1 and s_2,
- scale s_1 is larger than scale s_2,
- maps m_1 and m_2 contain scaled representations sr_1 and sr_2 that represent the same geographic thing g, and
- the geographic representation record grr_1 for sr_1 is more general than the one for sr_2, namely grr_2.

Although this rule may look quite complex it is indeed expressible in OWL. An automatized way of performing this transformation is presented in [3], as well as a procedure to check if a given rule is indeed expressible in OWL. A possible set of DL axioms equivalent to the previous rule is:

$$\begin{aligned}
hasScale^{-} \circ sharesApplicationWith \circ hasScale &\sqsubseteq R_1 \\
R_1 \sqcap isLargerThan &\sqsubseteq R_2 \\
isScaled \circ R_2 \circ isScaled &\sqsubseteq R_3 \\
isPresentedAs \circ isMoreGeneralThan^{-} \circ isPresentedAs^{-} &\sqsubseteq R_4 \\
representsObject \circ representsObject^{-} &\sqsubseteq R_5 \\
R_3 \sqcap R_4 \sqcap R_5 &\sqsubseteq R_{\bot} \\
\exists R_{\bot}.\top &\sqsubseteq \bot
\end{aligned}$$

where all R_i are freshly introduced roles that do not appear previously in the ontology.

4.4 Functionality of Scalings

Next, we enforce *functionality* constraints on the subgraph of scaled representations which denotes a scaling function, i.e., a singular scaling application.

First, we make use of simple OWL axioms to enforce functionality for the properties *isPresentedAs*, *isScaled*, and *representsObject*, which respectively connect a scaled representation with the geometric representation, the scale, and the geographic thing it is associated with.

$$\top \sqsubseteq \leq 1 isPresentedAs.\top \tag{18}$$

$$\top \sqsubseteq \leq 1 isScaled.\top \tag{19}$$

$$\top \sqsubseteq \leq 1 representsObject.\top \tag{20}$$

Second, a scaled representation is enforced to have a geometric representation, a scale, and a geographic thing associated to it. This is enforced using OWL axioms:

$$ScaledRep \sqsubseteq \exists isPresentedAs.GeometricRep \tag{21}$$

$$ScaledRep \sqsubseteq \exists isScaled.ScaleLevel \tag{22}$$

$$ScaledRep \sqsubseteq \exists representsObject.GeographicThing \tag{23}$$

And third, since the relation expressed by scaled representations stands for a scaling function, it needs to be restricted to be *functional* with respect to a scale level and a geographic phenomenon represented. Furthermore, since a scaling function corresponds to a particular scaling application, we need to restrict functionality to only those scaled representations that are part of a single scaling application.

Since every scale is only associated to one map within the same application we only need to verify that there only exists one scaled representation for each geographic phenomenon. This constraint is enforced with axioms (24) and (25).

$$isConstituentOf^{-} \circ representsObject \circ representsObject^{-} \sqsubseteq R_{aux} \tag{24}$$

$$\top \sqsubseteq \leq 1(R_{aux} \sqcap isConstituentOf^{-}).\top \tag{25}$$

Due to axioms (24) and (25) we have that two different scaled representations are collapsed into a single one if they are constituents of a given map and represent the same object. Therefore we are guaranteed that, within the same map there only exists one scaled representation representing the same geographic thing. Given that functionality only needs to be enforced across the maps within the same application, and that by previous restrictions we have that there are no two maps with the same scale within the same application we have that the mapping to a geometric representation is functional depending on the specific geographic thin represented and the scale for any given scale representation.

We show a translation of the axioms presented in this section in Appendix, which is part of the technical report of the paper. In the translation we also establish a set of domain and range restrictions based on the relations between properties and roles in Figure 2. As further constraints, we also declare all classes defined in the pattern to be disjoint in order to avoid possible mistakes in the declaration of individuals.

5 Application Scenario

When interpreting and comparing maps such as the global kernel density map (raster data)[10] in Fig. 3, where the color ramp of each pixel represents the *fraction of malaria transmitting mosquitos* [7], it is crucial to take into account the effect of scale. At its original resolution [11], each single pixel covers a width of approximately 17 kilometers at the equator. Assuming that the malaria data is displayed at a screen pixel size of 0,25 mm, the map image has an appropriate maximal scale of 1:68000000. Thus, for instance, using such a map to determine whether a particular village (many of which could be contained in a single pixel) is affected to a higher degree than others is difficult, since villages are represented on much larger scale levels.

[10] ©2010 Malaria Atlas Project, available under the Creative Commons Attribution 3.0 Unported License.

[11] Keeping in mind what we said about zooming and scale in digital images.

Similarly, as mosquitos require water for reproduction, one may be tempted to combine such a map with a river network layer. As in the case before, this particular map is too coarse to support a meaningful comparison, since river width is a fraction of the size of a pixel. Instead, scholars have to go back to the source data and generate raster data at a scale which is appropriate for the scale of river networks. While this is a simple operation for Geographic Information Systems (GIS), and may be automatized in terms of a map scaling service, the scale increase cannot be chanced ad libitum but has also a lower bound. While the upper bound is constrained by the process of cartographic abstraction mentioned before, the lower bound is limited by the measurement procedure. In this specific case, data was coded at the village level during the field study, and thus representing the data at a larger scale of, say, 1:2500, would create a misleading impression of accuracy.

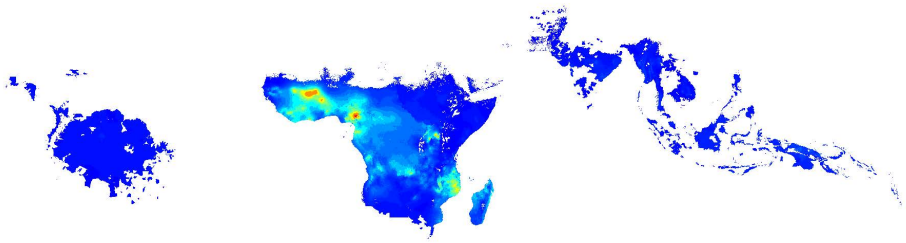


Fig. 3. Global kernel density map of the fraction of malaria infected mosquitos. The spectrum blue-yellow-orange-red denotes an increase of this fraction.

The question that we address here is whether the decision about appropriateness of scaled representations of phenomena can be automated in the Semantic Web, independently from and without (manual) interaction with particular scaling applications. In the following, we demonstrate how such a decision may be computed based on our pattern.

Suppose we have an *ABox* which describes map data from different scaling applications of the kind discussed above together with its scale level:

$$\begin{aligned}
&\mathit{ScaleLevel}(s1) && \mathit{GeographicThing}(\mathit{malaria})\\
&\mathit{GeometricRep}(\mathit{raster}) && \mathit{ScaledRep}(\mathit{sr1})\\
&\mathit{representsObject}(\mathit{sr1}, \mathit{malaria}) && \mathit{isScaled}(\mathit{sr1}, \mathit{s1})\\
&\mathit{isPresentedAs}(\mathit{sr1}, \mathit{raster}) &&\\
&\mathit{ScaleLevel}(\mathit{s2}) && \mathit{Village} \sqsubseteq \mathit{GeographicThing}\\
&\mathit{GeometricRep}(\mathit{polygon}) && \mathit{Village}(\mathit{village1})\\
&\mathit{representsObject}(\mathit{sr2}, \mathit{village1}) && \mathit{ScaledRep}(\mathit{sr2})\\
&\mathit{isPresentedAs}(\mathit{sr2}, \mathit{polygon}) && \mathit{isScaled}(\mathit{sr2}, \mathit{s2})\\
&\mathit{Map}(\mathit{m1}) && \mathit{Map}(\mathit{m2})\\
&\mathit{isConstituentOf}(\mathit{sr1}, \mathit{m1}) && \mathit{isConstituentOf}(\mathit{sr2}, \mathit{m2})\\
&\mathit{isCompatibleWith}(\mathit{s2}, \mathit{s1}) &&
\end{aligned}$$

The data provided may come from different users that uploaded data from different applications. We assume there is one user that wants to merge existing information about the malaria and existing villages. We define a new subclass *Village* of the general class *GeographicThing*. The user can now query for all existing villages that are represented on a compatible scale together with *sr2*, which is the scaled representation of malaria.

$$(x?) : \exists representObject.Village \sqcap \exists(isScaled \circ isCompatible \circ isScaled^{-}).\{sr2\}$$

The query will retrieve all scaled representations that represent a *Village* type individual and have a compatible scale with *sr2*. The user can then select the most appropriate for his visualization of the data. Making use of the *xsd:float* values associated with a scale allows also to retrieve the scaled representations within a certain range.

6 Discussion and Conclusion

In this paper, we suggested a formal ontology design pattern that describes cartographic map scaling on a semantic level in terms of a functional relationship among geometric representations, phenomena and scale levels. Map scaling applications are used on the web to represent and display phenomena at different scale levels. In the Semantic Web, the notions of *scale* and *resolution* have, to the best of our knowledge, not been introduced so far, even though they are of central importance to deal with information at different levels of granularity. Scale, granularity and resolution are central notions of cartography [14] and GIScience [13], however, formal approaches to describe map scaling are mostly focused on mathematical models of generalization and granularity change [19], not on making publicly available the application logic of actual scaling systems.

In the Web, granularity levels are needed to improve performance of querying, reasoning, as well as in order to display information meaningfully on a map. The challenge lies in preventing mashups of data at inappropriate resolutions, or visual clutter across scaling applications, as well as in enabling the tracing of geographic phenomena across different levels of detail and across different applications. Opening up existing scaling implementation logic for the Semantic Web not only allows cross-linking web map services based on geographic phenomena, it also has the potential to make the Semantic Web itself scale across different levels of detail. This is because it adds the crucial information about whether certain information can be used on certain scale levels or not. This information today seems to be missing from the Semantic Web. For example, the geographic reference of DBpedia or Linked Geodata [18] is a scale-free coordinate point.

In the paper, we proposed formal constraints to the pattern in a tractable fragment of DL, which can be used to compute inferences on *ABox* descriptions of actual scaling applications. For example, we showed that it is possible to check automatically whether data representations from scaling applications are compatible with respect to their scale levels, and thus, can be meaningfully displayed in a single map. The constraints also allow to check consistency of a

single scaling application, e.g., with respect to monotonicity and functionality of scaling. Future research may enrich the axiomatization based on a full functional specification in HOL, which could only be sketched in Figure 1. It may also address scalable reasoners for $\text{DLP}_{\exists}$, which would allow testing the pattern on a set of scaling applications described by the pattern. Even without computational reasoning, the pattern can be directly used to annotate and query existing scaling applications based on RDF. Furthermore, the pattern may be specialized by complementary patterns describing geometric data formats as known in GI-Science, different geographic ontologies, the relation of map displays and scale, as well as different notions of scale and resolution.

Acknowledgements. This work is a collaborative outcome of the GeoVoCampDayton2012[12]. Some of the authors credit funding from European Commission (ICT-FP7-249120 ENVISION project), as well as the German Research Foundation (Research fellowship grant DFG SCHE 1796/1-1). Authors from Wright State University acknowledge funding from by the National Science Foundation under award 1017225 III: Small: TROn – Tractable Reasoning with Ontologies.

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications, 2nd edn. Cambridge University Press (2007)
2. Bittner, T., Donnelly, M., Smith, B.: A spatio-temporal ontology for geographic information integration. Int. J. Geogr. Inf. Sci. 23(6), 765–798 (2009)
3. Carral Martínez, D., Hitzler, P.: Extending description logic rules. In: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 345–359. Springer, Heidelberg (2012)
4. Degbelo, A., Kuhn, W.: A Conceptual Analysis of Resolution. In: XIII GEOINFO, Brazilian Symposium on Geoinformatics, Campos do Jordão, Sao Paolo, Brazil, pp. 11–22 (2012)
5. Frank, A.: Scale is introduced in spatial datasets by observation processes. In: Spatial Data Quality From Process to Decision (6th ISSDQ 2009), pp. 17–29. CRC Press (2009)
6. Gangemi, A., Presutti, V.: Towards a pattern science for the semantic web. Semantic Web 1(1-2), 61–68 (2010)
7. Gething, P., Patil, A., Smith, D., Guerra, C., Elyazar, I., Johnston, G., Tatem, A., Hay, S.: A new world malaria map: Plasmodium falciparum endemicity in 2010. Malaria Journal 10(1), 378 (2011)
8. Goodchild, M.F., Proctor, J.: Scale in a digital geographic world. Geographical and Environmental Modelling, 5–23 (1997)
9. Grosof, B., Horrocks, I., Volz, R., Decker, S.: Description logic programs: combining logic programs with description logic. In: Proc. 12th Int. Conf. on World Wide Web (WWW 2003), pp. 48–57. ACM (2003)
10. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible $\mathcal{SROIQ}$. In: Proc. of the 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2006), pp. 57–67. AAAI Press (2006)

[12] `http://vocamp.org/wiki/GeoVoCampDayton2012`

11. Kazakov, Y.: Saturation-Based Decision Procedures for Extensions of the Guarded Fragment. Ph.D. thesis, Universität des Saarlandes, Saarbrücken, Germany (March 2006)
12. Krötzsch, M., Simancik, F., Horrocks, I.: A description logic primer. CoRR abs/1201.4089 (2012)
13. Kuhn, W.: Core Concepts of Spatial Information: A First Selection. In: XII GEOINFO, Campos do Jordão, Brazil, November 27-29, pp. 13–26 (2011)
14. McMaster, R.B., Shea, K.S.: Cartographic Generalization in a Digital Environment: A Framework for implementation in a GIS. In: GIS/LIS 1988, San Antonio, Texas, USA, pp. 240–249 (1988)
15. Müller, J., Lagrange, J., Weibel, R.: GIS and Generalization: Methodology and Practice. Taylor and Francis (1989)
16. Montello, D.: Scale in Geography. In: International Encyclopedia of the Social and Behavioral Sciences, pp. 13501–13504 (2001)
17. Scheider, S., Kuhn, W.: Affordance-based categorization of road network data using a grounded theory of channel networks. International Journal of Geographical Information Science 24(8), 1249–1267 (2010)
18. Stadler, C., Lehmann, J., Höffner, K., Auer, S.: LinkedGeoData: A core for a web of spatial open data. Semantic Web (2012)
19. Stell, J., Worboys, M.: Stratified map spaces: A formal basis for multi-resolution spatial databases. In: SDH 1998 Proceedings 8th International Symposium on Spatial Data Handling, pp. 180–189 (1998)
20. Wu, J., Li, H.: Concepts of scale and scaling, pp. 3–15. Springer (2006)

Locking for Concurrent Transactions on Ontologies

Stefan Scheglmann, Steffen Staab, Matthias Thimm, and Gerd Gröner

WeST – Institute for Web Science and Technologies
University of Koblenz-Landau
56070 Koblenz, Germany
{schegi,staab,thimm,groener}@uni-koblenz.de

Abstract. Collaborative editing on large-scale ontologies imposes serious demands on concurrent modifications and conflict resolution. In order to enable robust handling of concurrent modifications, we propose a locking-based approach that ensures independent transactions to simultaneously work on an ontology while blocking those transactions that might influence other transactions. In the logical context of ontologies, dependence and independence of transactions do not only rely on the single data items that are modified, but also on the inferences drawn from these items. In order to address this issue, we utilize logical modularization of ontologies and lock the parts of the ontology that share inferential dependencies for an ongoing transaction. We compare and evaluate modularization and the naive approach of locking the whole ontology for each transaction and analyze the trade-off between the time needed for computing locks and the time gained by running transactions concurrently.

1 Introduction

Ontologies, as a prominent knowledge representation approach on the Web, are often collaboratively developed, distributed and extended by multiple users. In general, users modify ontologies independently from each other and they are not aware of edits of other users. Accordingly, approaches for enabling concurrent editing of large ontologies have to ensure that modifications of users are not contradicting each other. Concurrent ontology editing and knowledge base authoring has been the topic of several previous works, which can be roughly partitioned into two categories. First, optimistic versioning-based approaches, like in Karp et al. [10] or ContentCVS by Ruiz et al. [9], make users feel as in a single-user setting — by distinguishing between a private (editable) knowledge base and a public version users can only commit their changes to. In general, commits in these systems consist of multiple changes and these systems provide conflict resolution functionalities. Second, systems like [15] address conflict resolution for parallel editing over a Web interface. The latter systems usually focus more on the social component by making simultaneous changes of different users possible and showing them immediately to all users. In terms of time spans

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 94–108, 2013.

between two commits/edits of a single user, these two categories are the endpoints of a wide spectrum of approaches for dealing with concurrent knowledge base and ontology editing. However, the trade-off between "isolated" access and interleaving operations is also studied in traditional transaction management for databases, which is the foundation for the approach to deal with concurrent access to ontologies. There, sophisticated access methods and protocols avoid unwanted intermediate results and guarantee a consistent synchronization between users. This is achieved by introducing transactions and specific means for handling them. A transaction is defined by an opening statement ('begin of transaction'), some arbitrary program code that includes interactions with the database and a conclusion statement, i. e., either a 'commit' that finalizes the transaction or an 'abort' that erases all effects of this transaction. Using a transaction, the individual user should be shielded from influences of other users. The easiest way to achieve such isolation would be a strict serial execution of all transactions. Because individual transactions, however, may contain time-consuming user code, the parallel execution of transactions seems to be a necessity for the performance of the system. Trading off between users' wishes for isolation from effects of other users led to the notion of serializability [2]. If transactions are scheduled in a—typically interleaved—way that is equivalent to some serial schedule of the same transactions, then the schedule is called *serializable* and the program code defining the transactions behaves functionally as if it has exclusive access to the database. Obviously, such a scheme is not only desirable to have for databases but is also highly desirable to have in the case of frequently accessed ontologies. However, there arise several issues that need to be tackled to carry over the notions of 'transaction' and 'serializability' from databases to ontologies: (1)The notion of 'serializability' is based on the notion of 'equivalence' of transaction schedules, but what does it mean that two schedules are equivalent if also computational inferences in ontologies need to be accounted for? (2)As will be shown below, 'serializability' is typically based on locking data items such that different transactions do not interact with each other. But what should be locked when logical inference comes into play? (3)Locking data items for transaction scheduling is beneficial as the actual locking process is computationally cheap. However, in the context of ontologies computing the axioms to be locked may become computationally expensive. What is the trade-off between concurrency of ontology access and determining the locks for transactions on an ontology? To illustrate the above challenges, we consider the following example:

Example 1. Let $\mathcal{O} = (\mathcal{T}, \mathcal{A})$ be an ontology with the following axioms in the $\mathcal{T}$-Box:

$$A_1 \equiv \forall R.D_1 \quad (1) \qquad A_2 \equiv \forall R.D_2 \quad (2) \qquad D_1 \sqcap D_2 \sqsubseteq \bot \quad (3)$$
$$B \sqsubseteq D_1 \quad (4) \qquad A \sqsubseteq \forall R.B \quad (5)$$

and some arbitrary ABox $\mathcal{A}$. Assume that one user intends to replace Axiom (4) $B \sqsubseteq D_1$ by a new Axiom (6) $B \sqsubseteq D_2$. Imagine a second user is asking (at the same time) for all concepts that subsume A. Before the change of the first

user (replacement of Axiom (4) by (6)), concept A_1 subsumes A, but after the change, A_2 subsumes A. Before the first user starts the transaction the result to the second user's query would be $A \sqsubseteq A_1$ and afterwards $A \sqsubseteq A_2$. However, the relationship does not hold after the first user has deleted Axiom (4) and not yet added Axiom (6), the result would be neither $A \sqsubseteq A_1$ nor $A \sqsubseteq A_2$.

In this paper, we present a locking-based framework for handling concurrent transactions on ontologies. We define a notion of conflict that prevents different transactions to be executed in an arbitrary way (Sect. 4) and adapt a two-phase locking approach from databases [3] (Sect. 5). Whenever a user issues some operation the necessary locks are acquired. For computing the locking areas for transactions, we utilize the modules of an ontology [6].

2 Foundations and Related Work

The first part of this section introduces fundamentals on concurrent transactions and locking principles, rooted in the database research field. The second part gives an overview on related work of concurrent ontology editing.

2.1 Foundations of Transaction and Locking

Transaction management guarantees the isolation of a transaction execution from the inference with other transactions. Transactions in databases ensure the following properties [2]: *Atomicity:* A transaction is either completely executed or not executed; *Consistency:* The execution of a transaction has to maintain the consistency of a database; *Isolation:* The execution of a set of transactions has the same effect as all transactions would be executed individually; *Durability:* After executing a transaction, all modifications need to be stored in the database. Technically, isolation can be ensured by *serializability*, which guarantees that the outcome of a schedule is equal to the outcome of the same transactions executed one after the other. Such a schedule is called *serializable*. The serializability is guaranteed by concurrency control mechanisms like locking, e. g., the *two-phase locking (2PL)* [3], where data of potential competing transactions are locked in two phases: In the 'expanding phase', the transaction successively tries to acquire locks for the resources of each single *atomic operation*. If it successfully acquires a lock then it performs the operations and continues. If the resource of an operation is already locked by another transaction, the current transaction will stop and consecutively try to acquire a lock for this resource until it succeeds. After all operations of an transaction are performed, the transaction will enter the 'shrinking phase' and free all of its locks.

Following this line of argumentation, a key issue is to determine the resources that need to be locked in order to execute an atomic operation. Obviously, the locked area should be as small as possible to enable interleaving transactions, but the area should be as large as necessary to avoid conflicts.

2.2 Concurrent Ontology Editing

The need for concurrency control in knowledge bases was already acknowledged by Chaudhri et al. [4]. They show the inadequacy of concurrency control mechanism from databases and present Dynamic Directed Graph (DDG), a concurrency control mechanism for rule-based knowledge bases. Their setting and approach is similar to ours but use a very restrictive knowledge representation formalism which simplifies transaction schedule computations.

Other approaches can be roughly partitioned into two categories. The first category [10,9] extends versioning systems to the knowledge base setting and implement an optimistic conflict resolution schema. The second category [15] applies ideas of online editors to the field of collaborative ontology editing, without considering issues of conflict resolution directly. The rationale behind using these two approaches base on different assumptions. For the first category, it is assumed that knowledge bases are created over a large period of time. For both, it is assumed that the areas of responsibility of different contributors are relatively independent, i. e. they usually modify different parts of the knowledge base. However these assumptions do not necessarily hold in many of application areas, where e. g. already deployed ontologies are modified more frequently. In this paper, we focus on scenarios that need not satisfy these assumptions. Nonetheless, we now look at some of these approaches in more detail.

In [10], Karp et al. introduced an authoring tool for knowledge bases based on frame logic, a predecessor of modern ontology languages. Along with the collaborative subsystem they define the notion of conflicts regarding knowledge base operations and they provide conflict detection mechanisms for the merge process. A similar approach is pursued in ContentCVS [9]. The authors adopted the popular concurrent versioning approach CVS to the field of collaborative ontology development. They include structural and semantic-based conflict detection and state-of-the-art ontology debugging and repair techniques to help the user in conflict resolution. Both approaches make use of an optimistic versioning based approach which detects and resolves conflicting edits in commits on merge time without locking. In [15], Tudorache et al. evaluate the collaboration features of WebProtégé within an intense user study during the development process of the 11th version of the International Classification of Diseases (ICD-11). WebProtégé's collaborative features are all directly integrated in the editing process and make all users aware of all edits currently happening. Additional WebProtégé provides features for incorporating, tracking and reviewing changes on-the-fly. This way of collaborative ontology editing is focusing on conflict prevention or just-in-time conflict resolution. To provide the users of such an editor with useful information about possible conflicts resulting from their edits, an approach similar to our approach could be facilitated. In such a setting our approach would not lock resources but make users aware of possible conflicts calculate from the current edits.

For further related work, Falconer et al. [5] describe patterns of editing behavior and roles of the contributors for large scale ontology-development projects. This is of particular interest for the design and implementation of collaborative

editing environments for ontology. The concurrency control mechanism, described in this paper, builds the basis for such systems and the calculation of areas affected by a transaction might benefit from contributor roles and predefined behavior patterns.

For OWL ontologies, Seidenberg and Rector [13] discuss basic principles for multi-user ontology editing. They indicate that due to *inference capabilities* the computation of locking areas goes beyond transaction management principles in databases since changes of a class might lead to different subsumptions of *other* classes, for instance: (i) classes with different names are classified as equal; (ii) a class is classified as a new subclass of a new/changed class; (iii) a class might become unsatisfiable. In this paper, we tackle this indicated challenge of computing locking areas for transaction management.

In order to handle locking, it is necessary to identify areas that are affected by a transaction. Subsequently, we call such areas of an ontology the *area of influence* of a single operation. These areas are obtained by computing modules, either in terms of structural areas, which are built by traversal techniques [14,11], or in terms of semantic influence areas [6], as it is used in our work.

3 Preliminaries

In this section, we introduce description logics [1], the language family that underlies modern ontology languages like OWL2 [8]. For purpose of presentation, we refer to $\mathcal{ALC}$, but our approach can be generalized to any other description logic where module computation [6] is supported. The signature $Sig_{\mathcal{L}} = \mathbf{C} \uplus \mathbf{R} \uplus \mathbf{I}$ of $\mathcal{L}$ is composed of a set $\mathbf{C}$ of atomic concepts denoted by $A, B, C, \ldots$, a set $\mathbf{R}$ of atomic roles denoted by $r, s, \ldots$, and a set $\mathbf{I}$ of individuals denoted by $a, b, c, \ldots$, and subsets of $Sig_{\mathcal{L}}$ are denoted $\mathbf{S}, \mathbf{S}_1, \mathbf{S}_2, \ldots$. Concepts in $\mathcal{L}$ are built using the symbols in $Sig_{\mathcal{L}}$ and the following syntax rules:

$$C ::= A \mid \top \mid \bot \mid (\neg C) \mid (C \sqcap C) \mid (C \sqcup C) \mid (\exists\ r.C) \mid (\forall\ r.C) \mid$$

where $A \in \mathbf{C}$ is a concept name, $r \in \mathbf{R}$ is a role name and $a_1, \ldots, a_n \in \mathbf{I}$ are individuals. If C_1, C_2 are concepts then $C_1 \sqsubseteq C_2$ is an *inclusion axiom*. If C is a concept, $r \in \mathbf{R}$ is a role, and $a, b \in \mathbf{I}$ are individuals, then $C(a)$ and $r(a, b)$ are *assertional axioms*. An *ontology* $\mathcal{O}$ is a pair $\mathcal{O} = (\mathcal{T}, \mathcal{A})$ where $\mathcal{T}$ is a finite set of inclusion axioms (called the Tbox) and $\mathcal{A}$ is a finite set of assertional axioms (called the Abox). The signature $Sig(\mathcal{O})$ of an ontology $\mathcal{O}$ is the set $Sig(\mathcal{O}) \subseteq Sig_{\mathcal{L}}$ of symbols occurring in $\mathcal{O}$. The signature $Sig(\alpha)$ of an axiom α is defined analogously. If $\mathcal{O} = (\mathcal{T}, \mathcal{A})$ is an ontology and α is an axiom we define $\mathcal{O} \cup \{\alpha\}$ to be either $\mathcal{O} \cup \{\alpha\} = (\mathcal{T} \cup \{\alpha\}, \mathcal{A})$ or $\mathcal{O} \cup \{\alpha\} = (\mathcal{T}, \mathcal{A} \cup \{\alpha\})$, depending on whether α is an inclusion or assertional axiom. The set difference is defined analogously. We assume the standard first-order semantics of $\mathcal{O}$, given by Tarski style model-theoretic semantics using interpretations like in [1].

4 Transactions on Ontologies

In this section, we illustrate the problem of concurrent transaction management for ontologies and specify the notion of atomic operations, transactions, transaction schedules and serializability. By a slight abuse of the notation, we use standard set operators in the context of sequences, e. g., $a \in (a_1, \ldots, a_n) \iff a \in \{a_1, \ldots, a_n\}$. The union $\cup$ of two sequences is defined as the set $(a_1, \ldots, a_n) \cup (b_1, \ldots, b_m) = \{a_1, \ldots, a_n, b_1, \ldots, b_m\}$ and the concatenation $\circ$ of two sequences is defined as the sequence $(a_1, \ldots, a_n) \circ (b_1, \ldots, b_m) = (a_1, \ldots, a_n, b_1, \ldots, b_m)$.

Definition 1. *Let $\mathcal{O}$ be an ontology in language $\mathcal{L}$ and $\mathbf{V}_C$, $\mathbf{V}_R$ and $\mathbf{V}_I$ be sets of variable names for concepts, roles and individuals. Then an atomic operation a on $\mathcal{O}$ is a tuple $a = (o, \alpha)$, consisting of one operation $o \in \{ask, tell, forget\}$ and an axiom α with 1.) $\alpha \in \mathcal{L}'$ with $Sig_{\mathcal{L}'} = (\mathbf{C} \cup \mathbf{V}_C) \uplus (\mathbf{R} \cup \mathbf{V}_R) \uplus (\mathbf{I} \cup \mathbf{V}_I)$ (if $o = ask$), 2.) $\alpha \in \mathcal{L}$ (if $o = tell$), or 3.) $\alpha \in \mathcal{O}$ (if $o = forget$).*

For an atomic operation (ask, α), we allow α to contain variable names in order to ask for more general formulas, e. g., the operation $(ask,\ A \sqsubseteq ?X)$ or $(ask,\ ?Y \sqsubseteq B)$ with $?X, ?Y \in \mathbf{V}_C$, asking for axioms with concept descriptions C such that $\mathcal{O} \models A \sqsubseteq C$ is true or for all axioms with concept descriptions D that $\mathcal{O} \models D \sqsubseteq B$ is true, respectively. Hence, an empty result to an *ask* operation means *false* whereas some result would mean *true*. The operation $(forget, \alpha)$ triggers a *contraction* of $\mathcal{O}$ by $\alpha \in \mathcal{O}$ yielding a new ontology $\mathcal{O}' = \mathcal{O} \setminus \{\alpha\}$. The operation $(tell, \alpha)$ triggers an expansion of $\mathcal{O}$ by α yielding a new ontology $\mathcal{O}' = \mathcal{O} \cup \{\alpha\}$. Note that we do not consider the general problem of complex belief dynamics in ontologies [12]. For example, we do not consider the problem of *revising* an ontology by a possibly contradicting axiom α such that the new ontology remains consistent. To formalize the above intuition, we introduce two functions that describe the results of an atomic operation. *ans*—answer, returns a set of axioms for a given pair of ontology and atomic operation—and *upd*—update returns a new (updated) ontology, for a given pair of ontology and atomic operation. For $\alpha \in \mathcal{L}'$ with $Sig_{\mathcal{L}'} = (\mathbf{C} \cup \mathbf{V}_C) \uplus (\mathbf{R} \cup \mathbf{V}_R) \uplus (\mathbf{I} \cup \mathbf{V}_I)$ let $gr(\alpha)$ be the set of *groundings* of α in $\mathcal{L}$, i. e., the set of all axioms that are the same as α but every variable is substituted by some concept description, role description, or individual. Let $\mathcal{O}$ be an ontology and $a = (o, \alpha)$ an atomic operation. Then define

$$ans(\mathcal{O}, (o, \alpha)) = \begin{cases} \{\alpha' \in gr(\alpha) \mid \mathcal{O} \models \alpha'\} & o = ask \\ \emptyset & \text{otherwise} \end{cases}$$

$$upd(\mathcal{O}, (o, \alpha)) = \begin{cases} \mathcal{O} & o = ask \\ \mathcal{O} \cup \{\alpha\} & o = tell \\ \mathcal{O} \setminus \{\alpha\} & o = forget \end{cases}$$

Note that only the *ask* operation may yield a non-empty answer and only *tell* and $forget$ operations actually update the ontology. Based on these *atomic operations*, we are able to define *ontology transactions* as follows.

Definition 2. *An* ontology transaction θ *(or* transaction *for short) is a finite sequence $\theta = (a_1, \ldots, a_n)$ of atomic operations $a_1, \ldots, a_n$.*

A transaction bundles a sequence of atomic operations to be executed on behalf of a user. For a transaction $\theta = ((o_1, \alpha_1), \ldots, (o_n, \alpha_n))$ let $\mathsf{axioms}(\theta) = \{\alpha_1, \ldots, \alpha_n\}$. We denote with $Sig(\theta) \subseteq Sig_{\mathcal{L}'}$ the signature of all axioms of a transaction θ, with $\mathcal{L}'$ being the same as in Definition 1.

For an ontology $\mathcal{O}$ and a sequence of atomic operations $(a_1, \ldots, a_n)$, in order to take cumulative changes of $\mathcal{O}$ into account, we abbreviate

$$upd(\mathcal{O}, ()) = \mathcal{O} \tag{1}$$

$$upd(\mathcal{O}, (a_1, \ldots, a_n)) = upd(upd(\mathcal{O}, (a_1, \ldots, a_{n-1})), a_n) \tag{2}$$

for all $i = 1, \ldots, n$. In other words, $upd(\mathcal{O}, (a_1, \ldots, a_n))$ is the ontology resulting after sequentially executing the atomic operations $a_1, \ldots, a_n$. Analogously, $ans(\mathcal{O}, a_n)$ is the answer of the atomic operation a_n on $upd(\mathcal{O}, (a_1, \ldots, a_{n-1}))$.

$$ans(\mathcal{O}, a_n) = ans(upd(\mathcal{O}, (a_1, \ldots, a_{n-1})), a_n) \tag{3}$$

Example 2. To clarify this, we continue with formalizing our Example 2 from the introduction according to the definitions made so far. Let $\theta_1 = (a_1, a_2), \theta_2 = (b_1)$ be the two transactions on $\mathcal{O}$ defined as:

$$a_1 = (forget, B \sqsubseteq D_1) \qquad a_2 = (tell, B \sqsubseteq D_2)$$
$$b_1 = (ask, A \sqsubseteq ?X)$$

As defined in the introduction, transaction θ_1 intends to replace the axiom $B \sqsubseteq D_1$ by $B \sqsubseteq D_2$ while transaction θ_2 asks for all subsumption relations of the form $A \sqsubseteq ?X$. Figure 1 shows the interaction between these two transactions. The left part of the figure shows transaction θ_1, while the right part shows the three possible execution orders of transactions θ_1 and θ_2. As we can see, depending on the transaction order, the outcome differs. The outcome of the operation of $b_1 = (ask, (A \sqsubseteq ?X))$ is $A \sqsubseteq A_1$ if the operation takes place before and $A \sqsubseteq A_2$ after the operations of θ_1, a_1, a_2. Both cases refer to a serial schedule. However, in the second case, we observe unintended answers of transaction θ_2. Since the operation (b_1) takes place in between the operations a_1 and a_2 (non-serial schedule) the only concept that subsumes A is the universal concept $\top$.

Definition 3. *Let $\Theta = \{\theta_1, \ldots, \theta_k\}$ with $\theta_i = (a_{i,1}, \ldots, a_{i,m_i})$ for $i = 1, \ldots, k$ be a set of transactions. A* transaction schedule *π of Θ is a transaction $\pi = (c_1, \ldots, c_m)$ such that*

$$\{c_1, \ldots, c_m\} = \theta_1 \cup \theta_2 \cup \ldots \cup \theta_k \tag{4}$$

and for all $i = 1, \ldots, k$ we have $u < r$ iff $s < t$ for $c_u = a_{i,s}$ and $c_r = a_{i,t}$.

Definition 4. *Let $\Theta = \{\theta_1, \ldots, \theta_k\}$ be a set of transactions. A transaction schedule π is a* serial transaction schedule *of Θ if there is a permutation $\sigma : \{1, \ldots, k\} \to \{1, \ldots, k\}$ such that*

$$\pi = \theta_{\sigma(1)} \circ \theta_{\sigma(2)} \circ \ldots \circ \theta_{\sigma(k)} \tag{5}$$

Let Π_{ser_Θ} be the set of all possible serial transaction schedules *for a given set of transactions Θ.*

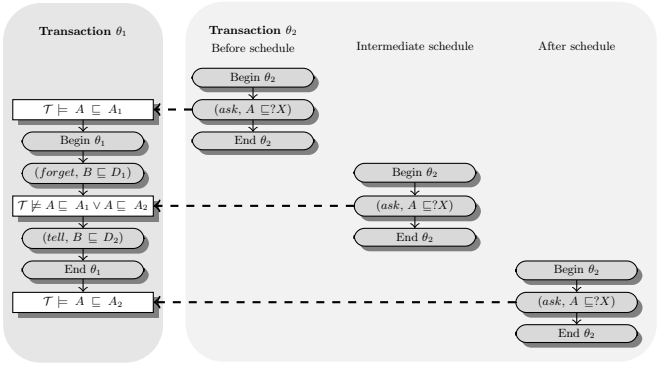


Fig. 1. Transaction Processing

Obviously, a *serial transaction schedule* π_{ser} is a transaction schedule that respects the original order of the atomic operations in the original transactions and executes operations of the individual transactions in distinguishable batches. For a set Θ of n transactions θ_i with $i = 1, \ldots, n$ there exist $n!$ different *serial transaction schedules*. Apart from the *serial transaction schedules*, a vast number of other interleaving schedules exists, e. g., for two transactions of lengths m_1, m_2 the possible number of schedules is $\binom{m_1+m_2}{m_1}$. So there is, in general, a large number of possibilities for transactions to interleave. In order to both preserve the intended semantics of a set of transactions and optimizing performance we are interested in *serializable* schedules.

Definition 5. *Let $\mathcal{O}$ be an ontology and $\Theta = \{\theta_1, \ldots, \theta_k\}$ be a set of transactions on $\mathcal{O}$. A transaction schedule $\pi' = (c_1, \ldots, c_n)$ of Θ is* serializable *if there exists a serial transaction schedule $\pi_{ser} = (d_1, \ldots, d_n)$ such that $c_i = d_{\sigma(i)}$ (for $i = 1, \ldots, n$) for some bijection $\sigma : \{1, \ldots, n\} \to \{1, \ldots, n\}$ of Θ such that*

1. $upd(\mathcal{O}, \pi') = upd(\mathcal{O}, \pi_{ser})$
2. $ans(\mathcal{O}, (c_1, \ldots, c_i)) = ans(\mathcal{O}, (d_{\sigma(1)}, \ldots, d_{\sigma(i)}))$ *for* $i = 1, \ldots, n$

In other words, a transaction schedule π' is serializable wrt. $\mathcal{O}$ if there is a *serial transaction schedule* π_{ser} such that applying π' on $\mathcal{O}$ yields the same ontology as applying π_{ser} on $\mathcal{O}$ and all answers to queries stay the same.

Example 3. We continue Example 2 with the two transactions $\theta_1 = (a_1, a_2)$ and $\theta_2 = (b_1)$. Possible transaction schedules, which adhere to a fixed order in operations of the same transaction, are $\pi_1 = (a_1, a_2, b_1)$, $\pi_2 = (a_1, b_1, a_2)$ and $\pi_3 = (b_1, a_1, a_2)$. Both π_1 and π_3 are serial transaction schedules. The schedule π_2 is not serializable due to

$$ans_{\pi_1}(\mathcal{O}, b_1) = \{A \sqsubseteq A_1\} \qquad ans_{\pi_3}(\mathcal{O}, b_1) = \{A \sqsubseteq A_2\}$$
$$ans_{\pi_2}(\mathcal{O}, b_1) \cap \{A_1, A_2\} = \emptyset$$

Definition 6. *A set of transaction $\Theta = \{\theta_1, \ldots, \theta_n\}$ is* conflicting *if there is a transaction schedule π that is not serializable.*

Obviously, in the case of conflicting transactions, some mechanism need to decide how transactions have to be scheduled in order to have a well-defined outcome of concurrent transactions. In the following, we address this issue in a conservative way by restricting interleaving executions of possibly conflicting transactions using locking.

5 What Has to Be Locked?

A problem of concurrent transaction management is to find, if existing, a *serializable transaction schedule* for a sequence of transactions $\theta_1, \ldots, \theta_n$. The simplest *serializable transaction schedule* for a given set of transactions Θ would be a *serial transaction schedule*, i. e., locking the whole ontology. However, such a schedule would potentially suffer from execution delays regarding multiple transactions. The other extreme is to lock exactly this part of the ontology necessary to avoid conflicts, but this could suffer from a potentially expensive calculation of the concrete locking area. To remedy this trade-off, we investigate the problem of determining the right part of the ontology that has to be locked. Based on this, we are able to investigate the problem of acquiring locks and determining a *serializable interleaving transaction schedule*.

5.1 Modules of an Ontology

According to our example in Sect. 4, a lock has to be acquired on more than just the axioms of the operations of a transaction ($\mathsf{axioms}(\theta)$). Additionally, also the logical consequences, constructed using symbols from $Sig(\theta)$, should be locked. Thus, for a transaction θ over ontology $\mathcal{O}$, we have to lock a sub-ontology $\mathcal{O}_\theta \subseteq \mathcal{O} \cup \mathsf{axioms}(\theta)$, so that every logical consequence α constructed using *only* symbols from $Sig(\theta)$ with $\mathcal{O} \cup \mathsf{axioms}(\theta) \models \alpha$ is already a logical consequence of $\mathcal{O}_\theta$. It is possible to define finite sets of axioms $\mathcal{M} \subseteq \mathcal{O}$ such that for all axioms α with terms only from some Signature $\mathbf{S} \subseteq Sig(\mathcal{O})$, we have that $\mathcal{M} \models \alpha$ iff $\mathcal{O} \models \alpha$. In such case $\mathcal{M}$ is called $\mathbf{S}$-module of $\mathcal{O}$, cf. [6].

Definition 7. *Let $\mathcal{O}' \subseteq \mathcal{O}$ be ontologies and $\mathbf{S}$ be a signature. Then $\mathcal{O}'$ is a module for $\mathbf{S}$ of $\mathcal{O}$, if for all axioms α with $Sig(\alpha) \subseteq \mathbf{S}$, it holds that $\mathcal{O}' \models \alpha$ if and only if $\mathcal{O} \models \alpha$.*

An important property of modules is *convexity*, i. e., given three ontologies $\mathcal{O}_1 \subseteq \mathcal{O}_2 \subseteq \mathcal{O}_3$ if $\mathcal{O}_1$ is an $\mathbf{S}$-module in $\mathcal{O}_3$ then $\mathcal{O}_1$ is an $\mathbf{S}$-module in $\mathcal{O}_2$ and $\mathcal{O}_2$ is an $\mathbf{S}$-module in $\mathcal{O}_3$ [6]. This means that it is sufficient to focus on minimal $\mathbf{S}$-modules. An $\mathbf{S}$-module $\mathcal{O}_1$ is minimal if there is no other $\mathbf{S}$-module $\mathcal{O}_2 \subsetneq \mathcal{O}_1$. This is also advantageous from a locking point of view, locking less is better since is is more likely that other transactions could also be executed. However, just one module is not enough since for a given signature $\mathbf{S}$ and an ontology $\mathcal{O}$ there might be multiple $\mathbf{S}$-modules and for our task we are interested in the fragment $\mathcal{O}_\theta$ that covers all axioms essential for the transaction θ. For such kind of fragment of an ontology the literature gives us the following definition, cf. [6].

Definition 8. *For a signature* $\mathbf{S}$ *and an ontology* $\mathcal{O}$*, we say that an axiom* $\alpha \in \mathcal{O}$ *is* $\mathbf{S}$*-essential in* $\mathcal{O}$ *wrt.* $\mathcal{L}$ *if* α *belongs to some minimal* $\mathbf{S}$*-module in* $\mathcal{O}$ *wrt.* $\mathcal{L}$*.*

Unfortunately, it has been shown in the literature that deciding if a set of axioms is a module is hard or even undecidable for expressive DLs [6,7]. But there exists several alternative (approximative) definitions of modules. One of them is the so called locality-based module (LBM) [16], which comes in two flavors, syntactic and semantic LBM. For syntactic LBMs it is known that they contain the corresponding semantic LBM and for their calculation algorithms with polynomial runtime wrt. the size of the ontology are known [16].

Based on the definition above, we can now state our notion of *influence area*, which describes the set of all axioms and entailments, which could be influenced by a single *atomic operation*.

Definition 9. *The minimal influence area* Ω_a *of an atomic operation* $a = (o, \alpha)$ *with respect to an ontology* $\mathcal{O}$ *is the set of all* $Sig(\alpha)$*-essential axioms in* $\mathcal{O}$*. If* $o =$ *tell we extend the definition to all* $Sig(\alpha)$*-essential axioms in* $\mathcal{O} \cup \{\alpha\}$*.*

5.2 Two-Phase Locking for Ontologies

Now, we are able to define a 2PL based locking mechanism for ontology transactions. Algorithm 1 displays the locking procedure. The input to the algorithm is the transaction θ_i and a global lock $GLock$, which is synchronized for all running instances of this procedure. The algorithm can be subdivided into three parts. First the initialization part, in which a local empty lock $TLock$ is initialized, line (2). The second part of the algorithm complies the 'Expanding Phase' of the *2PL* mechanism. The algorithm picks the current atomic operation (a) (4).

Algorithm 1: ExecuteTransaction

input : θ, a single transaction, $GLock$ a globale syncronized Lock

```
1  begin
       /* Initialization                                          */
2      TLock ⟵ ∅;
       /* Expanding phase: acquire locks                          */
3      for i = 1 to |θ| do
4          a ⟵ θ[i];
5          while ((GLock\TLock) ∩ Ω_Sig(a) ≠ ∅) do
6              wait;
7          GLock ⟵ GLock\TLock;
8          TLock ⟵ Ω_Sig(a_1...a_i);
9          GLock ⟵ GLock ∪ TLock;
10         execute a;
       /* Shrinking phase: remove all locks                       */
11     GLock ⟵ GLock\TLock;
```

Only if the intersection between this Ω_a and the global lock *GLock* is empty the algorithm will continue, otherwise it will wait (5,6). During the time procedure (a) is waiting for resources to be freed, it could happen that another parallel working procedure (b) changes the ontology in two ways that could affect (a). First, an axiom currently in the $TLock$ of (a) is removed by (b), then the $TLock$ of (a) is just too big but the locking is still valid. Second, a new axiom that should be part of $TLock$ (a) is added by (b), then the calculated $TLock$ of (a) is to small and therefore it has to be constantly recalculated. If it is empty the procedure acquires the lock for a, by adding Ω_a to $TLock$ as well as to *GLock*, lines (7,8,9). Then the procedure could execute the *atomic operation* a, line (10). As soon as all *atomic operations* of θ_i are processed, the procedure enters the 'Shrinking Phase' (third part) and frees all acquired locks (line (11)).

Theorem 1. *Let $\Theta = \{\theta_1, \ldots, \theta_n\}$ be a set of transactions. Any transaction schedule that is emitted by parallel executions of Algorithm 1 for each transaction $\theta_1, \ldots, \theta_n$ is serializable.*

Proof (Sketch). Let $\Theta = \{\theta_1, \ldots, \theta_n\}$ with $\theta_i = (a_{i,1}, \ldots, a_{i,m_i})$ for $i = 1, \ldots, k$ be a set of transactions. Let $\pi = (c_1, \ldots, c_n)$ be a transaction schedule emitted by the parallel executions of Algorithm 1. Consider the serial transaction schedule $\pi_{ser} = \theta_{\sigma(1)} \circ \ldots \theta_{\sigma(n)}$ with a permutation $\sigma : \{1, \ldots, n\} \to \{1, \ldots, n\}$ and $\sigma(i) < \sigma(j)$ iff $k < l$ for $c_k = a_{i,1}$ and $c_l = a_{j,1}$. In other words, π_{ser} is the serial schedule obtained from π by ordering the transactions according to their first operation in π. It suffices to show that π_{ser} is the witness of π's serializability according to Definition 5. Assume $upd(\mathcal{O}, \pi') = upd(\mathcal{O}, \pi_{ser})$ does not hold. Then there are transactions θ, θ' that manipulate some axiom $\alpha \in \mathcal{O}$. Without loss of generality assume θ appears before θ' in π_{ser}. Then θ acquires a lock on at least the axiom α—note that always $\alpha \in \Omega_{Sig(\alpha)}$—in line 9 of Algorithm 1 and releases it only after executing the whole transaction in line 11. Then θ' is blocked and $\mathcal{O}$ is updated in the same way as a serial execution of θ and θ', as in π_{ser}. It follows $upd(\mathcal{O}, \pi') = upd(\mathcal{O}, \pi_{ser})$. Similarly, it also holds that the answer behavior is the same for both π and π_{ser} by taking into account that the subset $\Omega'_{Sig(\alpha)} \subseteq \mathcal{O}$ that suffices to produce answers for an operation (ask, α)—i. e. $ans(\mathcal{O}, (ask, \alpha)) = \{\alpha' \in gr(\alpha) \mid \mathcal{O} \models \alpha'\} = \{\alpha' \in gr(\alpha) \mid \Omega_{Sig(\alpha)} \models \alpha'\}$— is accessed by only one transaction at a time as well. □

6 Evaluation

For our evaluation, we use different versions of the National Cancer Institute Thesaurus (NCIt) which are available as OWL EL++ ontologies[1]. As there are no real transaction logs available for NCIt (or any other versioned ontology), we perform our evaluation using transactions artificially generated from four consecutive versions available for NCIt. More specifically, for each two consecutive

[1] NCIt archive `http://evs.nci.nih.gov/ftp1/NCI_Thesaurus/archive`, Nov 2012.

versions of the NCIt ontology, we generate around 140 different transactions, each consisting of 6-12 atomic operations, which contain *tell*-operations on axioms that are present in the more recent version but missing in the previous version, *forget*-operations on axioms that are present in the previous version but missing in the more recent version, and *ask*-operations on axioms artificially generated partially from the signature of the *tell*- and *forget*-operations in the same transaction and potentially other symbols. We computed schedules for around 240 different combinations of these transaction.

Our evaluation aims at measuring the potential benefit of the module-based locking approach in terms of total execution time. For each atomic operation in a transaction, we compute the locking areas based on syntactic locality as described in Sec. 5. While the time needed for computing a module-based lock is, in general, much larger than for the whole ontology (which is almost immediate) we estimate a benefit when taking varying execution times of *non-critical* operations—i. e. user code that is contained in a transaction—into account. We expect that with increasing average execution time of non-critical operations the effort for computing a more specific locking area becomes negligible.

6.1 Evaluation Setup

In order to compensate for the lack of existing real transaction logs, we implemented Algorithm 1 in a non-parallel fashion and compute all serializable transaction schedules that are consistent with our locking approach. Let Θ^{mod} resp. Θ^{onto} be these sets of serializable transaction schedules. For the approach of locking the whole ontology for each atomic operation it follows that Θ^{onto} is the set of all serial transaction schedules. For both locking approaches and each transaction schedule $\theta = (c_1, \ldots, c_n)$ obtained in this way, we estimate the running time for executing the schedule as follows. Each atomic operation c_i $(i = 1, \ldots, n)$ can be decomposed via $c_i = c_i' c_i'' c_i'''$, where in c_i' the lock is acquired—which might take some time of lock calculation and the locking itself—c_i'' is the *critical operation*—which contains the actual database access and is the reason for acquiring the lock—and c_i''' is a *non-critical operation*, which might contain user interaction and other user code. For each non-critical operation c_i''', we consider different (but uniform over all non-critical operations) execution times while we assume critical operations to be immediate, i. e. they have an execution time of zero. If θ contains a sequence $c_i c_{i+1} = c_i' c_i'' c_i''' c_{i+1}' c_{i+1}'' c_{i+1}'''$ where c_i and c_{i+1} originate from different transactions we assume that c_i''' and $c_{i+1}' c_{i+1}'' c_{i+1}'''$ can be executed in parallel, thus decreasing total execution time. A *parallelization* f_θ of θ is a function $f_\theta : \{1, \ldots, n\} \to \{1, \ldots, n\}$ that satisfies

1. $f_\theta(i) \leq f_\theta(j)$ for all $i, j = 1, \ldots, n$,
2. If $f_\theta(i) = f_\theta(j)$ then c_i'' and c_j'' come from different transactions, and
3. there is $n' \leq n$ with $\mathsf{Im}\ f_\theta = \{1, \ldots, n'\}$ ($\mathsf{Im}\ f$ is the image of a function f)

Therefore, a parallelization f_θ says that all c_i'' with $f_\theta(i) = 0$ are executed in parallel at a first step (after their corresponding c_i'). Then all c_i'' with $f_\theta(i) = 1$

are executed in parallel, and so on. The first requirement above ensures that no c_i is executed before c_j if $j < i$. The second requirement says that only operations of different transactions can be executed in parallel, and the third requirement states that there are no steps in the execution where nothing is executed. Due to the assumed execution time of zero for critical operations, we can neglect those. Let F_θ be the set of all parallelizations of θ. As there may be different variants on how to parallelize a single transaction schedule we average the total execution time over all of them. Let t_{nc} be the average execution time for a non-critical operation and let $t_X(\theta)$ be the total time needed for computing locks in θ wrt. the approach $X \in \{\text{onto}, \text{mod}\}$. Then we estimate the total execution time for a transaction schedule $\theta = (c_1, \ldots, c_n)$ via

$$T^X_{t_{nc}}(\theta) = t_X(\theta) + t_{nc} \frac{\sum_{f_\theta \in F_\theta} \max \mathsf{lm}\ f_\theta}{|F_\theta|}$$

Finally, for each t_{nc} we take the average total execution time over all transaction schedules for both approaches, i. e.

$$T^X_{t_{nc}} = \frac{\sum_{\theta \in \Theta^X} T^X_{t_{nc}}(\theta)}{|\Theta^X|}$$

with $X \in \{\text{onto}, \text{mod}\}$. The implementation used for our evaluation can be downloaded from `https://launchpad.net/ontotrans`.

6.2 Results

As mentioned, we considered different combinations of transactions of different lengths. For around 30% of these tested combinations (≈ 240 combinations), we could find serializable interleaving schedules. This seams to be strongly related to our strategy of randomly picking axioms to generate the operations of a transaction. The influence area of a whole transaction consisting of randomly generated operations can be quite large so that the only possible serializable schedules for a combination of such transactions are the serial ones. For real transactions, we assume the axioms in the single operations to be more related to each other and therefore the influence areas to be smaller. Due to reasons of execution time, we decided to compute a maximum of 30 schedules per tested transaction combination. With these settings, we were able to find around 1200 serializable transaction schedules. The average serializable transaction schedules has only 76.642% of the length of the serial schedules and a single computation of the two modules, one for the global lock and one for the current atomic operation takes in average 2.832 seconds. Figure 2, displays the average total execution time for a schedule of average length of ten, considering different execution times for the non-critical part c_i''' of the atomic operation. The figure shows that starting from a average execution time for a non-critical operations of around 12 seconds the locking based approach starts to perform better.

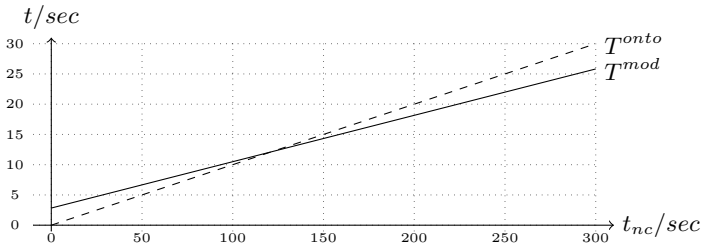


Fig. 2. Average Total Execution Time for T^{mod} vs. T^{onto}

6.3 Lessons Learned and Discussion

The relatively high threshold shown in Fig. 2 is the result of the expensive module calculation. Due to a lack of implementations of incremental module calculation mechanisms like those introduced in [17], we use the locality-based module calculation of the OWLAPI which recalculates the global module for every comparison. It turns out that this global lock calculation takes on average over 90% of the whole time spend on module calculation. Thus, applying an optimized incremental module calculation and efficient caching strategies would lead to a significant decrease in average module calculation time and therefore to a significantly lower threshold. However, even with our naive implementation our results depicted in Fig. 2 clearly show the benefit of computing module-based locks as total execution time decreases compared to the naive approach.

7 Conclusion

In this paper, we have presented a locking approach for concurrent ontology transactions. While the management of transactions in general is a challenging problem on its own, it becomes more complicated for ontologies since changes in an ontology also affect the entailments of the ontology. Thus, the management of transactions has to take the entailments of an ontology into account. Several research has been done in order to analyze changes in ontologies and to compare versions of ontologies or to build links between ontology versions. The locking approach in this paper is a further step towards collaborative ontology management. The locking principle takes the dependencies between axioms regarding the DL entailment into account, by determining the influence area of transactions. Locking policies lock ontologies according to the influence area of a transaction.

As a next step, we plan to investigate efficient scheduling of ontology transactions, while the presented locking principles and locking policies are the fundamental building blocks of a scheduling approach.

Acknowledgments. The research reported here was partially supported by the SocialSensor FP7 project (EC under contract number 287975).

References

1. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook. Cambridge University Press (2003)
2. Bernstein, P.A., Newcomer, E.: Principles of Transaction Processing. Morgan Kaufmann (1997)
3. Bernstein, P.A., Hadzilacos, V., Goodman, N.: Concurrency Control and Recovery in Database Systems. Addison-Wesley (1987)
4. Chaudhri, V.K., Hadzilacos, V., Mylopoulos, J.: Concurrency Control for Knowledge Bases. In: Nebel, B., Rich, C., Swartout, W.R. (eds.) KR, pp. 762–773. Morgan Kaufmann (1992)
5. Falconer, S.M., Tudorache, T., Noy, N.F.: An analysis of collaborative patterns in large-scale ontology development projects. In: K-CAP, pp. 25–32. ACM (2011)
6. Grau, B.C., Horrocks, I., Kazakov, Y., Sattler, U.: Modular Reuse of Ontologies: Theory and Practice. Journal of Artificial Intelligence Research 31, 273–318 (2008)
7. Grau, B.C., Horrocks, I., Kazakov, Y., Sattler, U.: Extracting Modules from Ontologies: A Logic-Based Approach. In: Stuckenschmidt, H., Parent, C., Spaccapietra, S. (eds.) Modular Ontologies. LNCS, vol. 5445, pp. 159–186. Springer, Heidelberg (2009)
8. Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P.F., Rudolph, S.: OWL 2 Web Ontology Language Primer. W3C Recommendation 27, 1–123 (2009)
9. Jiménez-Ruiz, E., Grau, B.C., Horrocks, I., Llavori, R.B.: Supporting concurrent ontology development: Framework, algorithms and tool. Data Knowl. Eng. 70(1), 146–164 (2011)
10. Karp, P.D., Chaudhri, V.K., Paley, S.M.: A Collaborative Environment for Authoring Large Knowledge Bases. J. Intell. Inf. Syst. 13(3), 155–194 (1999)
11. Noy, N.F., Musen, M.A.: Specifying Ontology Views by Traversal. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 713–725. Springer, Heidelberg (2004)
12. Qi, G., Yang, F.: A Survey of Revision Approaches in Description Logics. In: Calvanese, D., Lausen, G. (eds.) RR 2008. LNCS, vol. 5341, pp. 74–88. Springer, Heidelberg (2008)
13. Seidenberg, J., Rector, A.L.: A methodology for asynchronous multi-user editing of semantic web ontologies. In: Sleeman, D.H., Barker, K. (eds.) K-CAP, pp. 127–134. ACM (2007)
14. Seidenberg, J., Rector, A.L.: Web ontology segmentation: analysis, classification and use. In: Carr, L., De Roure, D., Iyengar, A., Goble, C.A., Dahlin, M. (eds.) WWW, pp. 13–22. ACM (2006)
15. Tudorache, T., Falconer, S., Noy, N.F., Nyulas, C., Üstün, T.B., Storey, M.-A., Musen, M.A.: Ontology development for the masses: Creating ICD-11 in webProtégé. In: Cimiano, P., Pinto, H.S. (eds.) EKAW 2010. LNCS, vol. 6317, pp. 74–89. Springer, Heidelberg (2010)
16. Del Vescovo, C., Klinov, P., Parsia, B., Sattler, U., Schneider, T., Tsarkov, D.: Syntactic vs. semantic locality: How good is a cheap approximation? CoRR, abs/1207.1641 (2012)
17. Del Vescovo, C., Parsia, B., Sattler, U., Schneider, T.: The modular structure of an ontology: Atomic decomposition and module count. In: WoMO, vol. 230, pp. 25–39. IOS Press (2011)

Predicting the Understandability of OWL Inferences

Tu Anh T. Nguyen, Richard Power, Paul Piwek, and Sandra Williams

Department of Computing, The Open University, Milton Keynes, UK
{t.nguyen,r.power,p.piwek,s.h.williams}@open.ac.uk

Abstract. In this paper, we describe a method for predicting the understandability level of inferences with OWL. Specifically, we present a probabilistic model for measuring the understandability of a multiple-step inference based on the measurement of the understandability of individual inference steps. We also present an evaluation study which confirms that our model works relatively well for two-step inferences with OWL. This model has been applied in our research on generating accessible explanations for an entailment of OWL ontologies, to determine the most understandable inference among alternatives, from which the final explanation is generated.

1 Introduction

The emergence of the semantic web community during the last decade has led to agreement on a common ontology language for exchanging knowledge called OWL (Web Ontology Language) [1]. Since being adopted as a standard language by the W3C in 2004, OWL has become widespread in many domains. Research on reasoning services for automatically computing logical inferences from OWL ontologies has also been intensively investigated since then, and resulted in automated reasoners such as FaCT++ [15], Pellet [14], and HermiT [10]. However, there has been little research investigating the *cognitive* difficulty of OWL inferences for humans, which is an essential problem in ontology debugging.

An important tool in debugging ontologies is to inspect entailments generated by an automated reasoner. An obviously incorrect entailed statement such as *SubClassOf(Person,Movie)* ("Every person is a movie") signals that something has gone wrong. However, many developers, especially those with limited knowledge of OWL, will need more information in order to make the necessary corrections: they need to understand *why* this entailment follows from the ontology, before they can start to repair it. Various *axiom pinpointing* tools have been proposed to compute *justifications* of an entailment—defined as any minimal subset of the ontology from which the entailment can be drawn—including both reasoner-dependent approaches [13,2] and reasoner-independent approaches [7,6]. A justification provides a set of premises for an entailment, so is helpful for diagnosing an erroneous entailment; however, unlike a proof, it does not explain how the premises combine with each other to produce the entailment. A user study [5] has

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 109–123, 2013.

shown that for many justifications (an example is shown in Table 1) even OWL experts were unable to work out how the conclusion follows from the premises without further explanation. For non-expert developers, the opacity of standard OWL syntaxes such as OWL/RDF, which are designed for efficient processing by computer programs and not for fast comprehension by people, can be another obstacle. As a possible solution to this problem, we are developing a system that explains, in English, why an entailment follows from an ontology.

Table 1. An example explanation generated by our prototype

Input	**Entailment:** *SubClassOf(Person,Movie)* **Justification:** 1. *EquivalentClasses(GoodMovie,ObjectAllValuesFrom(hasRating,FourStarRating))* 2. *ObjectPropertyDomain(hasRating,Movie)* 3. *SubClassOf(GoodMovie,StarRatedMovie)* 4. *SubClassOf(StarRatedMovie,Movie)*
Output	The statement "Every person is a movie" follows because: - everything is a movie (a). Statement (a) follows because: - anything that has as rating something is a movie (from axiom 2), and - everything that has no rating at all is a movie (b). Statement (b) follows because: - everything that has no rating at all is a good movie (c), and - every good movie is a movie (d). Statement (c) follows because axiom 1 in the justification means that "a good movie is anything that has as rating nothing at all, or has as rating only four-star ratings". Statement (d) follows because: - every good movie is a star rated movie (from axiom 3), and - every star rated movie is a movie (from axiom 4).

Table 1 shows an explanation generated by our prototype for the (obviously absurd) entailment "Every person is a movie" based on the proof tree in Figure 1. The key to understanding this proof lies in the step from axiom 1 to statement (c), which is an example of an inference in need of "further elucidation".

To generate such explanations, our system starts from a justification of the entailment, which can be computed using the method described by Kalyanpur et al. [7], and constructs *proof trees* in which the root node is the entailment, the terminal nodes are the axioms in the justification, and other nodes are intermediate statements (i.e., lemmas). Proof trees are constructed from a set of intuitively plausible *deduction rules* which account for a large collection of deduction patterns, with each local tree corresponding to a rule. For a given justification, the deduction rules might allow several proof trees, in which case we need a criterion for choosing the most understandable one.[1] From the

[1] Alternatively the deduction rules might not yield any proof trees, in which case the system has to fall back on simply verbalising the justification. Obviously such cases will become rarer as we expand the set of rules.

selected proof tree, the system generates an English explanation. Hard inference steps will be identified, and further elucidation will be added when necessary to make them understandable for most people. Such an explanation should be easier to understand than one based on the justification alone, as it replaces a single complex inference step with a number of simpler steps.

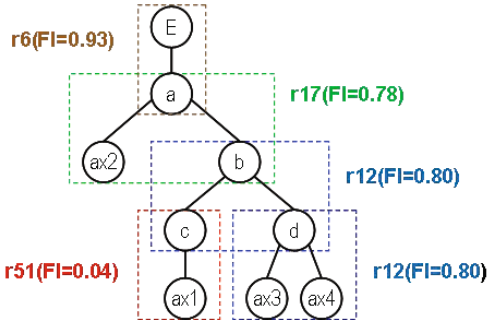


Fig. 1. The proof tree of the explanation in Table 1. The labels r6 etc. refer to rules listed in Table 4. FI values represent how easy it is to understand the rules—their *Facility Indexes*—with values ranging from 0.0 (hardest) to 1.0 (easiest).

As mentioned before, there may be multiple proof trees linking a justification to an entailment, and so multiple potential explanations of how the justification and the entailment are connected, some of which may be easier to follow than others. Therefore, being able to predict the understandability of a proof tree would be of great help in planning effective explanations for a given entailment. Specifically, it would enable the system to identify the most understandable explanation for a given justification. Additionally, when multiple justifications for an entailment are found, it would enable the system to sort explanations in order of decreasing understandability, which is very useful for end-users.

In prior work [12], we described how our current set of deduction rules was collected through analysis of a large corpus of approximately 500 OWL ontologies, and reported on an empirical study that allowed us to assign understandability indexes to the deduction rules. We called these indexes *facility indexes* (FIs). An FI of a deduction rule provides our best estimate of the probability that a person will understand the relevant inference step—i.e., that a person will recognise that the conclusion of the rule follows from the premises. Therefore, it ranges from 0.0 to 1.0, and the higher it is, the easier the inference. The result of this work is a list of 51 single-step inferences with known FIs, as shown in Table 4 (at the end of this paper) with the inferences sorted by FIs.

This paper focusses on the understandability of an entire proof tree. A proof tree can be viewed as a complex inference. When a tree has no lemma nodes, it corresponds to a single-step inference. Otherwise, it corresponds to a multiple-step inference, as in Figure 1. We propose here a model which predicts the understandability of a multiple-step inference based on the FIs of individual steps.

We also report on an evaluation study which confirms that our model works relatively well in detecting differences in understandability of two-step inferences. In this study we analysed both the participants' subjective reported understanding (how difficult they found the task), and their objective performance on the task (how often they got it right). The proposed model has been applied in our system to identify the best explanation for a given justification as well as to sort explanations by decreasing understandability when multiple justifications for a given entailment are found. We envisage that this model can be used by others to predict the understandability of different kinds of inferences.

2 Related Work

Several support tools have been proposed to help ontology developers to identify the causes of class unsatisfiability [8], and to rewrite potentially problematic axioms [9]. Two studies have been conducted [8,9] to evaluate how ontology developers debug ontologies with and without the tools. However, these studies focus on how people with a good understanding of OWL perform debugging, but not on how well they understand OWL inferences.

In a deduction rule, the conclusion can be viewed as an entailment, and the premises can be viewed as a justification of the entailment. Horridge et al. have proposed a model for measuring the cognitive difficulty of a justification [3]. In this model, they provide a list of *components*, each of which has an associated weight. For a given justification, the model checks for all appearances of these components, sums the weighted number of occurrences of the components, and outputs the result as the justification's difficulty score. The choice of the components and their weights is based on the authors' observations from an exploratory study [5] and their intuitions. Moreover, most of the proposed components are based on the syntactic analysis of justifications such as the number of premises in a justification, and these syntax-based components are mostly assigned a high weight. There are also several components for revealing difficult phenomena such as the trivial satisfaction of universal restriction[2] in OWL; however, the weights of these components are often low and are chosen intuitively. Therefore, this model predicts the difficulty of a justification in a manner that is biassed towards its structural complexity rather than its cognitive difficulty.

An empirical study was conducted by the model's authors to evaluate how well it predicts the difficulty of justifications. In this study, they created a deduction problem, presented in Manchester OWL Syntax [4] with alpha-numeric characters as class and property names, for testing a justification. In each problem, a justification and its entailment were given and subjects were asked whether the justification implied the entailment. A weakness of this study was that response bias was not controlled—i.e., if subjects had a positive response bias then they would have answered most questions correctly. Additionally, this study tested the model based on analysis of subjective understanding only.

[2] That is, if $\langle x, y\rangle \notin R^{\mathcal{I}}$ for all $y \in \Delta^{\mathcal{I}}$ then $x \in (\forall R.C)^{\mathcal{I}}$.

The above-mentioned complexity model and evaluation study were, in fact, inspired by those of Newstead et al. [11], which were proposed for measuring the difficulty of "Analytical Reasoning" (AR) problems in Graduate Record Examination (GRE) tests. An AR problem is a deductive reasoning problem in which an initial scenario is given along with a number of constraints called *rules*, and the examinee is asked to determine a possible solution for the problem among five choices. Like Horridge et al., Newstead et al. identified a set of difficulty factors and their weights through an intensive pilot study, and they built a preliminary difficulty model based on these factors and weights. After that, a series of large-scale studies was conducted to validate as well as adjust the model. Leaving aside the fact that these reasoning problems are different from OWL inferences, a strength of this work was that response bias of all types was successfully controlled. However, in both Newstead et al.'s and Horridge et al.'s work there was no clear explanation of how weights were assigned, suggesting that the choice might have been based partly on intuition.

3 An Understandability Model

This section describes our model for predicting the understandability of an OWL inference. Of course there is no fixed understandability for a given OWL inference as it depends on the readers' knowledge of OWL as well as their deductive reasoning ability. For this reason, it is impossible to provide an accurate measurement of the understandability of an inference that is correct for most people. However, what we expect from this model is the ability to detect the *difference* in the understandability between any two inferences. For example, if an inference is easier than another then we expect that our model will be able to detect it.

In prior work [12], we reported an empirical study for measuring the understandability of deduction rules that have been combined to construct proof trees for OWL justifications. A deduction rule is an inferential step from premises to a conclusion, which cannot be effectively simplified by introducing substeps (and hence, intermediate conclusions). Therefore, the understandability of a rule is, in fact, the understandability of the associated single-step OWL inference.

To measure the understandability of a deduction rule, we devised a deduction problem in which premises of the rule were given in English, replacing class or property variables by fictional nouns and verbs so that the reader would not be biassed by domain knowledge, and the subjects were asked whether the entailment of the rule followed from the premises.[3] The correct answer was always "Follows". To control for response bias (i.e., favouring a positive, or a negative, answer to *any* question), we included easy questions for both "Follows" and "Does not Follow" as *control questions* (as opposed to *test questions*). The complete discussion of the design of this study can be found in [12].

[3] Fictional words are nonsense words selected from various sources, such as Lewis Carroll's Jabberwocky poem (`http://en.wikipedia.org/wiki/Jabberwocky`), an automatic generator (`http://www.soybomb.com/tricks/words/`), and so on.

We used the proportion of correct answers for each test question as an index of understandability of the associated deduction rule, which we call its *facility index*. This index provides our best estimate of the probability that a person will understand the relevant inference step—i.e., that a person will recognise that the conclusion follows from the premises. Therefore, it ranges from 0.00 to 1.00, and the higher this value, obviously, the easier. Values of the FI for 51 rules tested in this study are shown in Table 4, ordered from high values to low. In this table, the rules r6, r12, and r17 used in the explanation in Table 1 are relatively easy, with FIs of 0.93, 0.80, and 0.78. By contrast rule r51, which infers statement (c) from axiom 1 in the example, is the hardest, with an FI of only 0.04.

To understand a more complex inference consisting of multiple inference steps, it is essential to be able to understand each individual inference step within it. Given a proof tree with FIs assigned to each inference step, such as the proof tree in Figure 1, a natural method of combining indexes would be to multiply them, so computing the joint probability of all steps being followed—in other words, the *facility index* of the proof tree. As before, the higher this value, the easier the proof tree. According to this model, the understandability of the proof tree in Figure 1 would be 0.93*0.78*0.80*0.04*0.80 or 0.02, indicating that the proof tree is very difficult to understand. This prediction is supported by the claim from the study conducted by Horridge and colleagues that this inference is very difficult even for OWL experts [5].

4 An Evaluation Study

In this section we report an experiment for evaluating our proposed model. We focussed on how well the model can detect differences in understandability between inferences. We adapted the use of *bins* for grouping inferences having close FIs from the study of Horridge et al. [3], but used a different experimental protocol and materials. Moreover, as mentioned in Section 1, both objective and subjective understanding of the subjects were analysed.[4]

4.1 Materials

We carried out the study with 15 proof trees collected from our ontology corpus. Each proof tree was assigned to an understandability bin on the basis of the FI predicted by our model. For our purpose, a total of five understandability bins were constructed over the range from 0.00 to 1.00, each with an interval of 0.20.[5] The test proof trees were selected so that there would be three for each bin, and additionally they would cover as many deduction rules as possible. In fact, our test proof trees included 25 of 51 rules from Table 4. For simplicity we only tested

[4] All the materials and results of this study can found at `http://mcs.open.ac.uk/nlg/SWAT/ESWC2013.html`.

[5] The ranges of the five bins were as follows: (B1) $0.80<x\leq1.00$, (B2) $0.60<x\leq0.80$, (B3) $0.40<x\leq0.60$, (B4) $0.20<x\leq0.40$, and (B5) $0\leq x\leq0.20$, respectively. B1 is the easiest bin and B5 is the hardest bin.

Table 2. The list of tested inferences and their predicted FIs

ID	Tested Inference	FI	ID	Tested Inference	FI
1.1	EqvCla(C0,C1) ∧ObjPropDom(r0,C0) →ObjPropDom(r0,C1) *(Rules used: r3, r1)*	**0.96**	2.1	ObjPropRng(r0,C1) ∧SymObjProp(r0) ∧SubClaOf(C1,C0) →ObjPropDom(r0,C0) *(Rules used: r18, r3)*	**0.74**
1.2	SubClaOf(ObjUniOf(C0,C1),C2) ∧SubClaOf(C0,C3) →SubClaOf(C0,ObjIntOf(C2,C3)) *(Rules used: r4, r5)*	**0.90**	2.2	SubClaOf(C0,C1) ∧SubClaOf(C1,C2) ∧ObjPropRng(r0,C0) →ObjPropRng(r0,C2) *(Rules used: r12, r8)*	**0.72**
1.3	SubClaOf(C0,ObjIntOf(C1,C2)) ∧ObjPropRng(r0,C0) →ObjPropRng(r0,C1) *(Rules used: r2, r8)*	**0.86**	2.3	EqvCla(C1,ObjUniOf(C2,C3)) ∧SubClaOf(C0,C2) →SubClaOf(C0,C1) *(Rules used: r10, r12)*	**0.66**
3.1	SubClaOf(ObjCompOf(C1),C2) ∧SubClaOf(C1,C0) ∧SubClaOf(C2,C0) →SubClaOf(⊤,C0) *(Rules used: r25, r24)*	**0.53**	4.1	ObjPropRng(r0,C1) ∧InvObjProp(r1,r0) ∧SubClaOf(C0,ObjSomValF(r1,C2)) →SubClaOf(C0,C1) *(Rules used: r44, r9)*	**0.34**
3.2	SubObjPpOf(r0,r1) ∧SubObjPpOf(r1,r2) ∧ObjPropDom(r2,C0) →ObjPropDom(r0,C0) *(Rules used: r14, r33)*	**0.48**	4.2	SubClaOf(C0,ObjSomValF(r0,C2)) ∧ObjPropRng(r0,C1) ∧DisCla(C1,C2) →SubClaOf(C0,⊥) *(Rules used: r30, r40)*	**0.32**
3.3	SubClaOf(C0,ObjMinCard(1,r1,C2)) ∧SubObjPpOf(r1,r0) ∧SubClaOf(ObjSomValF(r0,C2),C1) →SubClaOf(C0,C1) *(Rules used: r37, r11)*	**0.45**	4.3	SubClaOf(C2,ObjAllValF(r0,C1)) ∧InvObjProp(r0,r1) ∧SubClaOf(C0,ObjSomValF(r1,C2)) →SubClaOf(C0,C1) *(Rules used: r48, r12)*	**0.26**
5.1	FunDataProp(d0) ∧SubClaOf(C0,DataHasVal(d0,l0⋆DT0)) ∧SubClaOf(C0,DataHasVal(d0,l1⋆DT0)), l1≠l0 ∧SubClaOf(C1,ObjMinCard(2,r0,C0)) →SubClassOf(C1,⊥) *(Rules used: r45, r42)*	**0.18**			
5.2	SubClaOf(C1,ObjSomValF(r0,DataHasVal(d0,l0⋆DT0))) ∧DataPropRng(d0,DT1), D0 and DT1 are disjoint ∧SubClaOf(C0,ObjSomValF(r1,C1)) →SubClassOf(C0,⊥) *(Rules used: r49, r42)*	**0.09**			
5.3	EqvCla(C0,ObjAllValF(r0,C1)) ∧ObjPropDom(r0,C0) →SubClaOf(⊤,C0) *(Rules used: r51, r17)*	**0.03**			

proof trees consisting of exactly two deduction rules (i.e., two-step inferences). The list of tested inferences and their predicted FIs is shown in Table 2.

For each proof tree, we devised a *test problem* in which the proof tree was given to the subjects in the form of a simple explanation in English, and the subjects were asked whether the explanation is correct. We also asked the subject to rank how difficult they found the question on a scale from 5 (very easy) to 1 (very difficult). When presenting the test proof trees, we used fictional nouns and verbs so that the reader would not be biassed by domain knowledge, and labels such as (a), (b), and so on, to help subjects in locating the statements quicker. Since the correct answers to all test questions were "Yes", we controlled for response bias (i.e., favouring either positive or negative answers) by including a number of *control problems* as well as test problems. An example test problem in our study is shown in Figure 2.

Our control problems were designed to be similar to our test problems but were obvious to subjects who did the test seriously (rather than responding casually without reading the problem properly). We created two types of control problems: *non-entailment* and *trivial* problems. In a non-entailment problem the test proof tree includes a lemma or a conclusion about an object, a relationship, or both, that are not mentioned in the premises. The correct answer for

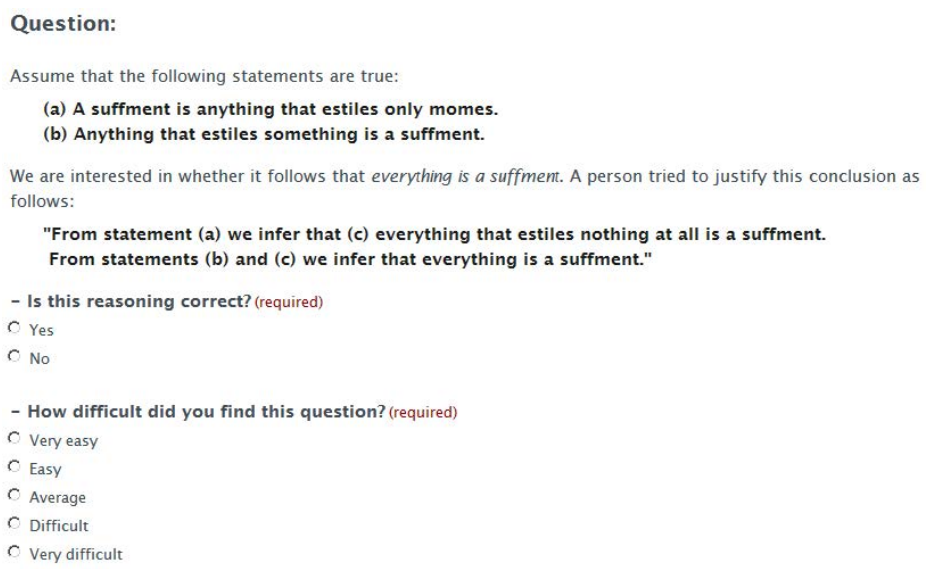


Fig. 2. A test problem in which the FI of the proof tree is 0.03 (0.04 * 0.78)

non-entailment problems is "No", trivially. In order to create such problems, we examined three possibilities for which the entailment is invalid:

1. First inference step is invalid, second inference step is valid
2. First inference step is valid, second inference step is invalid
3. Both inference steps are invalid

Among the three above-mentioned cases, one would expect fewer mistakes for the third case since they had two opportunities to detect a mistake in the reasoning. Therefore, in this study we used either the first or the second case. In both of these cases, we could not introduce unrelated objects into a premise as this violated the assumption of a test problem that all given premises were true; therefore, we only introduced new objects into the lemma in the first case or the entailment in the second case.

A trivial problem was one in which the test proof tree included only obviously correct inferences, so the correct answer was, also trivially, "Yes". Making trivial problems was quite tricky in this study as we could not merely use repetitions of premises, as we did in the previous study [12]. This is because people might get confused about whether a statement explained an entailment if it merely repeated the entailment. Since people usually reason better with individuals than with general statements, we used inferences with individuals in trivial problems.

As mentioned before, there were 15 test problems for which the correct answers were always positive. For balancing, we created 15 additional control problems,

five of which having positive answers and the remaining problems having negative answers. This resulted in 20 positive and 10 negative problems—i.e., 67% positive vs. 33% negative.

4.2 Method

The study was conducted on CrowdFlower, a crowdsourcing service that allows customers to upload tasks to be passed to labour channel partners such as Amazon Mechanical Turk[6]. We set up the operation so that tasks were channelled only to Amazon Mechanical Turk, and were restricted to subjects from Australia, the United Kingdom and the United States since we were aiming to recruit as many (self-reported) native speakers of English as possible.

To eliminate responses from 'scammers' (people who respond casually without considering the problem seriously), we used CrowdFlower's quality control service which is based on *gold-standard data*: we provided problems called *gold units* for which the correct answer is specified, allowing CrowdFlower to filter automatically any subjects whose performance on gold units falls below a threshold (75%). In our study, we selected five of our of fifteen control problems as gold units. The management of these gold units was internal to CrowdFlower, and the order for which these gold units would be presented varied randomly on subjects. As in our previous study, the control problems were used only in checking response biases and were not be counted in our main analysis.

It is important to note that in CrowdFlower subjects are not required to complete all problems. They can give up whenever they want, and their responses will be accepted so long as they perform well on gold units. CrowdFlower randomly assigns non-gold problems to subjects until it collects up to a specified number of valid responses for each problem. In our study we specified 80. However, since we were only interested in responses in which all 30 problems were answered, we selected only 59 valid responses.

5 Results

5.1 Control Problems

Figure 3 shows that for the 59 participants, there are 7 who answered fewer than 70% of the control questions correctly, suggesting that they were not performing the test seriously; their results were accordingly discarded. Of the 52 subjects remaining, only one claimed familiarity with OWL, 45 reported no familiarity, and the others did not specify (this question was optional).

5.2 Response Bias

Table 3 shows the absolute frequencies of the subjects' responses "Yes" (+Y) and "No" (−Y) for all problems in the study—both control and test. It also subdivides these frequencies according to whether the response was correct (+C) or

[6] `http://crowdflower.com/` and `http://www.mturk.com/`

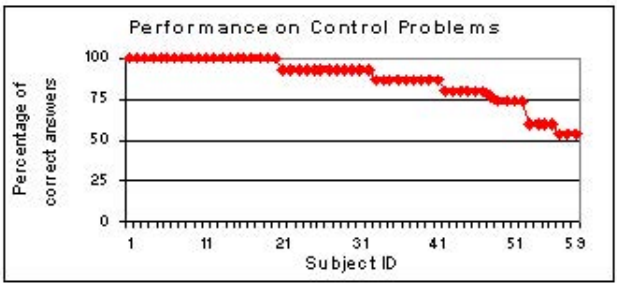


Fig. 3. The subjects' performance on the control problems sorted decreasingly

Table 3. The distribution of the subjects' responses—"Yes" (+Y) and "No" (−Y)—according to their correctness—"Correct" (+C) and "Incorrect" (−C)

	+Y	**−Y**	**TOTAL**
+C	774	458	1232
−C	59	265	324
TOTAL	833	723	1556

incorrect (−C). Thus for instance the cell +Y+C counts cases in which subjects answered "Yes" when this was the correct answer, while +Y−C counts cases in which they answered "Yes" when this was incorrect.

Recall that for 67% of the problems the correct answers were "Yes", and for all the remaining problems they were "No". If subjects had a positive response bias we would expect an overall rate much higher than 67%, but in fact we obtained 833/1556 or 54%, suggesting no positive response bias.

Looking at the distribution of incorrect answers, we can also ask whether subjects erred through being too ready to accept invalid conclusions (+Y−C), or too willing to reject conclusions that were in reality valid (−Y−C). The table shows a clear tendency towards the latter, with 265 responses in −Y−C compared with an expected value of 324*723/1556=151 calculated from the overall frequencies. In other words, subjects were more likely to err by rejecting a valid conclusion than by accepting an invalid one, a finding confirmed statistically by the extremely significant association between response (±Y) and correctness (±C) on a 2×2 chi-square test (χ^2=205.3, df=1, p<0.0001).

5.3 Analysis of Objective Understanding

Figure 4 shows the relationship between the predicted FIs and the proportions of correct answers for tested proof trees. Our analysis indicates a statistically significant relationship between the two values (r=0.88, p<0.0001) (Pearson's r correlation). For most tested proof trees the predicted FIs are lower than the

actual proportions of correct answers. A possible explanation is that all of the control questions in this study are two-step inferences whereas those in the previous study [12] are single-step inferences, and the use of more complex control questions in this study may have caused us to recruit better subjects than those of the previous study. However, for detecting differences in understandability of proof trees, our model works relatively well. Among the 15 tested trees in this study, there are 105 pairs on which difficulty comparisons can be made; of these, 93 comparisons were ordered in difficulty as predicted (i.e., an accuracy of 89%).

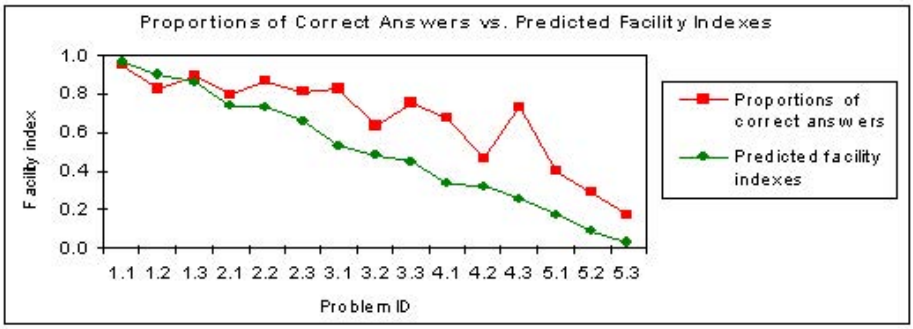


Fig. 4. The predicted FIs vs. the proportions of correct answers

We also tested how well our model can detect differences in understandability of proof trees by analysing the performance of the subjects by bins. For each of the 52 subjects, we counted the number of correct answers for the three questions in each bin, so obtaining a value of 0 to 3 for the associated bin. After that, we applied a Friedman test on the obtained values, which confirmed that there were statistically significant differences in performance between the five bins (χ^2=108.95, df=4, p<0.0001). Follow-up pairwise comparisons using a Wilcoxon Signed Ranks test showed that there were statistically significant differences in performance between any bin pair (p<0.05) except between bins 2 and 3. (This could be because subjects found questions 3.1 and 3.3 easier than expected, thus reducing the difference between bins 2 and 3.)

It is also clear from Figure 4 that there are exceptional cases for which the subjects performed much better than we expected, such as proof trees 4.3, 4.1, 3.3, and 3.2. The changes of verbalisations used in this study may be the main reason for these exceptions. Proof trees 4.1 and 4.3 are the only two cases which include an *InverseObjectProperties(r1,r0)* axiom. In the previous study [12], we used the verbalisation "X r0 Y if and only if Y r1 X" to present this axiom in rules 44 and 48 (in Table 4). The FIs we measured for these rules when using this verbalisation are 0.40 and 0.32 respectively. In this study, we used the verbalisation ""X r0 Y" means the same as "Y r1 X"", which is less technical than the former, for testing trees 4.1 and 4.3; this might explain why participants

performed better on these trees than we expected. The proportions of correct answers for trees 4.1 and 4.3 are 0.67 and 0.73.

Similarly, proof trees 3.2 and 3.3 are the only two cases which include *SubObjectPropertyOf(r1,r0)* axioms. In our previous study [12], we used the verbalisation "The property r1 is a sub-property of r0" to present this axiom in rules 33 and 37 (in Table 4). The FIs we measured for these rules when using this verbalisation are 0.61 and 0.55. In the present study, we used the less technical verbalisation "If X r1 Y then X r0 Y", which might again explain why performance on these trees was better than expected. The proportions of correct answers for trees 3.2 and 3.3 are 0.63 and 0.75.

5.4 Analysis of Subjective Understanding

Figure 5 plots the predicted FIs for test problems against the mean difficulty ratings (ranging from 1, very difficult, to 5, very easy) reported by subjects. The correlation between FIs and difficulty ratings is high (r=0.85) and significant (p<0.0001) (Pearson's r correlation).

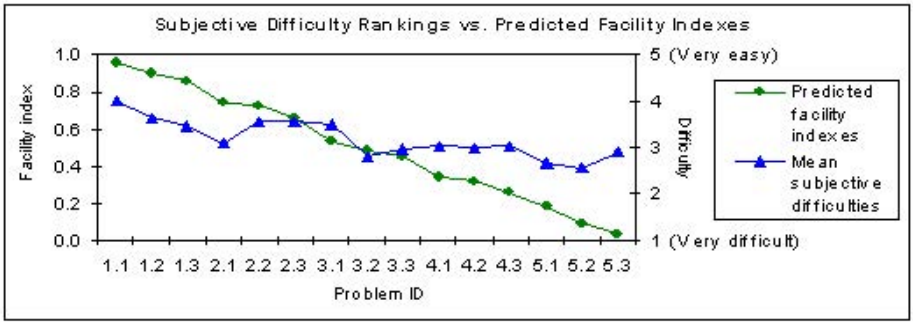


Fig. 5. The predicted FIs vs. the mean subjective difficulty ratings

As in the analysis of objective understanding, we tested how our model can detect differences in understandability of proof trees by analysing difficulty rankings by bins. For each of the 52 subjects, we computed the mean value of difficulty rankings for the three questions of each bin, and so obtained a value of 0 to 5 for the associated bin. After that, we applied a Friedman test on the obtained values, which confirmed that there were statistically significant differences in difficulty ranking between the five bins (χ^2=88.66, df=4, p<0.0001). Follow-up pairwise comparisons using a Wilcoxon Signed Ranks test showed that there were statistically significant differences in difficulty ranking between any bin pair (p<0.05) except between bins 3 and 4, for which the results might have been affected (as explained in section 5.3) by the more accessible verbalisations used in the present study for the proof trees 3.2, 3.3, 4.1, and 4.3. Proof tree 5.3 is an exception as it was ranked as easier than 5.2 while our model predicted

Table 4. Deduction rules and their facility indexes (FI). For short, the names of OWL functors are abbreviated.

ID	Rule	FI	ID	Rule	FI
1	EqvCla(X,Y...) →SubClaOf(X,Y)	**1.00**	2	SubClaOf(X,ObjIntOf(Y,Z...)) →SubClaOf(X,Y)	**0.96**
3	ObjPropDom(r0,X) ∧ SubClaOf(X,Y) →ObjPropDom(r0,Y)	**0.96**	4	SubClaOf(ObjUniOf(X,Y...),Z) →SubClaOf(X,Z)	**0.96**
5	SubClaOf(X,Y) ∧ SubClaOf(X,Z) →SubClaOf(X,ObjIntOf(Y,Z))	**0.94**	6	SubClaOf(⊤,Y) →SubClaOf(X,Y)	**0.93**
7	SubClaOf(X,ObjSomValF(r0,⊤)) ∧ SubClaOf(X,ObjAllValF(r0,Y)) →SubClaOf(X,ObjSomValF(r0,Y))	**0.90**	8	ObjPropRng(r0,X) ∧ SubClaOf(X,Y) →ObjPropRng(r0,Y)	**0.90**
9	ObjPropDom(r0,Y) ∧ SubClaOf(X,ObjSomValF(r0,Z)) →SubClaOf(X,Y)	**0.86**	10	EqvCla(X,ObjUniOf(Y,Z...)) →SubClaOf(Y,X)	**0.82**
11	SubClaOf(X,ObjSomValF(r0,Y)) ∧ SubClaOf(ObjMinCard(1,r0,Y),Z)) →SubClaOf(X,Z)	**0.82**	12	SubClaOf(X,Y) ∧ SubClaOf(Y,Z) →SubClaOf(X,Z)	**0.80**
13	SubClaOf(X,ObjCompOf(X)) →SubClaOf(X,⊥)	**0.80**	14	SubObjPpOf(r0,r1) ∧ SubObjPpOf(r1,r2) →SubObjPpOf(r0,r2)	**0.79**
15	SubClaOf(X,ObjSomValF(r0,Y)) ∧ SubClaOf(Y,Z) →SubClaOf(X,ObjSomValF(r0,Z))	**0.79**	16	EqvCla(X,ObjIntOf(Y,Z...)) →SubClaOf(X,Y)	**0.79**
17	ObjPropDom(r0,X) ∧ SubClaOf(ObjAllValF(r0,⊥),X) →SubClaOf(⊤,X)	**0.78**	18	ObjPropRng(r0,X) ∧ SymObjProp(r0) →ObjPropDom(r0,X)	**0.77**
19	SubClaOf(Y,X) ∧ SubClaOf(ObjCompOf(Y),X) →SubClaOf(⊤,X)	**0.77**	20	ObjPropDom(r0,⊥) →SubClaOf(⊤,ObjAllValF(r0,⊥))	**0.76**
21	ObjPropRng(r0,⊥) →SubClaOf(⊤,ObjAllValF(r0,⊥))	**0.76**	22	DisCla(X,Y...) ∧ SubClaOf(Z,X) ∧ SubClaOf(W,Y) →DisCla(Z,W)	**0.76**
23	SubClaOf(X,ObjSomValF(r0,Y)) ∧ SubClaOf(Y,ObjSomValF(r0,Z)) ∧ TrnObjProp(r0) →SubClaOf(X,ObjSomValF(r0,Z))	**0.75**	24	SubClaOf(X,ObjUniOf(Y,Z)) ∧ SubClaOf(Y,W) ∧ SubClaOf(Z,W) →SubClaOf(X,W)	**0.73**
25	SubClaOf(ObjCompOf(X),Y) →SubClaOf(⊤,ObjUniOf(X,Y))	**0.72**	26	SubClaOf(X,ObjUniOf(Y,Z)) ∧ SubClaOf(Y,Z) →SubClaOf(X,Z)	**0.71**
27	SubClaOf(ObjSomValF(r0,X),Y) ∧ SubClaOf(ObjAllValF(r0,⊥),Y) →SubClaOf(ObjAllValF(r0,X),Y)	**0.71**	28	ObjPropDom(r0,X) ∧ SymObjProp(r0) →ObjPropRng(r0,X)	**0.69**
29	SubClaOf(X,ObjSomValF(r0,ObjSomValF(r0,Y))) ∧ TrnObjProp(r0) →SubClaOf(X,ObjSomValF(r0,Y))	**0.68**	30	ObjPropRng(r0,Z) ∧ SubClaOf(X,ObjSomValF(r0,Y)) →SubClaOf(X,ObjSomValF(r0,ObjIntOf(Y,Z)))	**0.64**
31	SubClaOf(⊤,Y) ∧ DisCla(X,Y) →SubClaOf(X,⊥)	**0.64**	32	SubClaOf(X,ObjExtCard(n1,r0,Y)) →SubClaOf(X,ObjMinCard(n2,r0,Y)), 0<n2≤n1	**0.63**
33	ObjPropDom(r0,X) ∧ SubObjPpOf(r1,r0) →ObjPropDom(r1,X)	**0.61**	34	SubClaOf(X,Y) ∧ DisCla(X,Y) →SubClaOf(X,⊥)	**0.57**
35	SubClaOf(X,Y) ∧ SubClaOf(X,Z) ∧ DisCla(Y,Z) →SubClaOf(X,⊥)	**0.56**	36	TrnObjProp(r0) ∧ InvObjProp(r0,r1) →TrnObjProp(r1)	**0.55**
37	SubClaOf(X,ObjSomValF(r0,Y)) ∧ SubObjPropOf(r0,r1) →SubClaOf(X,ObjSomValF(r1,Y))	**0.55**	38	ObjPropRng(r0,X) ∧ SubObjPropOf(r1,r0) →ObjPropRng(r1,X)	**0.52**
39	SubClaOf(X,Y) ∧ SubClaOf(X,ObjCompOf(Y)) →SubClaOf(X,⊥)	**0.51**	40	SubClaOf(X,ObjSomValF(r0,ObjIntOf(Y,Z...))) ∧ DisCla(Y,Z) →SubClaOf(X,⊥)	**0.50**
41	SubClaOf(X,ObjMinCard(n1,r0,Dor⊤)) ∧ SubClaOf(X,ObjMaxCard(n2,r0,⊤)), 0<n2<n1 →SubClaOf(X,⊥)	**0.48**	42	SubClaOf(X,ObjSomValF(r0,Y)) ∧ SubClaOf(Y,⊥) →SubClaOf(X,⊥)	**0.45**
43	FuncDatProp(d0) ∧ SubClaOf(X,DatMinCard(n,d0,DR0)), n>1 →SubClaOf(X,⊥)	**0.41**	44	ObjPropRng(r0,X) ∧ InvObjProp(r0,r1) →ObjPropDom(r1,X)	**0.40**
45	FuncDatProp(d0) ∧ SubClaOf(X,DatHasVal(d0,l0⋆DT0)) ∧ SubClaOf(X,DatHasVal(d0,l1⋆DT1)) where DT0 and DT1 are disjoint or l0 ≠ l1 →SubClaOf(X,⊥)	**0.40**	46	FuncObjProp(r0) ∧ SubClaOf(X,ObjHasVal(r0,i0)) ∧ SubClaOf(X,ObjHasVal(r0,i1)) ∧ DiffInd(i0,i1...) →SubClaOf(X,⊥)	**0.39**
47	ObjPropDom(r0,X) ∧ InvObjProp(r0,r1) →ObjPropRng(r1,X)	**0.38**	48	SubClaOf(X,ObjAllValF(r0,Y) ∧ InvObjProp(r0,r1) →SubClaOf(ObjSomValF(r1,X),Y)	**0.32**
49	DatPropRng(d0,DR0) ∧ X⊑ObjSomValF(r0,DatHasVal(d0,l0⋆DT1)) where DR0 & DT1 are disjoint →SubClaOf(X,⊥)	**0.19**	50	DatPropRng(d0,DR0) ∧ SubClaOf(X,DatSomeValFrm(d0,DR1)) where DR0 & DR1 are disjoint →SubClaOf(X,⊥)	**0.18**
51	EqvCla(X,ObjAllValF(r0,Y)) →SubClaOf(ObjAllValF(r0,⊥),X)	**0.04**			

the opposite direction. Our prediction was supported by the analysis of objective understanding presented previously. This result suggests a failure in understanding this proof tree—that is, the subjects thought that they had understood the inference correctly but actually they had not.

6 Conclusions and Future Work

This paper describes a method for predicting the understandability of OWL inferences, focussing on people with limited knowledge of OWL. We present a probabilistic model for measuring the understandability of a multiple-step inference based on measurement of the understandability of single-step inferences. First the FIs of 51 single-step inferences were measured in an empirical study resulting in estimates of the probability that a person will understand the inference. Then by multiplying the FIs of individual inference steps, we can compute the joint probability of all steps being followed as the FI of the associated multiple-step inference. We also report an evaluation study which confirms that our model works relatively well for two-step inferences in OWL. This model has been applied in our research on generating accessible explanations for entailments derived from OWL ontologies, to determine the most understandable among alternative inferences from a justification, as well as to sort explanations in order of decreasing understandability when multiple justifications are found.[7]

The proposed model grounds FIs in a well-established probabilistic interpretation. This gives us confidence that the good performance of the model on two-step inferences will extend to n-step inferences for n>2. This has, however, to be balanced with the somewhat better performance of the theoretically less well-founded approach of taking the minimum, which for two-step inferences achieves an accuracy of 94%. Further work is needed to compare these models for inferences with more than two steps.

In addition to improving the understandability model, we will aim to make our explanations for absurd entailments more focused; for instance, by tracing from the entailment in Table 1 to a sequence of absurd lemmas, including "Everything is a movie", "Everything that has no rating at all is a movie", and "Everything that has no rating at all is a good movie", and finally reaching the misused axiom "A good movie is anything that has as ratings only four stars". Leaving aside the way the proposed model was used in our work, we believe it can be used by others to predict the understandability of different kinds of inferences, and so is worth reporting as a resource for other researchers.

Acknowledgments. This research was undertaken as part of the SWAT (Semantic Web Authoring Tool) project, supported by the UK Engineering and Physical Sciences Research Council (EPSRC grant no. G033579/1). We thank our colleagues and the anonymous viewers.

[7] We have implemented a prototype of this model as a plug-in of the SWAT ontology editing tool, which will be published soon at `http://mcs.open.ac.uk/nlg/SWAT/`.

References

1. OWL 2 Web Ontology Language Document Overview, 2nd edn., http://www.w3.org/TR/owl2-overview/ (last accessed: February 1, 2013)
2. Baader, F., Peñaloza, R., Suntisrivaraporn, B.: Pinpointing in the Description Logic $\mathcal{EL}^+$. In: Hertzberg, J., Beetz, M., Englert, R. (eds.) KI 2007. LNCS (LNAI), vol. 4667, pp. 52–67. Springer, Heidelberg (2007)
3. Horridge, M., Bail, S., Parsia, B., Sattler, U.: The Cognitive Complexity of OWL Justifications. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 241–256. Springer, Heidelberg (2011)
4. Horridge, M., Drummond, N., Goodwin, J., Rector, A., Stevens, R., Wang, H.: The Manchester OWL Syntax. In: International Workshop on OWL: Experiences and Directions (OWLED 2006) (2006)
5. Horridge, M., Parsia, B., Sattler, U.: Lemmas for Justifications in OWL. In: International Workshop on Description Logics (DL 2009) (2009)
6. Ji, Q., Qi, G., Haase, P.: A Relevance-Directed Algorithm for Finding Justifications of DL Entailments. In: Gómez-Pérez, A., Yu, Y., Ding, Y. (eds.) ASWC 2009. LNCS, vol. 5926, pp. 306–320. Springer, Heidelberg (2009)
7. Kalyanpur, A., Parsia, B., Horridge, M., Sirin, E.: Finding All Justifications of OWL DL Entailments. In: Aberer, K., et al. (eds.) ISWC/ASWC 2007. LNCS, vol. 4825, pp. 267–280. Springer, Heidelberg (2007)
8. Kalyanpur, A., Parsia, B., Sirin, E., Hendler, J.: Debugging Unsatisfiable Classes in OWL Ontologies. Journal of Web Semantics 3(4), 268–293 (2005)
9. Lam, J.S.C., Sleeman, D., Pan, J.Z., Vasconcelos, W.W.: A Fine-Grained Approach to Resolving Unsatisfiable Ontologies. In: Spaccapietra, S. (ed.) Journal on Data Semantics X. LNCS, vol. 4900, pp. 62–95. Springer, Heidelberg (2008)
10. Motik, B., Shearer, R., Horrocks, I.: A Hypertableau Calculus for $\mathcal{SHIQ}$. In: International Workshop on Description Logics (DL 2007), pp. 419–426 (2007)
11. Newstead, S.E., Bradon, P., Handley, S.J., Dennis, I., Evans, J.S.B.T.: Predicting the difficulty of complex logical reasoning problems. Thinking & Reasoning 12(1), 62–90 (2006)
12. Nguyen, T.A.T., Power, R., Piwek, P., Williams, S.: Measuring the Understandability of Deduction Rules for OWL. In: International Workshop on Debugging Ontologies and Ontology Mappings (WoDOOM 2012) (2012)
13. Schlobach, S., Cornet, R.: Non-standard Reasoning Services for the Debugging of Description Logic Terminologies. In: International Joint Conference on Artificial Intelligence (IJCAI 2003), pp. 355–360 (2003)
14. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A Practical OWL-DL Reasoner. Journal of Web Semantics 5, 51–53 (2007)
15. Tsarkov, D., Horrocks, I.: FaCT++ Description Logic Reasoner: System Description. In: Furbach, U., Shankar, N. (eds.) IJCAR 2006. LNCS (LNAI), vol. 4130, pp. 292–297. Springer, Heidelberg (2006)

Detecting SPARQL Query Templates for Data Prefetching

Johannes Lorey and Felix Naumann

Hasso Plattner Institute,
Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam, Germany
{johannes.lorey,felix.naumann}@hpi.uni-potsdam.de

Abstract. Publicly available Linked Data repositories provide a multitude of information. By utilizing SPARQL, Web sites and services can consume this data and present it in a user-friendly form, e.g., in mash-ups. To gather RDF triples for this task, machine agents typically issue similarly structured queries with recurring patterns against the SPARQL endpoint. These queries usually differ only in a small number of individual triple pattern parts, such as resource labels or literals in objects. We present an approach to detect such recurring patterns in queries and introduce the notion of query templates, which represent clusters of similar queries exhibiting these recurrences. We describe a matching algorithm to extract query templates and illustrate the benefits of prefetching data by utilizing these templates. Finally, we comment on the applicability of our approach using results from real-world SPARQL query logs.

1 Introduction

Public SPARQL endpoints provide valuable resources for various information needs, e.g., drug information[1] or government spending data[2]. While end users are in most cases free to query these endpoints using Web forms, a much more widespread way to consume the provided data is through an intermediary software or service [14], including mash-ups[3,4] or general-purpose exploration[5] tools.

Whereas such frontends may increase usability, they typically reduce the scope of issued queries. Depending on the architecture and purpose of the software, requests exhibit certain recurring patterns [14], e.g., based on interaction with a fixed user interface. Potentially, these patterns result from combining Linked Data with unstructured or semi-structured information. For example, literals, such as labels or latitude and longitude specifications, may be extracted from user input and serve as objects of an individual triple pattern within a query, whereas the overall structure of this query is hard-coded. Hence, the application

[1] http://www4.wiwiss.fu-berlin.de/drugbank/sparql
[2] http://govwild.hpi-web.de/sparql
[3] http://km.aifb.kit.edu/sites/spark/
[4] http://code.google.com/p/sgvizler/
[5] http://iwb.fluidops.com/

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 124–139, 2013.

or Web site issues many highly-similar queries on behalf of its users and utilizes only a subset of the information provided by the SPARQL endpoint. However, those (nearly) identical requests generated by user input increase the load on the SPARQL endpoint as well as the response time of the application's frontend. Additionally, in case the SPARQL endpoint becomes unavailable, the entire application has no access to the data.

One solution to this problem is to employ result caching. Caching eliminates the need to issue identical requests to the SPARQL endpoint multiple times assuming the knowledge base does not change over time. However, this solution works only if the exact same query is discovered in subsequent requests. In a real-world scenario however, it is more likely to encounter similar queries retrieving information about related resources. For these new queries, none or only partial locally cached information of previous requests can be used.

However, it might prove beneficial to gather the data relevant for related resources if a recurring access pattern is discovered. There exist different approaches of how to detect such related resources, including considering ontology information or graph distance metrics. In this work, we do not assume knowledge of such metadata and instead focus on structural elements of the SPARQL queries to determine the relatedness of RDF resources. We present an approach to detect recurring query patterns and use these patterns to introduce the notion of query templates. Query templates can be considered representatives of potentially overlapping query clusters containing structurally similar SPARQL queries. Furthermore, we introduce a use case for these query templates where the idea is to reduce the number of queries issued against a SPARQL endpoint by prefetching data relevant for subsequent requests.

This paper is organized as follows: We present related research in the fields of SPARQL query profiling and semantic caching in Sec. 2. In Sec. 3, we introduce fundamental notions required for this work. Section 4 provides details of our approach for discovering triple pattern mappings and graph pattern mappings as well as an algorithm for detecting and extracting query templates. We present some results for determining query templates in query sessions and evaluate our query rewriting approach on real-world SPARQL query logs in Sec. 5. Lastly, we conclude this work and comment on future research activities in Sec. 6.

2 Related Work

The related work for this paper draws mainly from two fields (i) *SPARQL Query Profiling*, e.g., the systematic analysis of queries to detect usage patterns, and (ii) *Semantic Caching and Prefetching*, e.g., techniques to either retain previously fetched data or retrieve data relevant for subsequent queries.

2.1 SPARQL Query Profiling

There have been a number of scientific projects aiming for a better understanding of structures and patterns of Linked Data. Here, most of the work has focused on

profiling the data itself, such as [1,4,8]. However, analyzing and profiling actual queries on Linked Data has recently also spawned a number of applications, such as SPARQL benchmarking [3,12] or providing query suggestions [9,16].

Our work is closely related to the latter topic. As the results in [14] suggest, there is great potential for discovering and reusing patterns of SPARQL queries. Indeed, in [9] the authors present a supervised machine learning framework to suggest SPARQL queries based on examples previously selected by the user. The authors claim that their approach benefits users who have no knowledge of the underlying schema or the SPARQL query language. A similar approach in [16] allows users to refine an initial query based on keywords.

In contrast to these works, the goal of our research is an automated approach to prefetch information without a priori knowledge of the knowledge base. Moreover, we rely on the structure of queries instead of applying natural language processing techniques on potentially unrelated keywords or resources. Additionally, we allow analysis of complex SPARQL queries and offer a means to cluster such queries for subsequent analysis. Overall, our research extends previous works on profiling Linked Open Data usage [11,14] by suggesting a concrete use case for recurring patterns in SPARQL queries.

2.2 Semantic Caching and Prefetching

Semantic caching builds on the idea of maintaining a local copy of retrieved data that can be used for subsequent queries. As with traditional caching, one of the motivations for semantic caching is to reduce the transmission overhead when retrieving data over a network link. Conventional approaches, such as tuple or page caching, usually retain fetched data based on temporal or frequency aspects, e.g., by prioritizing least-recently or least-frequently used items. Such techniques also exist for SPARQL query result caching [10,15]. Compared to this, semantic caching employs more fine-grained information to characterize data, e.g., in order to establish variable-sized semantic regions containing related tuples [5].

Closely related to semantic caching and our work is prefetching. Instead of simply retaining tuples retrieved previously, prefetching allows to gather data that is potentially useful for subsequent queries based on semantic information derived from past queries or the overall system state. In computer architecture design, prefetching is usually employed to request instructions that are anticipated to be executed in the future and place them in the CPU cache. For information retrieval, query prefetching typically assumes a probabilistic model, e.g., considering temporal features [6]. However, to the best of our knowledge, there have been no attempts to prefetch RDF data based on the structure of sequential related SPARQL queries within and across query sessions.

3 SPARQL Preliminaries

SPARQL is the de facto standard query language for RDF triples. In this section, we introduce some basic notions of SPARQL. Based on this, we illustrate several concepts used in this work to identify individual elements of a query. We use

these concepts in Sec. 4 to describe a matching algorithm for SPARQL queries based on an underlying query normal form.

3.1 SPARQL Graph Patterns

One central concept of a SPARQL query is that of a triple pattern $T = (s, p, o) \in (V \cup U) \times (V \cup U) \times (V \cup U \cup L)$ with V being a set of variables, U being a set of URIs, and L being a set of literals. A SPARQL query Q contains a number of graph patterns $P_1, P_2, \ldots$, which are defined recursively: (i) A valid triple pattern T is a graph pattern. (ii) If P_1 and P_2 are graph patterns, then P_1 `AND` P_2, P_1 `UNION` P_2, and P_1 `OPTIONAL` P_2 are graph patterns [13]. While there is the notion of empty graph patterns in SPARQL, we consider only non-empty graph patterns. Additionally, we focus on SELECT queries. An example of such a query is illustrated in Listing 1.

```
SELECT * WHERE {
  {
    ?p1 foaf:firstName "Alice" .
    ?p1 ?associationWith example:Bob .
  } UNION {
    ?p2 foaf:firstName "Carol" .
    OPTIONAL {
      ?p2 ?associationWith ?p1 .
    }
  }
}
```

Listing 1. SPARQL query example

In terms of relational operations, the keyword `AND` represents an inner join of the two graph patterns, `UNION` unsurprisingly denotes their union, and `OPTIONAL` indicates a left outer join between P_1 and P_2. Whereas `UNION` and `OPTIONAL` are reserved keywords in actual SPARQL queries to indicate the corresponding connection between two graph patterns, the `AND` keyword is omitted. In [13], it is shown that there exists a notion of a normal form for SPARQL queries based on the recursive graph pattern structure presented earlier and the precedence of the operators connecting those graph patterns. Hence, for this work we assume a SPARQL SELECT query can always be expressed as a composition of graph patterns, connected either by `UNION`, `AND`, or `OPTIONAL`.

Curly braces delimiting a graph pattern (i.e., $\{P\}$) are syntactically required for both P_1 and P_2 in a `UNION` statement and only for P_2 in an `OPTIONAL` statement. Furthermore, we refer to the largest delimited graph pattern P contained in a SPARQL query Q as the *query pattern* P_Q. Note that every query has exactly one query pattern P_Q. To increase readability and avoid confusion with set braces, we omit the brace delimiters in this work whenever possible. For the remainder of this work, P_i denotes a valid graph pattern contained in P_Q.

In Sec. 4, we introduce a matching algorithm for graph patterns. One necessary prerequisite for this algorithm is to identify individual child graph patterns

contained in P_Q. For example, the query in Listing 1 contains the following three non-trivial child graph patterns $P_{\texttt{AND}}$, $P_{\texttt{OPTIONAL}}$, and $P_{\texttt{UNION}}$:

$P_{\texttt{AND}}$:=
```
?p1 foaf:firstName "Alice" .
?p1 ?associationWith example:Bob .
```

$P_{\texttt{OPTIONAL}}$:=
```
?p2 foaf:firstName "Carol" .
OPTIONAL {
  ?p2 ?associationWith ?p1 .
}
```

$P_{\texttt{UNION}} = P_Q := P_{\texttt{AND}}$ `UNION` $P_{\texttt{OPTIONAL}}$

3.2 Graph Pattern Decomposition

To extract child graph patterns, we introduce the three functions $\Theta_{\texttt{UNION}}(P)$, $\Theta_{\texttt{AND}}(P)$, and $\Theta_{\texttt{OPTIONAL}}(P)$. They each take as input a graph pattern P and totally decompose P into the set of its non-empty child graph patterns $P_1, P_2, \ldots, P_n$, all conjoined exclusively by `UNION`, `AND`, or `OPTIONAL`, respectively. The three functions can then be applied recursively on the individual elements $P_1, P_2, \ldots, P_n$ in the result set, possibly yielding further non-trivial results.

For example, if we apply $\Theta_{\texttt{UNION}}(P_Q)$ on the query pattern in Listing 1, we retrieve the set $\{P_{\texttt{AND}}, P_{\texttt{OPTIONAL}}\}$. Similarly, $\Theta_{\texttt{AND}}(P_{\texttt{AND}})$ retrieves a set containing the two triple patterns listed above as elements. If no such total decomposition can be derived, the result set is empty, e.g., $\Theta_{\texttt{AND}}(P_Q) = \emptyset$ or $\Theta_{\texttt{UNION}}(P_{\texttt{AND}}) = \emptyset$.

Whereas in general, for $\oplus \in \{\texttt{UNION}, \texttt{AND}, \texttt{OPTIONAL}\}$:

$$\Theta_{\oplus}(P) \neq \emptyset \Leftrightarrow P := P_1 \oplus P_2 \oplus \ldots \oplus P_n,$$

the individual functions are defined as follows (all $n \geq 2$):

$$\Theta_{\texttt{UNION}}(P) := \begin{cases} \{P_1, \ldots, P_n\}, & \text{iff } P := P_1 \texttt{ UNION } P_2 \texttt{ ... UNION } P_n \\ \emptyset, & \text{else.} \end{cases}$$

$$\Theta_{\texttt{AND}}(P) := \begin{cases} \{P\}, & \text{iff } P \text{ is a triple pattern} \\ \{P_1, \ldots, P_n\}, & \text{iff } P := P_1 \texttt{ AND } P_2 \texttt{ ... AND } P_n \\ \emptyset, & \text{else.} \end{cases}$$

$$\Theta_{\texttt{OPTIONAL}}(P) := \begin{cases} \{P_1, \ldots, P_n\}, & \text{iff } P := P_1 \texttt{ OPTIONAL } P_2 \texttt{ ... OPTIONAL } P_n \\ \emptyset, & \text{else.} \end{cases}$$

We also define the function $\Theta(P)$ as a convenience method to detect whether for a graph pattern P a decomposition exists for either $\Theta_{\texttt{UNION}}(P)$, $\Theta_{\texttt{OPTIONAL}}(P)$, or $\Theta_{\texttt{AND}}(P)$ (in this order):

$$\Theta(P) := \begin{cases} \Theta_{\texttt{UNION}}(P), & \text{iff } \Theta_{\texttt{UNION}}(P) \neq \emptyset \\ \Theta_{\texttt{OPTIONAL}}(P), & \text{iff } \Theta_{\texttt{OPTIONAL}}(P) \neq \emptyset \\ \Theta_{\texttt{AND}}(P), & \text{else.} \end{cases}$$

Except for when P is a triple pattern and we apply $\Theta_{\texttt{AND}}(P) = P$, we also assume that all decompositions are non-trivial, i.e., $\Theta_{\oplus}(P) \neq \{P\}$. Hence, according to the underlying graph pattern normal form, all the above cases are mutually exclusive. We call $|P| = |\Theta(P)|$ the *size* of a graph pattern.

In addition, we introduce the function $\kappa(P)$ for a graph pattern P:

$$\kappa(P) := \begin{cases} \texttt{UNION}, & \text{iff } \exists P_1 \in P_Q : P \in \Theta_{\texttt{UNION}}(P_1) \\ \texttt{OPTIONAL}, & \text{iff } \exists P_1, P_2 \in P_Q : P, P_2 \in \Theta_{\texttt{OPTIONAL}}(P_1) \wedge P_2 \texttt{ OPTIONAL } P \\ \texttt{AND}, & \text{else.} \end{cases}$$

The function $\kappa(P)$ allows to determine how P is connected to other graph patterns in a graph pattern decomposition, e.g., $\forall P_i \in \Theta_{\texttt{UNION}}(P) : \kappa(P_i) = \texttt{UNION}$. We incorporate the results from both $\kappa(P)$ and $\Theta(P)$ in the algorithm presented in the next section. This information allows us to decide whether two graph patterns can be matched to one another or not.

4 Query Templates

In real-world applications, a large number of queries processed by a SPARQL endpoint exhibit similar structures and vary only in a certain number of resources. In this section, we present query templates that can be used to cluster these similar SPARQL query structures. To identify such query structures, we present a triple pattern similarity measure that is used for our recursive graph pattern matching algorithm. If the algorithm detects a match between the query patterns of two queries, they share a common query template.

4.1 Triple Pattern Similarity and Merging

We first define similar triple patterns that can be mapped to and merged with one another. To establish a mapping between two triple patterns $T_1 = (s_1, p_1, o_1)$ and $T_2 = (s_2, p_2, o_2)$, we try to match the individual elements of T_1 with the corresponding part of T_2, i.e., we align x_1 with x_2 for $x \in \{s, p, o\}$. To calculate the distance of such mappings, we introduce the score $\Delta(x_1, x_2)$:

$$\Delta(x_1, x_2) := \begin{cases} \dfrac{d(x_1, x_2)}{max(|x_1|, |x_2|) + 1} * k, & \text{if } x_1 \in V \wedge x_2 \in V, \text{ with } 0 \leq k < 1 \\ \dfrac{d(x_1, x_2)}{max(|x_1|, |x_2|) + 1}, & \text{if } (x_1 \in U \wedge x_2 \in U) \vee (x_1 \in L \wedge x_2 \in L) \\ 1, & \text{else.} \end{cases}$$

Here, V, U, L are the sets of variables, URIs, and literals, respectively, $|x|$ is the string length of x and $d(x_1, x_2) \rightarrow \mathbb{N}_0$ is a string distance measure with $d(x_1, x_2) = 0 \Leftrightarrow x_1 = x_2$. In our work, we use the Levenshtein distance. Notice that we apply the Levenshtein distance on the entire resource strings, i.e., including possible prefix definitions for URIs or types for literals.

If two queries are identical in structure and content except for their variable names, the binding result set of those variables retrieved from a SPARQL endpoint is the same. Hence, we assume that variables can be mapped more easily to one another than URIs or literals, and apply a factor $k \leq 1$ to $\Delta(x_1, x_2)$, if both x_1 and x_2 are variables. In our implementation, we use $k = \frac{1}{3}$.

To evaluate how easily two triple patterns can be merged, we introduce the triple pattern distance score $\Delta(T_1, T_2)$ that sums up the individual distance scores, i.e., $\Delta(T_1, T_2) := \Delta(s_1, s_2) + \Delta(p_1, p_2) + \Delta(o_1, o_2)$.

```
1 ?p1 foaf:firstName "Alice" .
2 ?p1 ?associationWith example:Bob .
3 example:Bob foaf:firstName "Bob" .
4 example:Bob foaf:lastName "Alice" .
5 ?p2 foaf:firstName "Carol" .
6 ?p2 ?associationWith ?p1 .
```

Listing 2. Triple pattern similarity example

Consider the first triple pattern T_1 in Listing 2: The minimum distance score between T_1 and all other triple patterns shown, i.e., $min(\Delta(T_1, T_2), \ldots, \Delta(T_1, T_6))$, is $(\frac{1}{12} + 0 + \frac{5}{8}) \approx 0.71$ for T_5. For T_2, the minimum value is $(\frac{1}{12} + 0 + 1) \approx 1.08$ for T_6, and for T_3 it is $(0 + \frac{3}{15} + \frac{5}{8}) \approx 0.83$ for T_4.

We also allow the calculation of distance scores between two graph patterns P_1, P_2 as follows:

$$\Delta(P_1, P_2) := \begin{cases} \Delta(T_1, T_2), & \text{if } \Theta(P_1) = \{T_1\} \wedge \Theta(P_2) = \{T_2\} \\ \infty, & \text{else.} \end{cases}$$

This notation mainly serves as a shorthand for analyzing graph patterns with size 1, i.e., graph patterns that constitute triple patterns.

Finally, we introduce the generalization function $\lambda(T_1, T_2) = \hat{T}$ that takes as input two triple patterns T_1, T_2 and merges them into one $\hat{T} = (\hat{s}, \hat{p}, \hat{o})$. It does so by replacing the non-equal triple pattern elements between $T_1 = (s_1, p_1, o_1)$ and $T_2 = (s_2, p_2, o_2)$ with arbitrary, uniquely named variables. More formally, we first define $\lambda(x_1, x_2)$ on the triple pattern parts with $x \in \{s, p, o\}$:

$$\lambda(x_1, x_2) := \begin{cases} x_1, & \text{if } \Delta(x_1, x_2) = 0 \\ ?var, & \text{else.} \end{cases}$$

Here, $?var$ represents a variable unique to both triple patterns. The distance of any two of these introduced variables $\Delta(?var_1, ?var_2) = 0$. Thus,

$$\lambda(T_1, T_2) := (\lambda(s_1, s_2), \lambda(p_1, p_2), \lambda(o_1, o_2))$$

In particular, this means that $\hat{T} = T_1$ iff $\Delta(T_1, T_2) = 0$, i.e., no merging is necessary, if the two triple patterns are identical. As with Δ, we use the shorthand notation $\lambda(P_1, P_2)$, if $|P_1| = |P_2| = 1$.

4.2 Graph Pattern Matching

Using the triple pattern distance notion, we can now derive matchings between graph pattern by mapping their individual triple patterns. We consider the task of finding a match a variation of the stable marriage problem [7], which we solve greedily using Algorithm 1. The recursive algorithm takes as arguments two graph patterns P_1, P_2, a maximum distance threshold Δ_{max} for mapping any two triple patterns, and an existing mapping between triple patterns. This mapping is initially empty and is established in polynomial time by iterating over all graph patterns contained in P_1 and P_2. If no complete matching between P_1 and P_2 can be derived, the result of the algorithm is an empty set of mappings.

Two necessary conditions for a match are $\kappa(P_1) = \kappa(P_2)$ and $|\Theta(P_1)| = |\Theta(P_2)|$ (Line 1). Hence, the algorithm does not establish a match between graph patterns with different keywords or sizes. These conditions are necessary, as there might exist partial (i.e., non-perfect) matches between graph patterns of different sizes, but we are interested in discovering only complete matches.

The algorithm traverses over the graph patterns P_1^i contained in S_1 (which is initialized with the results of $\Theta(P_1)$) and tries to match these graph patterns with the graph patterns P_2^j in S_2 (comprising the results of $\Theta(P_2)$) (Line 1). In case both graph patterns currently in consideration have size 1, i.e., they are triple patterns (Line 1), the algorithm checks whether a mapping can be established between these two triple patterns.

Given that P_1^i and P_2^j exhibit the same keyword (Line 1), a mapping between the two triple pattern can be established under two conditions: (i) $\Delta(P_1^i, P_2^j) \leq \Delta_{max}$ and there is currently no other mapping between P_2^j and another triple pattern (Line 1), or (ii) the current mapping of P_2^j has a higher distance score to it than $\Delta(P_1^i, P_2^j)$ (Line 1). In the first case, the mapping is established, in the second case, the existing mapping is changed accordingly, and the previously mapped element P_1^* is again added to S_1 (Line 1). This ensures that the algorithm tries to discover a new match for P_1^* in a subsequent iteration. In both cases, the algorithm sets the value of the Boolean variable $foundMapping$ to `true` and continues by examining the next element in S_1.

If P_1^i and P_2^j are not triple patterns, i.e., their size is greater than 1, the algorithm is executed recursively, using P_1^i and P_2^j along with *mappings* as arguments (Line 1). If *mappings* has changed, either because there were new mappings added or previous mappings altered, $foundMapping$ is set to `true`.

If throughout this iteration, no mapping was discovered between P_1^i and P_2^j, i.e., $foundMapping$ is `false`, the returned mappings are empty (Line 1). Potentially, some mappings could have been determined throughout the recursion and added to *mappings*, while overall the current graph pattern P_1^i cannot be matched to any other graph pattern. To avoid partial matches, *mappings* is returned only if matches were established for all child graph patterns.

As mentioned earlier, Algorithm 1 determines mappings between two triple patterns T_1, T_2 only if they reside in graph patterns P_1, P_2 with identical keyword and size (Line 1). While there might exist a triple pattern T_i in another graph

Algorithm 1. GraphPatternMatching

Input : P_1, P_2 : Two graph patterns
Input : Δ_{max} : Triple pattern distance threshold
Input : *mappings* : Current triple pattern mappings
Output: *mappings* : Symmetric triple pattern mappings between P_1, P_2

```
1  S_1 ← Θ(P_1), S_2 ← Θ(P_2)
2  if κ(P_1) ≠ κ(P_2) ∨ |S_1| ≠ |S_2| then
3      return ∅
4  while S_1 ≠ ∅ do
5      P_1^i ← S_1.pollFirst()
6      foundMapping ← false
7      foreach P_2^j ∈ S_2 do
8          if |P_1^i| = 1 ∧ |P_2^j| = 1 then
9              if κ(P_1^i) = κ(P_2^j) then
10                 P_1^* ← mappings.get(P_2^j)
11                 if P_1^* = NIL then
12                     if Δ(P_1^i, P_2^j) ≤ Δ_max then
13                         mappings.put(P_2^j, P_1^i)
14                         foundMapping ← true
15                         break
16                 else
17                     if Δ(P_1^i, P_2^j) < Δ(P_1^*, P_2^j) then
18                         mappings.put(P_2^j, P_1^i)
19                         S_1.add(P_1^*)
20                         foundMapping ← true
21                         break
22             else
23                 oldMappings ← mappings
24                 mappings ← GraphPatternMatching(P_1^i, P_2^j, mappings)
25                 if mappings ≠ ∅ ∧ mappings ≠ oldMappings then
26                     foundMapping ← true
27     if ¬foundMapping then
28         return ∅
29 return mappings
```

pattern P_i with $i > 2$ and a lower distance score $\Delta(T_1, T_i) < \Delta(T_1, T_2)$, these cannot be mapped, e.g., because of different keywords $\kappa(P_1) \neq \kappa(P_i)$.

Hence, any non-empty set of mappings resulting from Algorithm 1 can be considered stable in the sense that the mapped triple patterns have minimal distance to their mapping partner with respect to the graph pattern they are contained in. If there exists another possible mapping with a lower distance score for a particular triple pattern, this mapping would have been established instead of the current one (Line 1). Note however that the algorithm prefers

the first possible triple pattern mapping over all other possible mappings with identical triple pattern distance (Lines 1 and 1). If for any evaluated graph pattern no match could be determined, the overall return value of the algorithm is an empty set of mappings (Line 1). Conversely, any non-empty mapping result is complete (or perfect) and therefore maximal (the size of non-empty mappings is determined by the number of triple patterns contained in the graph pattern).

4.3 Query Templates and Clusters

Using the output of Algorithm 1, we can now discover *query templates*. The idea of query templates builds on the findings discussed in [14], where the authors mine SPARQL query log files to determine the behavior of agents issuing the respective query. We extend this approach by establishing a formal definition of what constitutes a query template and how to find it. In contrast to previous work, we also show a concrete application of query templates in the next section.

To generate a query template, we evaluate the mappings generated by GraphPatternMatching$(P_{Q_1}, P_{Q_2}, 1, \emptyset)$ for two SPARQL queries Q_1, Q_2 with query patterns P_{Q_1}, P_{Q_2}, respectively. If the output of Algorithm 1 is empty, no query template can be derived. Otherwise, we initialize the query template $\hat{Q}$ with the query Q_1 and replace all triple patterns T_i in $\hat{Q}$ with the merged triple pattern $\hat{T}$ that resulted from $\lambda(T_i, T_j)$ where $(T_i, T_j) \in map$. Whereas in general we require any introduced variable to be unique with relation to other variables in both P_{Q_1}, P_{Q_2}, if we observe a repeated merge between two identical triple pattern parts, e.g., two consecutive $\lambda(s_i, s_j) \neq s_i$, we re-use the variable introduced for the first merge. Finally, we consider $\hat{Q}$ to be a query template, if $\hat{Q} \neq Q_1$, i.e., $Q_1 \neq Q_2$ and at least one triple pattern mapping $(T_1, T_2) \in map$ is non-trivial.

Recall that variables introduced during merging have distance 0 to each other as defined in Subsec. 4.1. Hence, if we determine two query templates that are identical except for the identifiers of the introduced variables, we consider them to represent the same template. Thus, all queries sharing a query template form a *query cluster*, which may overlap. Notice that all queries in a cluster can be matched to the cluster's query template, albeit potentially only for $\Delta_{max} > 1$. We assume that for most queries in such a cluster a single resource or literal is replaced throughout all triple patterns as indicated by the findings in [14].

5 Evaluation

To evaluate our template discovery approach we analyzed the DBpedia 3.6 query log files from the USEWOD 2012 dataset [2]. In total, these files contain around 8.5 million anonymized queries received by the public DBpedia endpoint[6] on 14 individual days in 2011. We chose these particular log files mainly for three reasons: (i) the query intention is to some extent comprehensible to non-domain experts, (ii) the log files exhibit recurring query patterns [14], and (iii) all queries

[6] `http://dbpedia.org/sparql`

are assigned a source (hashed IP address) and timestamp (hourly granularity), allowing us to to coarsely delimit query sessions.

To illustrate the last point, an excerpt of the query log file *2011-01-24.log* is presented in Listing 3. Each line starts with the hashed IP address of the issuing source, followed by the timestamp and the actual query. We found that additional metadata provided in the log files, e.g., the user agent sending the requests, did not provide any information relevant to our work.

Listing 3 also indicates that the level of granularity in the query log is hours. For our experiments, we consider all queries from one user within one hour (i.e., with the same timestamp) to constitute a query session. Moreover, we also map queries in such a query session to the clusters they belong to. Hence, for the rest of our evaluation, query sessions can be considered sequences of query templates uniquely identified by a timestamp and user id.

```
1 237fbf63e8449c1ade56eb7d208ce219 - [24/Jan/2011 01:00:00 +0100] "/sparql/?query..." 200 512 "-" "-"
2 f452f4195b4d2046c77ad98496c1b127 - [24/Jan/2011 01:00:00 +0100] "/sparql/?query..." 200 1024 "-" "Java"
3 9b1d83195dd251275c55c12ac2efa43d - [24/Jan/2011 02:00:00 +0100] "/sparql/?query..." 200 512 "-" "Mozilla"
4 f452f4195b4d2046c77ad98496c1b127 - [24/Jan/2011 02:00:00 +0100] "/sparql/?query..." 200 1024 "-" "-"
```

Listing 3. Abbreviated excerpt from query log file *2011-01-24.log*

5.1 Query Session Analysis

For our first evaluation, we analyze the size, frequency, and contents of query sessions across all users. Figure 1 illustrates how often query sessions of different length occur. We distinguish between homogeneous query sessions, i.e., sessions containing only queries from the same cluster, and heterogeneous query sessions, i.e., sessions containing queries from at least two clusters. Overall, homogeneous query sessions occur far more often than heterogeneous query sessions, even if query sessions of length 1 (which are always homogeneous) are disregarded.

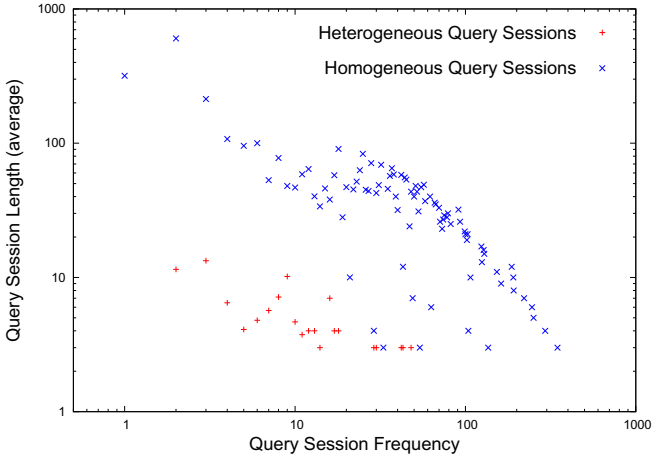


Fig. 1. Length of query sessions correlated with their frequency

Generally, the average length of homogeneous query sessions is also in order of magnitudes higher than the average length of heterogeneous query sessions.

Both these findings, i.e., the high frequency and length of homogeneous query sessions, indicate that most requests received by the DBpedia endpoint are similarly-structured SPARQL queries, most likely issued by machine agents. On the other hand, only a small percentage of relatively short query sessions are heterogeneous, possibly indicating human users querying the DBpedia endpoint.

We also evaluated the conditional probability of sequences of length 2 for all query clusters discovered in the log files and present results in Fig. 2. Here, both axes Q_i and Q_{i+1} of all individual diagrams correspond to the query clusters, where a single tick mark on each axis represents one cluster. Both axes are sorted in descending size of the represented query clusters. The values for $p(Q_{i+1}|Q_i)$ illustrate the probability of observing a query from a certain cluster given the cluster of the previous query. A high value (represented by a darker color) indicates that queries from two query clusters are likely to occur in sequence.

While the plots differ slightly for the various dates in Fig. 2, two general trends can be observed: First, the matrix of all conditional probabilities is sparsely populated, i.e., for a query belonging to any given query cluster discovered in the log, the subsequent query usually belongs to one of a limited number of

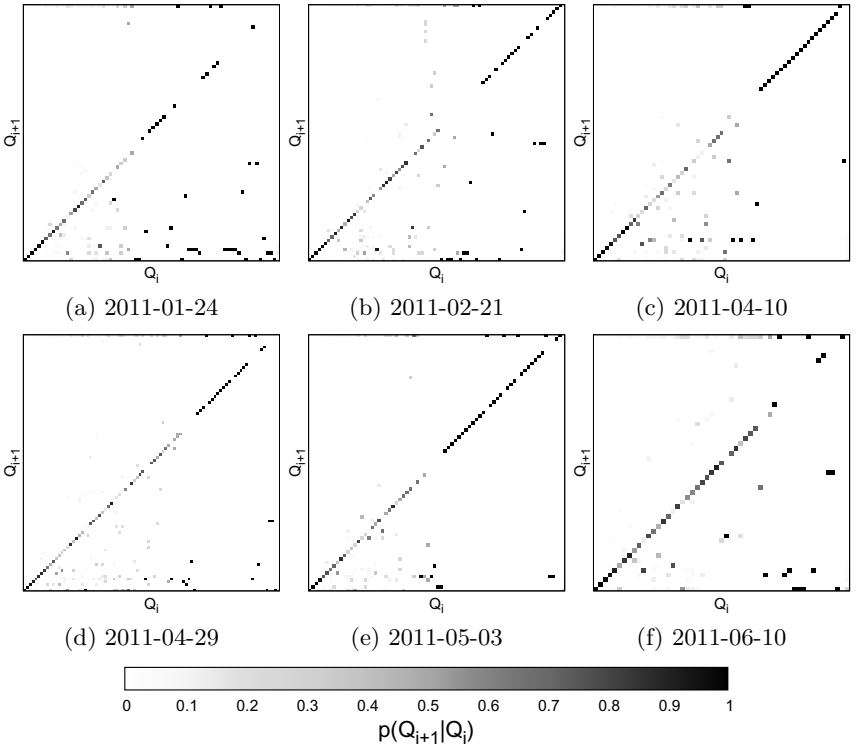


Fig. 2. Conditional probabilities for sequences of length 2 of query templates

clusters. In addition, there is a high probability that queries from one query cluster are followed by queries of the same cluster. This notion is illustrated by the high values on the diagonal of all diagrams.

5.2 Query Template Prefetching

If many individual similarly-structured queries, i.e., queries from the same query cluster, are issued in immediate succession by an agent, as observed in Sec. 5.1, this agent essentially utilizes only parts of the provided data. Moreover, while the relevant information is retrieved one query at a time, it could instead be gathered all at once and used to populate a locally materialized view on the knowledge base. This approach yields advantages for both the endpoint provider and the data consumer by reducing the number of connections on the SPARQL endpoint and eliminating latency overhead (e.g., for query planning, disk access, and data transmission) on each request, respectively.

Consider the sample query in Listing 4: This query retrieves the English language abstract of the resource `dbpedia:Charreada`. Similar queries concerning abstracts were discovered around 3.5 million times for varying subjects in the query log files. The longest individual query sequence consisting only of distinct queries from this cluster issued by a single user contains 56,633 queries. At the time of writing, the DBpedia endpoint provides English abstract information for 3,769,926 resources. Hence, during the longest query sequence around 1.5% of all English DBpedia abstracts are retrieved. We discovered similar request patterns for other query sessions of this user and among query sequences of other users.

```
SELECT ?abstract WHERE {
  dbpedia:Charreada dbpedia-owl:abstract ?abstract .
  FILTER (langMatches(lang(?abstract), "en"))
}
```

Listing 4. SPARQL sample query retrieving the English abstract of a resource

To evaluate the accumulated latency overhead caused by such a large amount of similar queries, we first randomly extracted 100 sample queries from the query cluster containing requests retrieving English abstracts of a resource. Then, we sent these queries to the public SPARQL endpoint. Based on our measurements, the average time between issuing a query and receiving a result was around 5.2 ms. Retrieving the abstracts of 1000 resources using the query template on the other hand took only around 611 ms. Hence, issuing a single query template to retrieve results for related resources instead of multiple queries each retrieving only bindings for one resource leads to an execution speedup of nearly factor 10. For different query templates, we measured similar speedup results.

The benefit of prefetching data for future queries depends on how many queries actually exploit this locally available data. This number is influenced by the length of the analyzed time frame. We illustrate the advantages of prefetching for distinct query sessions in Tab. 1a and for all queries from a specific user within one day in Tab. 1b. Here, we chose the five users (identified by their abbreviated IP hash) with the most queries in the respective time frame.

We identified the most common query cluster in this query set, gathered results for the corresponding template, and materialized these results locally. The coverage rate describes how many of these prefetched results were also retrieved individually by the queries within the respective time frame. Higher coverage indicates that more prefetched results were retrieved by actual queries.

Table 1. Template coverage rates for the top five users with the most queries

(a) Distinct query sessions

User ID	#Queries	Coverage Rate
237...	6,081	54.21%
ea0...	4,951	44.14%
6cb...	3,216	28.67%
e36...	3,106	27.69%
a40...	455	4.05%

(b) All queries within a day

User ID	#Queries	Coverage Rate
237...	68,472	100.00%
f45...	29,235	21.11%
6cb...	18,844	100.00%
5de...	13,500	100.00%
499...	9,747	27.84%

Table 1 illustrates that for a large number of queries issued over a short period of time by a distinct user, i.e., a single hour or day, a local cache containing the data retrieved in advance can efficiently provide results for these queries. This effect becomes more obvious for longer time periods: As Tab. 1b indicates, there are cases when prefetched data can be used for myriads of queries on a single day and all prefetched information is completely utilized during this time frame.

6 Conclusion

In this work, we presented the notion of SPARQL query templates. They represent potentially overlapping clusters of similarly-structured queries, where all elements within a cluster exhibit recurring query patterns and are subsumed by the template. We described an algorithm to detect and extract query templates based on a flexible resource similarity distance function. Furthermore, we evaluated our approach on real-world SPARQL query logs. Here, we discovered three main results: First, the large amount of SPARQL queries received by the DBpedia endpoint can be mapped to a small number of query clusters. In addition, resulting query sessions are mostly homogeneous, i.e., queries from a specific cluster are likely to be followed by queries from the same cluster. Lastly, retrieving combined results for queries from the same cluster instead of issuing individual queries reduces the latency overhead.

We have illustrated a specific use case for query templates by exploiting these findings: Result prefetching. Here, instead of issuing multiple queries from the same cluster, we instead issue the common query template that subsumes these queries. As we have shown in our evaluation, this is particularly useful for longer query sessions. If we assume that a cache containing these prefetched results is maintained in-between query sessions, even more cache hits are generated.

As the findings in this work have proven, there is a huge potential for retrieving semantically relevant data for future queries before these are actually issued.

Whereas the introduced approaches already work well for long query sessions or across query sessions, they do not cater to short query sessions with mixed requests, typically encountered when human agents issue exploratory SPARQL queries. Thus, in future work we plan to extend our query prefetching approach by adapting more sophisticated query rewriting approaches based on common information retrieval strategies for RDF data. Ultimately, our goal is to train a classifier to automatically choose the most suitable of these rewriting methods.

References

1. Bartolomeo, G., Salsano, S.: A spectrometry of linked data. In: Proceedings of the WWW Workshop on Linked Data on the Web (LDOW), Lyon, France (2012)
2. Berendt, B., Hollink, L., Hollink, V., Luczak-Rösch, M., Möller, K., Vallet, D.: USEWOD2012 – 2nd international workshop on usage analysis and the web of data. In: Proceedings of the International World Wide Web Conference (WWW), Lyon, France (2012)
3. Bizer, C., Schultz, A.: The Berlin SPARQL benchmark. International Journal on Semantic Web and Information Systems 5(2), 1–24 (2009)
4. Böhm, C., Lorey, J., Naumann, F.: Creating voiD descriptions for web-scale data. Journal of Web Semantics 9(3), 339–345 (2011)
5. Dar, S., Franklin, M.J., Jónsson, B.T., Srivastava, D., Tan, M.: Semantic data caching and replacement. In: Proceedings of the International Conference on Very Large Databases (VLDB), Bombay, India, pp. 330–341 (1996)
6. Fagni, T., Perego, R., Silvestri, F., Orlando, S.: Boosting the performance of web search engines: Caching and prefetching query results by exploiting historical usage data. ACM Transactions on Information Systems 24(1), 51–78 (2006)
7. Gale, D., Shapley, L.S.: College admissions and the stability of marriage. The American Mathematical Monthly 69(1), 9–15 (1962)
8. Khatchadourian, S., Consens, M.P.: Exploring RDF usage and interlinking in the linked open data cloud using explod. In: Proceedings of the WWW Workshop on Linked Data on the Web (LDOW) (2010)
9. Lehmann, J., Bühmann, L.: AutoSPARQL: Let users query your knowledge base. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) ESWC 2011, Part I. LNCS, vol. 6643, pp. 63–79. Springer, Heidelberg (2011)
10. Martin, M., Unbehauen, J., Auer, S.: Improving the performance of semantic web applications with SPARQL query caching. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010, Part II. LNCS, vol. 6089, pp. 304–318. Springer, Heidelberg (2010)
11. Möller, K., Hausenblas, M., Cyganiak, R., Grimnes, G.A.: Learning from linked open data usage: Patterns & metrics. In: Proceedings of the Web Science Conference, Raleigh, NC, USA (2010)
12. Morsey, M., Lehmann, J., Auer, S., Ngonga Ngomo, A.-C.: DBpedia SPARQL benchmark – performance assessment with real queries on real data. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 454–469. Springer, Heidelberg (2011)
13. Pérez, J., Arenas, M., Gutierrez, C.: Semantics and complexity of SPARQL. ACM Transactions on Database Systems (TODS) 34(3), 16:1–16:45 (2009)

14. Raghuveer, A.: Characterizing machine agent behavior through SPARQL query mining. In: Proceedings of the International Workshop on Usage Analysis and the Web of Data, Lyon, France (2012)
15. Yang, M., Wu, G.: Caching intermediate result of SPARQL queries. In: Proceedings of the International World Wide Web Conference (WWW), Hyderabad, India, pp. 159–160 (2011)
16. Zenz, G., Zhou, X., Minack, E., Siberski, W., Nejdl, W.: From keywords to semantic queries - incremental query construction on the semantic web. Journal of Web Semantics 7(3), 166–176 (2009)

Synonym Analysis for Predicate Expansion

Ziawasch Abedjan and Felix Naumann

Hasso Plattner Institute, Potsdam, Germany
{ziawasch.abedjan,felix.naumann}@hpi.uni-potsdam.de

Abstract. Despite unified data models, such as the Resource Description Framework (RDF) on structural level and the corresponding query language SPARQL, the integration and usage of Linked Open Data faces major heterogeneity challenges on the semantic level. Incorrect use of ontology concepts and class properties impede the goal of machine readability and knowledge discovery. For example, users searching for movies with a certain artist cannot rely on a single given property artist, because some movies may be connected to that artist by the predicate starring. In addition, the information need of a data consumer may not always be clear and her interpretation of given schemata may differ from the intentions of the ontology engineer or data publisher.

It is thus necessary to either support users during query formulation or to incorporate implicitly related facts through predicate expansion. To this end, we introduce a data-driven synonym discovery algorithm for predicate expansion. We applied our algorithm to various data sets as shown in a thorough evaluation of different strategies and rule-based techniques for this purpose.

1 Querying LOD

The increasing amount of Linked Open Data (LOD) is accompanied by an apparently unavoidable heterogeneity as data is published by different data publishers and extracted through different techniques and for different purposes. The heterogeneity leads to data inconsistencies and impedes applicability of LOD and raises new opportunities and challenges for the data mining community [16]. On the structural level, consistency already has been achieved, because (LOD) is often represented in the Resource Description Framework (RDF) data model: a triple structure consisting of a subject, a predicate, and an object (SPO). Each triple represents a statement/fact. This unified structure allows standard query languages, such as SPARQL, to be used. However for real applications also factual consistencies are relevant. When processing RDF data, meta information, such as ontological structures and exact range definitions of predicates, are desirable and ideally provided by a knowledge base. However in the context of LOD, knowledge bases are usually incomplete or simply not available. For example, our recent work showed that there is a mismatch between ontologies and their usage [1]. Evaluations on the DBpedia data set showed that some of the mismatches occurred, because predicates where used that were synonymous to a predicate defined by the ontology (e.g., city or location instead of locationCity). Of course two synonymous

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 140–154, 2013.

predicates may have been defined on purpose for two disjoint purposes, but because they have been used in substitution of each other, the data consumer has to deal with the inconsistency. As we analyzed a SPARQL query workload provided by usewod2012[1] we encountered multiple sets of SPARQL queries that included UNION constructions as illustrated in Table 1. These examples show that applications already try to deal with the predicate inconsistency within the data by expanding their queries with UNION constructions containing synonymously used predicates. These UNION constructions further join dozens of patterns intercepting schema and value errors and abbreviations.

In the fields of traditional information retrieval there are already intuitions and techniques for expanding keyword queries. They comprise techniques for synonym discovery, stemming of words, and spelling corrections. In this work we want to concentrate on the discovery of synonymously used predicates. The discovery of sameAs-links between subject/object resources has already been extensively subject of research. However the discovery of synonymously used predicates has not received any attention at all. Note, we explicitly talk about synonymously used predicates instead of synonym predicates. For example, predicates with more general or specific meaning often substitute each other in the data. E.g., artist is often used as a substitute for starring even though artist is more general than starring.

Table 1. Joined patterns with UNION in DBpedia query logs

Pattern pairs containing synonymous predicates
?company **dbpedia-prop:name** "International Business Machines Corporation"@en ?company **rdfs:label** "International Business Machines Corporation"@en
?place **dbpedia-prop:name** "Dublin"@en. ?place **dbpedia-prop:officialName** "Dublin"@en.
?airport **onto:iataLocationIdentifier** "CGN"@en. ?airport **prop:iata** "CGN"@en.

Synonym discovery is further interesting for the general purpose of enriching an existing synonym thesaurus with new synonyms that have evolved through the time as multiple people use different terms for describing the same phenomenon. Because LOD is formatted in RDF, synonym candidate terms are easy to extract and easier to compare with regard to their contextual occurrence. Note, synonym discovery in unstructured data, such as web documents, needs to consider natural language processing rules. Last but not least, the discovery of synonym predicates benefits the usage of LOD. Furthermore, for many data sources meta-data is only poorly provided. Identifying synonymously used predicates can support the evaluation and improvement of the underlying ontology and schema definitions. Usage of global synonym databases is not sufficient and might lead to misleading facts in this scenario, because of the heterogeneity of LOD, as predicates are used in different knowledge bases for different purposes

[1] http://data.semanticweb.org/usewod/2012/

by different data publishers. So a data-driven approach is necessary to dissolve the existing synonym dependencies.

In this paper, we present an approach for discovering predicate pairs that substitute each other in the data and are good candidates for query expansions. Our approach is based on aggregating positive and negative association rules at statement level based on the concept of mining configurations [2]. We only incorporate information based on the given RDF graph. As a proof-of-concept we applied our algorithm to several LOD sources including the popular DBpedia data set [8].

The rest of this paper is organized as follows: In the next section we present related work with regard to synonym discovery and schema matching. Next we present necessary foundations with regard to RDF and association rules. In Section 4 we describe our algorithm. We evaluate our approach and strategies in Section 5 and conclude in Section 6.

2 Related Work

We apply existing data mining algorithms to the new domain of LOD and propose predicate expansion approach based on synonym discovery. Therefore, we present related work with regard to data mining in the Semantic Web as well as existing applications in the fields of synonym discovery. As most of our techniques for synonym discovery derive from schema matching approaches, we also give an overview of relevant schema matching approaches.

2.1 Mining the Semantic Web

There is already much work on mining the Semantic Web in the fields of inductive logic programming and approaches that make use of the description logic of a knowledge base [21]. Those approaches concentrate on mining answer-sets of queries towards a knowledge base. Based on a general reference concept, additional logical relations are considered for refining the entries in an answer-set. This approach depends on a clean ontological knowledge base, which is usually not available. Furthermore, that approach ignores the interesting opportunities of mining of rules among predicates.

As RDF data spans a graph of resources connected by predicates as edges, another related field of research is mining frequent subgraphs or subtrees [17]. However, in LOD no two different nodes in an RDF graph have the same URI. Therefore, frequency analysis cannot be performed unless we assume duplicate entries in the data set. But if we consider the corresponding type of each URI pattern analysis can be performed, because multiple URIs belong to the same type. Thus, any graph mining would be restricted to type mining and not data mining.

Among profiling tools, ProLOD is a tool for profiling LOD, which includes association rule mining on predicates for the purpose of schema analysis [9]. An approach based on predicate mining was introduced for reconciling ontologies[1].

As similar approach was also used for schema induction [22].The method of mining association rules on predicates is also applied in our work, however we go further than just analyzing the schema and show a concrete application that is based on this method and show how it can be combined to rule mining scenarios that also involve the objects of RDF statements.

2.2 Query Expansion and Synonym Discovery

Research on query expansion includes stemming techniques, relevance feedback, and other dictionary based approaches [6]. On their technical level the approaches do not apply to our SPARQL scenario as we do not retrieve documents but structured entities. So far, Shady et al. have already presented a query expansion approach based on language models [12]. Our approach is based on association rules and a more simplistic model and we were able to process large datasets, such as DBpedia, in couple of minutes. Most existing work for discovering synonyms is based on different language processing and information retrieval techniques. A common approach is to look for co-occurrence of synonym candidates in web documents [7,23]. The idea behind this approach is that synonymous word co-occur in documents [15]. So they calculate the ratio of real co-occurrence of two terms and the independent occurrence of each term. Note that for these approaches there are already known candidate pairs that have to be validated. In our scenario this assumption does not hold, as we also have to retrieve the candidate pairs.

While Baronis work [7] concentrates on globally valid synonyms the authors of [23] address context sensitive synonym discovery by looking at co-clicked query results. Whenever the distance between two clusters of clicked query results is below a certain threshold, the query terms can be seen as synonyms.

The approaches so far are very different from our domain where we want to discover synonym schema elements in RDF data. An approach that has a similar characteristic is the synonym discovery approach based on extracted webtables [10]. The authors introduce a metric that enables to discover synonyms among table attributes. However their approach is quite restrictive: they assume a context attribute given for making attributes comparable. Furthermore, they ignore instance-based techniques as they process only extracted table schemata.

2.3 Schema Matching

Schema matching differs from synonym discovery within schemata in the sense that two schema elements may be synonyms but still may not share a remarkable number of values. On the other hands two attributes may share a lot of values but their corresponding labels may not be synonyms from a global point of view. Still approaches for the discovery of attribute matches and synonyms follow similar intuitions. According to the classification of Rahm and Bernstein [20], we would classify our approach as a mixture of an instance-based and a schema level matching algorithm. At schema level we apply existing techniques to RDF data and evaluate their effectivity.

Existing instance-based approaches are different from our work as they compare the content of each attribute column-wise [11,13,19]. Choosing features for matching is cumbersome and algorithms that look for value overlaps lack efficiency. We propose an association rule based approach that discovers overlaps between attribute values in an Rdf corpus.

One could also perform schema matching on element level by using dictionaries, however the performance of those approaches has been poor in real data scenarios [18]. In this paper we want to focus on mining based features for synonym discovery.

3 Preliminaries

Our approach is based on association rule mining that is enabled by two mining configurations introduced by [2]. First of all we give a brief introduction to concept of association rule mining. Next we introduce of mining configurations for Rdf data and outline how we apply them to our use case.

3.1 Association Rule Mining

The concept of association rules has been widely studied in the context of market basket analysis [3], however the formal definition is not restricted by any domain: Given a set of items $I = \{i_1, i_2, \ldots, i_m\}$, an association rule is an implication $X \rightarrow Y$ consisting of the *itemsets* $X, Y \subset I$ with $X \cap Y = \emptyset$. Given a set of transactions $T = \{t | t \subseteq I\}$, association rule mining aims at discovering rules holding two thresholds: minimum support and minimum confidence.

Support s of a rule $X \rightarrow Y$ denotes the fraction of transactions in T that include the union of the *antecedent* (left-hand side itemset X) and *consequent* (right-hand side itemset Y) of the rule, i.e., $s\%$ of the transactions in T contain $X \cup Y$. The confidence c of a rule denotes the statistical dependency of the *consequent* of a rule from the *antecedent*. The rule $X \rightarrow Y$ has confidence c if $c\%$ of the transactions T that contain X also contain Y. Algorithms to generate association rules decompose the problem into two separate steps:

1. Discover all frequent itemsets, i.e., itemsets that hold minimal support.
2. For each frequent itemset a generate rules of the form $l \rightarrow a - l$ with $l \subset a$, and check the confidence of the rule.

While the second step of the algorithm is straightforward, the first step marks the bottleneck of any algorithm. The three best known approaches to this problem are Apriori [4], FP-Growth [14], and Eclat [24]. For each of these algorithms, there exist multiple modifications and optimizations. We use the FP-Growth algorithm for our paper.

3.2 Mining Configurations

To apply association rule mining to Rdf data, it is necessary to identify the respective item set I as well as the transaction base T and its transactions.

Table 2. Facts in SPO structure from DBpedia

Subject	Predicate	Object
Obama	birthPlace	Hawaii
Obama	party	Democrats
Obama	orderInOffice	President
Merkel	birthPlace	Hamburg
Merkel	orderInOffice	Chancellor
Merkel	party	CDU
Brahms	born	Hamburg
Brahms	type	Musician

Table 3. Six configurations of context and target

Conf.	Context	Target	Use case
1	Subject	Predicate	Schema discovery
2	Subject	Object	Basket analysis
3	Predicate	Subject	Clustering
4	Predicate	Object	Range discovery
5	Object	Subject	Topical clustering
6	Object	Predicate	Schema matching

Our mining approach is based on the subject-predicate-object (SPO) view of RDF data as briefly introduced in [2]. Table 2 illustrates some SPO facts extracted from DBpedia. For legibility, we omit the complete URI representations of the resources and just give the human-readable values.

Any part of the SPO statement can be regarded as a *context*, which is used for grouping one of the two remaining parts of the statement as the *target* for mining. So, a transaction is a set of target elements associated with one context element that represents the transaction id (TID). We call each of those *context* and *target* combinations a *configuration*. Table 3 shows an overview of the six possible configurations and their preliminarily identified use-cases. Each can be further constrained to derive more refined configurations. For instance, the subjects may be constrained to be of type Person, as happens to be the case in our example.

The application of Configuration 1 from Tab. 3 to our example data set would transform the facts into three transactions, one for each distinct subject as illustrated in Tab. 4a. In this example, the itemset {*birthPlace, party, orderInOffice*} is a frequent itemset (support 66.7%), implying rules, such as *birthPlace* → *orderInOffice, party* and *orderInOffice* → *birthPlace, party* with 66.7% and 100% confidence, respectively. Furthermore, we can infere negative rules, such as *birthPlace* → ¬ *born*.

Configuration 6 in the context of objects would create the transactions presented in Tab. 4b. The frequent itemsets here contain predicates that are similar in their ranges, e.g., {*born, birthPlace*}. Given the negative rule in Conf. 1 and the pattern in Conf. 6, one could conclude that both predicates born and birthPlace have synonymous meanings.

Table 4. Configuration examples

(a) Context: Subject, Target: Predicate

TID	transaction
Obama	{*birthPlace, party, orderInOffice*}
Merkel	{*birthPlace, party, orderInOffice*}
Lennon	{*birthPlace, instrument*}

(b) Context: Object, Target: Predicate

TID	transaction
Musician	{*type*}
Hamburg	{*born, birthPlace*}
Hawaii	{*birthPlace*}
President	{*orderInOffice*}

4 Generation of Candidates for Predicate Expansion

Our approach aims at discovering all possible predicate pairs where each predicate could be the expansion of the other one. Having identified all such candidate pairs the expansion candidates of a given predicate can easily be retrieved by retrieving all pairs in which the respective to be expanded predicate occurs.

We introduce three basic strategies that we combine for the discovery of these candidate pairs. The first two strategies make direct usage of the mining configurations from Tab. 3. With Configuration 1 we perform schema analysis in the context of subjects. Configuration 6 enables us to mine similar predicates in the context of objects. Additionally, we look into range structure of predicates by looking at value type distributions. All three approaches are derived from existing schema-level and instance-based schema matching techniques.

4.1 Schema Analysis

Configuration 1 enables us to do frequency analysis and rule discovery per entity. For instance, positive rules between predicates can be used for re-validating existing ontologies [1]. In our use case we have a different intuition: Expansion candidates for a predicate should not co-occur with it for any entity. It is more likely for entities to include only one representative of a synonymous predicate group within their schema, e.g., either starring or artist. That is why we look for negative correlations in Configuration 1. For this purpose we developed an FP-Growth [14] implementation that retrieves all negative correlations for a set of candidate pairs. The approach can also be used stand-alone looking at all possible pairs that have a negative correlation in the data set. Negative correlation can be expressed by several score functions. One could look at the bidirectional correlation coefficient or consider some kind of aggregations of the negative rules' confidence values. In the following we describe each of the used scoring functions at schema level.

Confidence Aggregations. The confidence *conf* of the rule $p_1 \rightarrow \neg p_2$ describes the probability c% of predicate p_2 not to occur for the same entity where p_1 occurs. We refer to these rules as negative rules. If p_2 was a rare predicate that, however, occurs always with p_1, $conf(p_1 \rightarrow \neg p_2)$ might be considerably high however $conf(p_2 \rightarrow \neg p_1)$ would be close to 0%. Therefore we need to aggregate both confidence values. We experimented using the three aggregations maximum, minimum, and F-Measure (harmonic mean).

Reversed Correlation Coefficient. The drawback of confidence aggregation is that the scoring ignores the overall relevance of a pair within a data set. We apply the formula given in [5], which measures the linear relationship between two predicates:

$$cCoeff(X,Y) = \frac{N \cdot supp(X,Y) - supp(X) \cdot supp(Y)}{\sqrt{supp(Y) \cdot (N - supp(Y)) \cdot supp(X) \cdot (N - supp(X))}}$$

where N denotes the total number of baskets in the mining configuration, which, for Configuration 1, is equivalent to the total number of entities (distinct subjects) in the data set. For ranking purposes we reverse the sign of $cCoeff(X, Y)$, as we want to have positive scores on negative correlations. We label the score reversed correlation coefficient (RCC).

***Syn*-Function.** In [10] the authors introduce the *syn*-function for synonym discovery in different webtables. Their assumptions are also that synonyms never occur together. In case of LOD 'never' is a too strong assumption. Furthermore, their score is based on a set of context attributes. Unfortunately the authors did not mention how to choose this context attribute, if domain knowledge is not given. Nevertheless, the intuition behind their function that two synonymous predicates have the same odds in occurring with other predicates can also be applied in our scenario. Thus, we also adapted this score function and compared the results to the scoring functions named before.

Bare schema analysis leads to results also including incorrect pairs, such as `recordLabel` and `author` as both occur for different entities. While songs have the predicate `recordLabel`, books have the predicate `author`. So a negative correlation is not a sufficient condition for a predicate to be expanded by another. The context or the range of the predicates should also be taken into account. In the following we describe our strategies that complement the schema analysis by considering also the range of predicates.

4.2 Range Content Filtering

Our second intuition is that as synonym predicates have a similar meaning they also share a similar range of object values. Normally when trying to compute the value overlap between two predicates one would look at the ratio of overlaps depending on the total number of values of such a predicate. We apply a more efficient range content filtering approach (RCF) based on mining configurations (see Sec. 3.2).

Configuration 6 constitutes a mining scenario where each transaction is defined by a distinct object value. So each transaction consists of all predicates containing the distinct object value in there range. Frequent patterns in this configuration are sets of predicates that share a significant number of object values in their range. As each configuration is an adaption of frequent itemset mining the threshold that decides whether two predicates are similar or not is minimum support and depends on the number of all baskets or all existing distinct objects. Furthermore, our approach ignores value overlaps that occur due to multiple occurrence of one distinct value in the ranges. We analyze the effect of these differences and show that our approach is much more efficient without any loss of quality. Similar to the schema analysis strategy also the range content filtering based on value overlaps is not a sufficient condition for discovering synonymously used predicates. For example the predicates `birthPlace` and `deathPlace` share a remarkable percentage of their ranges but are obviously not

used synonymously. However this candidate pair can be pruned looking at their exclusion rate per entity during schema analysis.

4.3 Range Structure Filtering

In some scenarios value range content filtering might not be the most appropriate technique as it requires two synonym predicates to share a portion of exactly equal values. However, real world data might contain synonymous predicates with completely disjoint range sets where the range elements are only ontologically similar. This is often the case when looking at predicates describing numbers and dates. Therefore existing work looks not only at exact overlaps but also on general string or token characteristics, such as string length and character distributions [11,19]. As the goal of this paper is to analyse the capabilities graph data and statement level mining, we do not dive into literal similarities and character distributions. Furthermore our experiments showed that based on range similarity we are already able to discover all pairs that contain values with similar ranges. Instead we look at type distributions in predicate ranges. So for every object in the range of a predicate we retrieve its type from the graph. Then we create type vectors per predicate where each component contains the number of the occurrences of one type. As each entity in RDF might have several types due to existing type hierarchies, i.e., Barack Obama is a Politician as well as a Person, we considered only the most specific type of an entity.

Having type vectors for a predicate pair, the range type similarity can be computed using measures, such as cosine similarity or weighted Jaccard similarity. Preliminary experiments showed that weighted Jaccard similarity seems more promising because cosine similarity results into high scores as soon as one component value of one vector is very large although all other components have very small values. Missing type values, e.g., in case of dates and other numerical values, have been handled as unknown types, whereas no two unknown types are equal.

4.4 Combined Approach

We have introduced three different ways of generating and evaluating synonym candidate pairs. It is crucial to find a reasonable order for combining those three to make best use of the intuitions and achieve optimal quality and to be efficient at the same time. We decided on the following order: (1) first retrieve all predicate pairs through range content filtering, (2) filter those pairs by range structure filtering and then (3) analyze their schema co-occurrences. This strategy has two advantages: as retrieving negative correlations and type vectors is time-consuming, it is reasonable to perform both on given candidates instead of using them on the complete data set to retrieve candidates. Furthermore, the minimum support threshold for range value overlapping is a more expressive threshold than arbitrary correlation and scoring thresholds on schema level, which are more suited for ranking purposes of the filtered candidates. Consider that type range filtering can be applied only to data sets for which the type information is available. In our experiments

we could use the type filtering approach only for the DBpedia data, and even there it did not contribute to the precision on top of range content filtering.

5 Evaluation

We evaluated our approach with regard to precision and recall of generated expansion candidates. Table 5 shows the data sets with the corresponding numbers of distinct triples, subjects, predicates, and objects used for experiments. Because DBpedia contains data from different domains, we also performed our experiments on subsets of a certain domain, such as people and places. In the following we first show to which extent each component of our algorithm contributes to the quality of query expansion candidate analysis. Then we show overall precision results on multiple data sets. Last, we illustrate the efficiency gain of our frequent itemset based overlap discovery method towards the standard value-overlap approach.

Table 5. Datasets for evaluations

Source	#triples	#predicates	#subjects	#objects
Magnatune	243,855	24	33,643	68,440
Govwild	7,233,610	35	963,070	2,648,360
DBpedia 3.7	17,518,364	1,827,474	1,296	4,595,303
DBpedia Person	4,040,932	237	408,817	982,218
DBpedia Organisation	1,857,849	304	169,162	731,136
DBpedia Work	2,611,172	136	262,575	751,916

5.1 Step-Wise Evaluation of Recall and Precision

To evaluate the components of our algorithm, it is necessary to be able to classify good and poor expansion candidates. For this purpose, we manually classified 9,456 predicates pairs of a dataset. The classification of predicate pairs for expansion appropriateness is cumbersome, because one has to look for defined ranges, example values, and consider query intentions using these predicates. We chose the data sets with the lowest number of predicates, Magnatune, and the data set comprising all entities of type

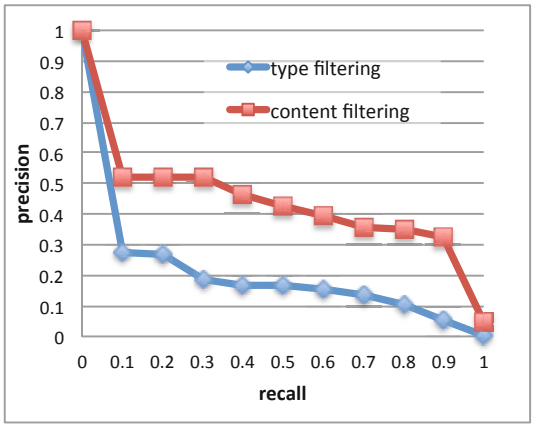
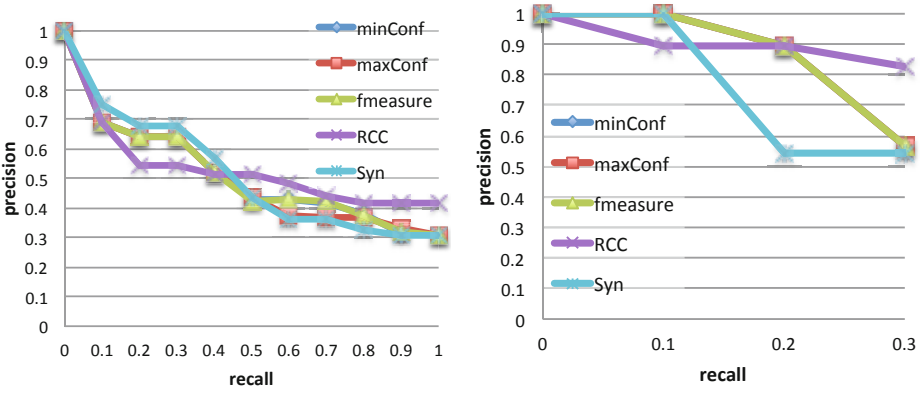


Fig. 1. Precision recall curve for the filtering methods

Work from DBpedia. A predicate pair is annotated as a correct expansion pair if both predicates are appropriate candidates for expanding the respective other predicate. Each classification result was validated by three experts (computer scientists). All in all, we discovered 82 expansion candidate pairs among the predicates for Work entities and 9 candidates in the Magnatune data set, out of 9,180 and 276 pairs, respectively.

First, we evaluated the precision/recall curve of the range-content and the range-type filtering approaches on the Work dataset as illustrated in Fig. 1. For this purpose we sorted all pairs twice, once with regard to their support value and once with regard to the weighted Jaccard distance of their range types. As the diagram illustrates, both approaches perform better than a random approach, which results in 0.8% precision on a recall of 100%. However, the precision of the range-content filtering method is on all recall levels better than the precision achieved with range-type filtering.



(a) 0.01% support (b) 0.1% support

Fig. 2. Precision recall curve of schema scores on the Work dataset

Figures 2a and 2b illustrate the ranking improvement of the algorithm using the schema scores. We chose the support thresholds 0.1% and 0.01% where the content filtering part resulted in 52% and 22% precision and recall levels of 28% and 98% respectively (see Fig. 1). At the support threshold of 0.01% the range content filtering achieved 22% precision and 98% recall. Figure 2a shows that all schema scores result in better precision on this recall level. Furthermore, on lower recall levels the precision is higher by orders of magnitudes. The precision improvement can be explained through the fact that predicate pairs with a very similar range but different semantics, such as album and previousWork, achieve lower scores on schema level as they often appear together. Looking at Fig. 2b the only difference is that at the highest possible recall level of 28% only the RCC score leads to better results. In general it can be observed that at high recall levels the RCC score performs better than all other scoring functions, while on low recall levels the *Syn* function performs better.

Regarding the results for Magnatune in Fig. 3, we can observe very high precision values even with range content filtering. However, even at a support threshold of 0.0% the schema-based scoring functions all perform better. If we raise the minimum support threshold to 0.01% or 0.1%, the precision remains 100% for all approaches, however the recall falls to 89% and 44%, respectively.

Next, we evaluate the precision of our combined approach on these two minimum support thresholds and fixed schema scoring thresholds.

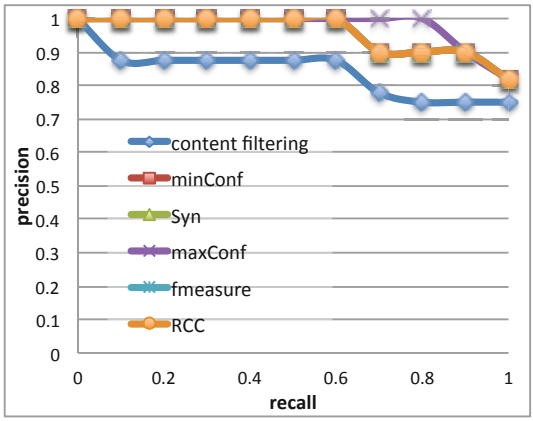


Fig. 3. Precision recall on Magnatune with 0.0% minimum support

5.2 Precision Quality

To evaluate the different approaches we defined minimum thresholds as follows: For the minimum, maximum, and f-measure confidence scores we fixed the threshold at 50% minimum confidence. For the RCC and *Syn* scores we set the threshold as >0.0. For RCC only scores above 0.0 indicate any negative correlation. The closer the value is to 0.0 the more random is the co-occurrence of two predicates. The *Syn* function results in scores above zero only if there is a significant correlation of the predicates. However, because the value is not normalized within a certain range, there is no basis for the choice of a higher threshold. That is why we use here the absolute value 0.0 as a threshold.

Comparing both Tables 6 and 7 one can see the precision improvement by leveraging the support threshold for RCF. Furthermore, one can observe that all schema scores behave very similar. The only significant differences can be observed for the Govwild data set, where minimum and f-measure confidence retrieve no correct results at all. The reason is that the Govwild dataset comprises data from different domains, such as people, locations, and organisations. That leads to false positives like name and city, because both people and organisations are connected to a city with the city attribute, while triples with cities as their subject use name for labeling the same city RDF object. The same reason also applies to the experiments on the complete DBpedia 3.7 data set. Looking at more specific domain data, such as Magnatune or DBpedia Work and Organisation the results are much better. Of course the numbers of retrieved results are much smaller, because the algorithm was able to filter nearly all true negatives.

One can conclude that the more cautious the thresholds are chosen the better quality can be achieved on all data sets. On data sets containing entities of very different domains, the algorithm produces too many false positives, so it is

Table 6. Precision at 0.01% RCF minimum support

Dataset	minConf	maxConf	f-Measure	RCC	Syn	RCF	# RCF results
Magnatune	**100%**	87.5%	**100%**	**100%**	87.5%	87.5%	8
Govwild	0%	**20%**	0%	14%	0%	20%	25
DBpedia 3.7	**32%**	**32%**	**32%**	15%	22%	32%	1115
DBpedia Person	32%	32%	32%	**35%**	26%	32%	308
DBpedia Work	49%	52%	50%	**61%**	60%	22%	256
DBpedia Organisation	**33%**	32%	32%	31%	32%	32%	412

Table 7. Precision values at 0.1% range content filtering minimum support

Dataset	minConf	maxConf	fMeasure	RCC	Syn	RCF	# RCF results
Magnatune	**100%**	**100%**	**100%**	**100%**	**100%**	100%	4
Govwild	0%	**56%**	0%	50%	0%	50%	10
DBpedia 3.7	40%	43%	38%	**46%**	45%	36%	64
DBpedia Person	56%	49%	50%	**60%**	-	40%	35
DBpedia Work	73%	57%	74%	78%	**89%**	52%	46
DBpedia Organisation	88%	86%	90%	89%	**95%**	85%	45

Table 8. Runtime experiment results

Dataset	RCF @ 0.1% support	RCF @ 0.01% support	naive RCF
Magnatune	4,116 ms	4,417 ms	18,122 ms
Govwild	66,297 ms	67,676 ms	> 3h
DBpedia Work	93,876 ms	97,676 ms	> 3h
DBpedia 3.7 (complete)	122,412 ms	127,964 ms	> 3h

always reasonable to perform the algorithm on each domain fraction of the data set separately. Performing the experiments on entities of the more specific type Actor that is a subclass of Person, we achieved much better precision, e.g., RCF and RCC values were 65% and 87% respectively.

5.3 Efficiency Analysis

We stated that our RCF approach for discovering value overlaps using Configuration 6 (see Sec. 3.2) is more efficient than pairwise comparison of predicates. Table 8 illustrates some runtime comparisons; we aborted runs after three hours. Our mining-based RCF approaches are always by faster than the naïve overlap approach by orders of magnitude, because predicate pairs with no overlap are filtered early. Furthermore the runtime of our approach is adaptive to support thresholds in the manner of frequent item mining, as it filters predicate pairs below the specified threshold in beforehand.

The total runtime of our algorithm including range content filtering and schema analysis is below 8 minutes for each presented dataset at a minimum support of 0.1% for range content filtering and below 10 minutes at the threshold of 0.01%. The experiments have been performed on a notebook with a 2.66 GHz Intel Core Duo processor and 4 GB DDR3 memory.

6 Conclusion

In this paper we addressed data inconsistencies due to synonymously used predicates in RDF and introduced the concept of predicate expansion for SPARQL patterns. We presented several strategies for automatically discovering expansion candidates. We showed the strength and weakness of the strategies on different datasets, proposing a stacked algorithm based on range content filtering and schema analysis. Our evaluation showed that our algorithm performs very good on data containing only subjects of one domain, but produces more false positives on RDF data where the subjects represent entities of many different types. We believe that providing an optional predicate expansion interface at SPARQL endpoints is useful. An alternative approach is to (semi-)automatically remove or change facts with wrongly used predicates, based on the results of our synonym discovery.

References

1. Abedjan, Z., Lorey, J., Naumann, F.: Reconciling ontologies and the web of data. In: Proceedings of the International Conference on Information and Knowledge Management (CIKM), Maui, Hawaii, pp. 1532–1536 (2012)
2. Abedjan, Z., Naumann, F.: Context and target configurations for mining RDF data. In: Proceedings of the International Workshop on Search and Mining Entity-Relationship Data (SMER), Glasgow (2011)
3. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of the ACM International Conference on Management of Data (SIGMOD), Washington, D.C., USA, pp. 207–216 (1993)
4. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules in Large Databases. In: Proceedings of the International Conference on Very Large Databases (VLDB), Santiago de Chile, Chile, pp. 487–499 (1994)
5. Antonie, M.-L., Zaïane, O.R.: Mining positive and negative association rules: An approach for confined rules. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) PKDD 2004. LNCS (LNAI), vol. 3202, pp. 27–38. Springer, Heidelberg (2004)
6. Baeza-Yates, R.A., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., Boston (1999)
7. Baroni, M., Bisi, S.: Using cooccurrence statistics and the web to discover synonyms in technical language. In: International Conference on Language Resources and Evaluation, pp. 1725–1728 (2004)
8. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia - A crystallization point for the Web of Data. Journal of Web Semantics (JWS) 7, 154–165 (2009)

9. Böhm, C., Naumann, F., Abedjan, Z., Fenz, D., Grütze, T., Hefenbrock, D., Pohl, M., Sonnabend, D.: Profiling linked open data with ProLOD. In: Proceedings of the International Workshop on New Trends in Information Integration (NTII), pp. 175–178 (2010)
10. Cafarella, M.J., Halevy, A., Wang, D.Z., Wu, E., Zhang, Y.: WebTables: exploring the power of tables on the web. Proceedings of the VLDB Endowment 1, 538–549 (2008)
11. Doan, A., Domingos, P., Halevy, A.Y.: Reconciling schemas of disparate data sources: a machine-learning approach. In: Proceedings of the ACM International Conference on Management of Data (SIGMOD), New York, NY, pp. 509–520 (2001)
12. Elbassuoni, S., Ramanath, M., Weikum, G.: Query relaxation for entity-relationship search. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) ESWC 2011, Part II. LNCS, vol. 6644, pp. 62–76. Springer, Heidelberg (2011)
13. Gottlob, G., Senellart, P.: Schema mapping discovery from data instances. Journal of the ACM 57(2), 6:1–6:37 (2010)
14. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: Proceedings of the ACM International Conference on Management of Data (SIGMOD), pp. 1–12 (2000)
15. Harris, Z.: Distributional structure. Word 10(23), 146–162 (1954)
16. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. Morgan Claypool Publishers (2011)
17. Kuramochi, M., Karypis, G.: Frequent subgraph discovery. In: Proceedings of the IEEE International Conference on Data Mining (ICDM), Washington, D.C., pp. 313–320 (2001)
18. Li, W.-S., Clifton, C.: Semint: A tool for identifying attribute correspondences in heterogeneous databases using neural networks. Data and Knowledge Engineering (DKE) 33(1), 49–84 (2000)
19. Naumann, F., Ho, C.-T., Tian, X., Haas, L.M., Megiddo, N.: Attribute classification using feature analysis. In: Proceedings of the International Conference on Data Engineering (ICDE), p. 271 (2002)
20. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB Journal 10(4), 334–350 (2001)
21. Rettinger, A., Lösch, U., Tresp, V., d'Amato, C., Fanizzi, N.: Mining the semantic web - statistical learning for next generation knowledge bases. Data Min. Knowl. Discov. 24(3), 613–662 (2012)
22. Völker, J., Niepert, M.: Statistical schema induction. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) ESWC 2011, Part I. LNCS, vol. 6643, pp. 124–138. Springer, Heidelberg (2011)
23. Wei, X., Peng, F., Tseng, H., Lu, Y., Dumoulin, B.: Context sensitive synonym discovery for web search queries. In: Proceedings of the International Conference on Information and Knowledge Management (CIKM), New York, NY, USA, pp. 1585–1588 (2009)
24. Zaki, M.J.: Scalable Algorithms for Association Mining. IEEE Transactions on Knowledge and Data Engineering (TKDE) 12, 372–390 (2000)

Instance-Based Ontological Knowledge Acquisition

Lihua Zhao[1,2] and Ryutaro Ichise[2,1]

[1] The Graduate University for Advanced Studies
[2] National Institute of Informatics, Tokyo, Japan
{lihua,ichise}@nii.ac.jp

Abstract. The Linked Open Data (LOD) cloud contains tremendous amounts of interlinked instances, from where we can retrieve abundant knowledge. However, because of the heterogeneous and big ontologies, it is time consuming to learn all the ontologies manually and it is difficult to observe which properties are important for describing instances of a specific class. In order to construct an ontology that can help users easily access to various data sets, we propose a semi-automatic ontology integration framework that can reduce the heterogeneity of ontologies and retrieve frequently used core properties for each class. The framework consists of three main components: graph-based ontology integration, machine-learning-based ontology schema extraction, and an ontology merger. By analyzing the instances of the linked data sets, this framework acquires ontological knowledge and constructs a high-quality integrated ontology, which is easily understandable and effective in knowledge acquisition from various data sets using simple SPARQL queries.

Keywords: Semantic Web, linked data, ontology integration, knowledge acquisition, machine learning.

1 Introduction

The Linked Open Data (LOD) is a collection of machine-readable structured data with over 31 billion Resource Description Framework (RDF) triples interlinked by around 504 million *SameAs* links (as of Sep. 2011). Instances are represented using the Uniform Resource Identifier (URI), and identical instances are linked with the built-in OWL property *owl:sameAs* [3]. The Web Ontology Language (OWL) is a semantic markup language developed as a vocabulary extension of the RDF with more vocabularies for describing properties and classes [2]. RDF Schema is a simple vocabulary for describing properties and classes of RDF resources. The OWL 2 Web Ontology Language [16] provides classes and properties as the old OWL 1[2], but with richer data types, data ranges, and disjoint properties, etc.

The LOD cloud has been growing rapidly over the past years and many Semantic Web applications have been developed by accessing the linked data sets [4]. However, in order to use the data sets, we have to understand their heterogeneous ontologies in advance. One possible solution for the ontology heterogeneity

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 155–169, 2013.

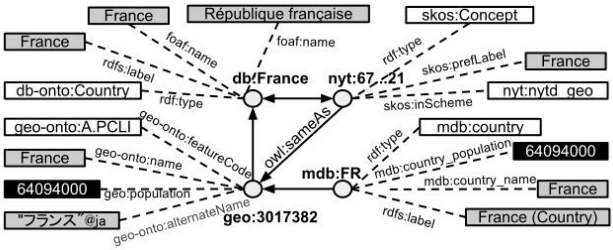


Fig. 1. Interlinked Instances of "France"

problem is constructing a global ontology that integrates various ontologies in the LOD cloud. Ontologies commonly consist of classes, properties, and relations between them. Although the built-in properties owl:equivalentClass and owl:equivalentProperty are designed to link classes or properties with the same concept, there are only few links at a class or property level [7]. Hence, it is difficult to directly retrieve equivalent classes and properties for integrating ontologies of various data sets.

In order to integrate ontologies of various data sets, we need to identify related classes and properties of the ontologies. Since the same instances are linked by *owl:sameAs*, we can create undirected graphs with the linked instances. Fig. 1 shows the interlinked instances of "France" and each instance is described using properties and objects. As shown in Fig. 1, all the properties (labeled on the dotted line) connected with the grey boxes (objects) represent the name of "France" and the properties connected to the black boxes represent the population. By analyzing the graph patterns, we can observe how the same classes and properties are represented differently in various data sets and integrate them.

Other than integrating related classes and properties, we also need frequently used core classes and properties to construct a high-quality integrated ontology. Machine learning methods such as association rule learning and rule-based classification can be applied to discover core properties for describing instances in a specific class. Apriori is a well-known algorithm for learning association rules in a big database [1], while the rule-based learning method - Decision Table can retrieve a subset of properties that leads to high prediction accuracy with cross-validation [10].

In this paper, we propose a framework that semi-automatically integrates heterogeneous ontologies for the linked data sets. The integrated ontology consists of frequently used core classes and properties that can help Semantic Web application developers easily understand the ontology schema of the data sets. Furthermore, the integrated ontology also includes related classes and properties, with which we can integrate data sets and find missing links between instances.

This paper is organized as follows. In Section 2, we discuss some related work and the limitation of their methods. In Section 3, we introduce our semi-automatic ontology integration framework in details. The experiments are discussed in Section 4. We conclude and propose future work in Section 5.

2 Related Work

The authors in [11] introduced a closed frequent graph mining algorithm to extract frequent graph patterns from the Linked Data Cloud. Then, they extracted features from the entities of the graph patterns to detect hidden *owl:sameAs* links or relations in geographic data sets. They applied a supervised learning method on the frequent graph patterns to find useful attributes that link instances. However, their approach only focused on geographic data and did not discuss about what kind of features are important for finding the hidden links.

A debugging method for mapping lightweight ontologies is introduced in [13]. They applied machine learning method to determine the disjointness of any pair of classes, with the features of the taxonomic overlap, semantic distance, object properties, label similarity, and WordNet similarity. Although their method performs better than other ontology matching systems, their method is limited to the expressive lightweight ontologies.

In [14], the authors focused on finding concept coverings between two sources by exploring disjunctions of restriction classes. Their approach produces coverings where concepts at different levels in the ontologies can be mapped even there is no direct equivalence. However, the work is mainly for specific domains and the alignments of ontologies are limited between two resources.

In contrast to the research described above, our approach retrieves related ontology schema and frequently used core properties and classes in each data set. Our method is domain-independent and successfully integrates heterogeneous ontologies by extracting related properties and classes that are critical for interlinking instances. In addition, for the instances of specific classes, we can recommend core properties that are frequently used for the instance description.

3 Semi-automatic Ontology Integration Framework

Constructing a global ontology by integrating heterogeneous ontologies of the linked data can help effectively integrate various data resources. In order to create an integrated ontology and decrease the ontology heterogeneity problem, we focus on retrieving related classes and properties, top-level classes, and frequent core properties. We can extract related classes and properties from the interlinked instances using the graph-based ontology integration component. In addition, we also need the top-level classes and frequent core properties in each data set, which can be extracted using machine learning methods. For instance, the Decision Table algorithm can retrieve a subset of properties that leads to high prediction accuracy with cross-validation and the Apriori algorithm can discover properties that occur frequently in the instances of top-level classes.

In this paper, we propose a semi-automatic ontology integration framework, which is an extension of the previous work in [18]. The semi-automatic ontology integration framework is shown in Fig. 2, which consists of three main components: graph-based ontology integration, machine-learning-based ontology schema extraction, and an ontology merger. In the following, we will describe each component in details.

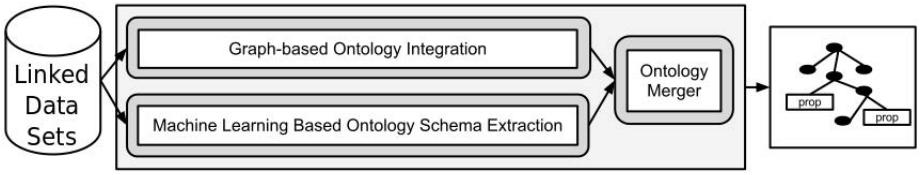


Fig. 2. Framework for Semi-automatic Ontology Integration

3.1 Graph-Based Ontology Integration

The graph-based ontology integration component semi-automatically finds related classes and properties by analyzing *SameAs* graph patterns in the linked data sets [18]. We will briefly describe this component, which is shown in Fig. 3. This component consists of five processes: graph pattern extraction, <Predicate, Object> collection, related classes and properties grouping, aggregation for all graph patterns, and manual revision.

Graph Pattern Extraction. Since the instances which refer to the same thing are interlinked by *owl:sameAs* in the LOD cloud, we collect all the linked instances and construct graph patterns according to the *SameAs* graphs extraction algorithm introduced in [18]. All the same *SameAs* graphs consist of a graph pattern, from which we can detect related classes and properties.

<Predicate, Object> Collection. An instance is described by a collection of RDF triples in the form of <subject, predicate, object>. Since a *SameAs* graph contains linked instances, we collect all the <Predicate, Object> (PO) pairs of the interlinked instances as the content of a *SameAs* graph and classify the PO pairs into five different types: Class, Date, URI, Number, and String.

Related Classes and Properties Grouping. We track subsumption relations to group related classes and apply different similarity matching methods to group related properties. In the following, we discuss how to retrieve and group related classes and properties from different types of PO pairs.

Class. For the PO pairs of type Class, we retrieve related classes from the most specific classes of the linked instances by tracking the subsumption relations such as owl:subClassOf and skos:inScheme. The classes and subsumption relations compose a tree, where the most specific classes are called leaf nodes in the tree.

Date and URI. We perform exact matching on the types of Date and URI, because even a slight difference of object values may refer to totally different properties.

Number and String. For the types of Number and String, the object values may vary in different data sets. For instance, the population of a country may be slightly different in diverse data sets and the values in String may have different representations for the same meaning. Hence, in order to discover similar

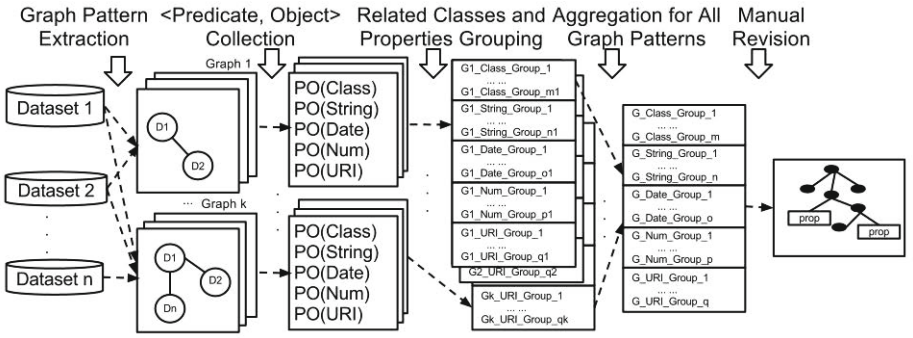


Fig. 3. Graph-Based Ontology Integration

properties for the types of Number and String, we apply similarity matching approach by extending the methods introduced in [19].

The string-based and knowledge-based similarity matching methods are commonly used to match ontologies at the concept level [5]. In our approach, we adopted three string-based similarity measures, namely, JaroWinkler distance [17], Levenshtein distance, and n-gram, as introduced in [8]. String-based similarity measures are applied to compare the objects of *PO* pairs that are classified in String.

The knowledge-based similarity measures are applied to compare the preprocessed terms of predicates, because most of the terms have semantic meanings that can be recognized as a concept. To extract the concepts of predicate terms, we pre-process the predicates of *PO* pairs by performing natural language processing (NLP). We adopted nine knowledge-based similarity measures [15], namely, LCH, RES, HSO, JCN, LESK, PATH,WUP, LIN, and VECTOR, which are based on WordNet [6] (a large lexical database of English).

Aggregation for All Graph Patterns. In this step, we aggregate the integrated groups of classes and properties from all the graph patterns. An integrated ontology is automatically constructed with the integrated sets of related classes and properties, automatically selected terms, and the designed relations that link groups of related classes and properties to the integrated ontology schema.

Manual Revision. The automatically integrated ontology of this framework includes related classes and properties from different data sets. However, not all the terms of classes and properties are properly selected. Hence, we need experts to work on revising the integrated ontology by choosing a proper term for each group of properties, and by amending wrong groups of classes and properties. Since the integrated ontology is much smaller than the original ontology schema, it is a lightweight work.

The graph-based ontology integration component can discover related classes and properties from various data sets. By analyzing the extracted graph

patterns, we detect related classes and properties which are classified into different data types: Date, URI, Number, and String. Similar classes are integrated by tracking subsumption relations and different similarity matching methods are applied on different types of *PO* pairs to retrieve similar properties. We automatically integrate related classes and properties for each graph pattern, and then aggregate all of them.

3.2 Machine-Learning-Based Ontology Schema Extraction

Although, the graph-based ontology integration method can retrieve related classes and properties from different ontologies, it misses some classes and frequent core properties. Therefore, we need another method to find top-level classes and frequent core properties, which are essential for describing instances.

By applying machine learning methods, we can find frequent core properties that are used to describe instances of a specific class. The Decision Table algorithm is a rule-based algorithm that can retrieve a subset of core properties and the Apriori algorithm can find a set of associated properties that are frequently used for describing instances. Hence, we apply the Decision Table and the Apriori algorithm to retrieve top-level classes and frequent core properties from the linked data sets.

In order to perform the machine learning methods, we randomly select a fixed number of instances for each top-level class from the data sets. For the data sets built based on ontology schema, we track subsumption relations to retrieve the top-level classes. For instance, we track owl:subClassOf subsumption relation to retrieve the top-level classes in DBpedia and track skos:inScheme in Geonames. However, some data sets use categories without any structured ontology schema. For this kind of data sets, we use the categories as top-level classes. For example, NYTimes instances are only categorized into people, locations, organizations, and descriptors. We use this strategy to collect the top-level classes in each data set, and then extract properties that appear more than the frequency threshold θ. The selected instances, properties, and top-level classes are used for performing machine learning methods.

Decision Table. The Decision Table is a simple rule-based supervised learning algorithm, which leads to high performance with simple hypothesis [10]. The Decision Table algorithm can retrieve a subset of core properties that can predict unlabeled instances with a high accuracy. Hence, properties retrieved by the Decision Table play an important role in the data description.

We convert the instances of linked data sets into data that is adaptable to the Decision Table algorithm. The data consists of a list of weights of properties and class labels, where the weight represents the importance of a property in an instance and the labels are top-level classes. The weight of a property in an instance is calculated in a similar way as the TF-IDF (Term Frequency - Inverse Document Frequency), which is often used as a weighting factor in information retrieval and text mining [12]. The TF-IDF value reflects how important a word

is to a document in a collection or corpus. The weight of each property in an instance is defined as the product of property frequency (PF) and the inverse instance frequency (IIF) in a similar way as the TF-IDF. The $pf(prop, inst)$ is the frequency of the property *prop* in the instance *inst*.

The inverse instance frequency of the property *prop* in the data set D is $iif(prop, D)$, calculated as follows:

$$iif(prop, D) = \log \frac{|D|}{|inst_{prop}|}$$

where $inst_{prop}$ indicates an instance that contains the property *prop*. The value of $iif(prop, D)$ is the logarithm of the ratio between the number of instances in D and the number of instances that contain the *prop*. If *prop* appears in *inst*, the weight of *prop* is calculated according to the following equation:

$$weight(prop, inst) = pf(prop, inst) \times iif(prop, D)$$

The properties retrieved with the Decision Table in each data set are critical for describing instances in the data set. Hence, we use these retrieved properties and top-level classes as parts of the final integrated ontology.

Apriori. Association rule learning method can extract a set of properties that occur frequently in instances. Apriori is a classic association rule mining algorithm, which is designed to operate on the databases of transactions. A frequent itemset is an itemset whose support is greater than the user-specified minimum support. Each instance in a specific class represents a transaction, and the properties that describe the instance are treated as items. Hence, the frequent itemsets represent frequently used properties for describing instances of a specific class. The frequent core properties can be recommended to the data publishers or help them find missing important descriptions of instances.

For each instance, we collect a top-level class and all the properties that appear in the instance as a transaction data. The Apriori algorithm can extract associated sets of properties that occur frequently in the instances of a top-level class. Hence, the retrieved sets of properties are essential for describing instances of a specific class. Furthermore, we can either identify commonly used properties in each data set or unique properties used in the instances of each class. Therefore, the properties extracted with the Apriori algorithm are necessary for the integrated ontology.

3.3 Ontology Merger

The third component is an ontology merger, which merges the ontology classes and properties extracted from the previous two components. The graph-based ontology integration component outputs groups of related classes and properties. On the other hand, the machine-learning-based ontology schema extraction component outputs a set of core properties retrieved by the Decision Table and a set of properties along with a top-level class retrieved using the Apriori.

We adopt OWL 2 for constructing an integrated ontology. During the merging process, we also add relations between classes and properties so that we can easily identify what kind of properties are used to describe instances of a specific class. We obey the following rules to construct the integrated ontology, where "ex-onto" and "ex-prop" are the prefixes of the integrated ontology.

Class. Related classes are collected from the graph-based ontology integration component and the top-level classes in each data set are collected from the machine-learning-based ontology schema extraction component.

Groups of classes from graph-based ontology integration. Related classes from different data sets are extracted by analyzing *SameAs* graph patterns and grouped into $cgroup_1, cgroup_2, ..., cgroup_z$. For each group, we automatically define a term *ex-onto:ClassTerm* for each group, where the $ClassTerm$ is the most frequent term in the group. For each class $c_i \in cgroup_k$, we add a triple $<$ *ex-onto:ClassTerm*$_k$, *ex-prop:hasMemberClasses*, $c_i >$.

Classes from machine-learning-based ontology schema extraction. Top-level classes in each data set are added to the integrated ontology. If a top-level class $c_i \notin cgroup_k (1 \leq k \leq z)$, we create a new group $cgroup_{z+1}$ for c_i and a new term *ex-onto:ClassTerm*$_{z+1}$ for the new group. Then we add a triple $<$ *ex-onto:ClassTerm*$_{z+1}$, *ex-prop:hasMemberClasses*, $c_i >$.

Property. The extracted properties from two components are merged according to the following rules. At first, we extract the existing property type and domain information of each property from the data sets. The property type is mainly defined as rdf:Property, owl:DataTypeProperty, and owl:ObjectProperty. If the type is not clearly defined, we set the type as rdf:Property.

Groups of properties from graph-based ontology integration. Related properties from various data sets are extracted by analyzing *SameAs* graph patterns and grouped into $pgroup_1, pgroup_2, ..., pgroup_p$. For each group, we choose the most frequent term *ex-onto:propTerm*. Then, for each property $prop_i \in pgroup_t$ $(1 \leq t \leq p)$, we add a triple $<$ *ex-onto:propTerm*$_t$, *ex-prop:hasMemberProperties*, $prop_i >$.

Properties from machine-learning-based ontology schema extraction. We automatically add domain information for the properties retrieved using the Apriori method. For each property *prop* extracted from the instances of class c, we add a triple $<$ *prop*, *rdfs:domain*, $c >$, if it's not defined in the data set.

The ontology merger constructs an integrated ontology using the triples created as above. The global integrated ontology constructed with the ontology merger can help us easily access to various data sets and discover missing links. Furthermore, the domain information of the properties are automatically added using the results of the Apriori algorithm.

Table 1. Data Sets for Experiments

Data Set	Instances	Selected Instances	Class	Top-level Class	Property	Selected Property
DBpedia	3,708,696	64,460	241	28	1385	840
Geonames	7,480,462	45,000	428	9	31	21
NYTimes	10,441	10,441	5	4	8	7
LinkedMDB	694,400	50,000	53	10	107	60

4 Experiments

In this section, we introduce the experimental data sets. Then we discuss whether we successfully retrieved related classes and properties using the graph-based ontology integration. We also discuss experimental results with the Decision Table and the Apriori algorithm that retrieve top-level classes and frequent core properties. Comparison with the previous work introduced in [18] and other ontology matching tools is also discussed in this section. At last, we discuss use cases with the integrated ontology and propose possible applications.

4.1 Data Sets

We selected DBpedia (v3.6), Geonames (v2.2.1), NYTimes and LinkedMDB from the LOD cloud to evaluate our framework. DBpedia is a cross-domain data set with about 8.9 million URIs and Geonames is a geographic domain data set with more than 7 million distinct URIs. NYTimes and LinkedMDB are both from media-domain with 10,467 and 0.5 million URIs, respectively.

The number of instances in our database are listed in the second column of Table 1. The graph-based ontology integration component uses all the instances in the data sets. For the machine learning methods, we randomly choose samples of the data sets to speed up the modeling process as well as to concern unbiased data size for each top-level class. We randomly selected 5000 instances per top-level class in Geonames and LinkedMDB, 3000 instances per top-level class in DBpedia, and used all the instances in NYTimes. The number of selected instances of DBpedia is less than 84,000, because some classes include less than 3000 instances.

The original number of classes and properties, the number of top-level classes and selected properties for the machine learning methods are listed in the Table 1. We track the subsumption relations such as owl:subClassOf and skos:inScheme to collect the top-level classes. Since there are a big number of properties in the data sets, we filter out infrequent properties that appear less than the frequency threshold θ. For each data set, we manually set a different frequency threshold θ as $\sqrt{n}$, where n is the total number of instances in the data set.

Table 2. Results for the Decision Table Algorithm

Data Set	Average Precision	Average Recall	Average F-Measure	Selected Properties
DBpedia	0.892	0.821	0.837	53
Geonames	0.472	0.4	0.324	10
NYTimes	0.795	0.792	0.785	5
LinkedMDB	1	1	1	11

4.2 Graph-Based Ontology Integration

The graph-based ontology integration component uses all the interlinked instances in the data sets. With this component, we retrieved 13 different graph patterns from the *SameAs* Graphs [18]. In total, we extracted 97 classes from the data sets and grouped them into 48 new classes. Each group contains at least two classes and one class can belong to several groups. For instance, the schema in NYTimes is too general, so that the nyt:nytd_geo belongs to any group that has geographical information. Here, we give an example of the integrated class ex-onto:Country, which contains geo-onto:A.PCLI, geo-onto:A.PCLD, mdb:country, db-onto:Country, and nyt:nytd_geo. This group contains the classes about a country from Geonames, LinkedMDB, and DBpedia, except the general geographic class nyt:nytd_geo from NYTimes.

We retrieved 357 properties from the graph patterns using exact or similarity matching, which are integrated into 38 groups. Because of the heterogeneous infobox properties in DBpedia, some groups contain more than one DBpedia property. For instance, the properties geo-onto:population, mdb:country_population, db-onto:populationTotal, db-prop:populationTotal, and other eight DBpedia properties are integrated into the property ex-prop:population.

The graph-based ontology integration can retrieve related classes and properties from directly or indirectly linked instances by analyzing graph patterns.

4.3 Decision Table

The Decision Table algorithm is used to discover a subset of features that can achieve high prediction accuracy with cross-validation. Hence, we apply the Decision Table to retrieve core properties that are essential in describing instances of the data sets. For each data set, we perform the Decision Table algorithm to retrieve core properties by analyzing randomly selected instances of the top-level classes.

In Table 2, we listed the percentage of the weighted average of precision, recall, and F-measure. Precision is the ratio of correct results to all the results retrieved, and recall is the percentage of the retrieved relevant results to all the relevant results. The F-measure is a measure of a test's accuracy, that considers both the precision and the recall. The F-measure is the weighted harmonic mean of precision and recall, calculated as:

$$F\text{-}measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

The F-measure reaches its best value at 1 and worst value at 0. A higher F-measure value means the retrieved subset of properties can well classify instances with unique and core properties. A lower F-measure fails to classify some instances, because the retrieved properties are commonly used in every instance. In the following, we will discuss the experimental results in each data set using the Decision Table algorithm.

DBpedia. The Decision Table algorithm extracted 53 DBpedia properties from 840 selected properties. For example, the properties db-prop:city, db-prop:debut, db-onto:formationYear, and db-prop:stateName are extracted from DBpedia instances. The precision, recall, and F-measure on DBpedia are 0.892, 0.821, and 0.837, respectively.

Geonames. We retrieved 10 properties from 21 selected properties, such as geo-onto:alternateName, geo-onto:countryCode, and wgs84_post:alt, etc. Since all the instances of Geonames are from geographic domain, the Decision Table algorithm can not well distinguish different classes with these commonly used properties. Hence, the evaluation results on Geonames are very low with 0.472 precision, 0.4 recall, and 0.324 F-measure.

NYTimes. Among 7 properties used in the data set, 5 properties are retrieved using the Decision Table algorithm. The extracted properties are skos:scopeNote, nyt:latest_use, nyt:topicPage, skos:definition, and wgs84_pos:long. In NYTimes, there are only few properties for describing news articles and most of them are commonly used in every instance. The cross-validation test with NYTimes are 0.795 precision, 0.792 recall and 0.785 F-measure.

LinkedMDB. The algorithm can classify all the instances in the LinkedMDB correctly with the 11 properties selected from 60 properties. Other than the commonly used properties such as foaf:page, rfs:label, we also extracted some unique properties such as mdb:performance_performanceid, mdb:writer_writerid, and director_directorid, etc.

The Decision Table algorithm retrieves a subset of core properties that are important to distinguish instances. We feed the extracted properties to the integrated ontology. However, the Decision Table can not find all the core properties in each data set.

4.4 Apriori

The Apriori algorithm is a classic algorithm for retrieving frequent itemsets based on the transaction data. For the experiment, we use the parameters upper and lower bound of minimum support as 1 and 0.2, respectively. We use the default minimum metric as 0.9 for the confidence metric. With a lower minimum support, we can retrieve more properties that frequently appear in the data.

Table 3. Examples of Retrieved Properties with the Apriori Algorithm

Data Set	Class	Properties
DBpedia	db:Event	db-onto:place, db-prop:date, db-onto:related/geo.
	db:Species	db-onto:kingdom, db-onto:class, db-onto:family.
	db:Person	foaf:givenName, foaf:surname, db-onto:birthDate.
Geonames	geo:P	geo-onto:alternateName, geo-onto:countryCode.
	geo:R	wgs84_pos:alt, geo-onto:name, geo-onto:countryCode.
NYTimes	nyt:nytd_geo	wgs84_pos:long.
	nyt:nytd_des	skos:scopeNote.
LinkedMDB	mdb:actor	mdb:performance, mdb:actor_name, mdb:actor_netflix_id.
	mdb:film	mdb:director, mdb:performane, mdb:actor, dc:date.

We retrieved frequently appeared core properties using the Apriori algorithm. Some examples are listed in Table 3. The first column lists the experimental data sets, and the second column lists samples of the top-level classes in each data set. The third column lists some of the retrieved interesting or unique properties from each top-level class. As we can see from Table 3, the place, date and geographic properties are important for describing events. The best-known taxonomies such as kingdom, class, and family are also extracted by analyzing the data of species. From the LinkedMDB, we extracted mdb:actor_netflix_id, mdb:actor_name, and mdb:performance, that are critical for distinguishing different instances. Furthermore, the properties of director, performance, actor and date of a film are extracted from instances in the class mdb:film.

In DBpedia and LinkedMDB, we retrieved some unique properties in each class. However, for Geonames and NYTimes, we only retrieved commonly used properties in the data sets. From the instances of Geonames, we found commonly used properties such as geo-onto:alternateName, wgs84_pos:alt, and geo-onto:countryCode, etc. NYTimes only has few properties that are commonly used in every instance, except the wgs84_pos:long in the nyt:nytd_geo class and skos:scopeNote in the nyt:nytd_des class. Hence, the weighted average F-measure of nyt:nytd_geo and nyt:nytd_des are much higher than other classes.

We retrieved frequent sets of properties in most of the cases except in the db:Planet class. Because db:Planet contains 201 different properties for describing instances, which are sparsely used. In addition, we only retrieved db-onto:title and rdfs:type from db:PersonFunction and only rdfs:type property from db:Sales. This is caused by the lack of descriptions in the instances: most of the instances in db:PersonFunction and db:Sales only defined the class information without other detailed descriptions.

The set of properties retrieved from each class imples that the properties are frequently used for instance description of the class. Hence, for each property *prop* retrieved from the instances of class *c*, we automatically added <*prop*, rdfs:domain, *c*> to assert that *prop* can be used for describing instances in the class *c*. Therefore, we can automatically recommend missing core properties for an instance based on its top-level class.

Table 4. Extracted Classes and Properties

	Previous Work	Machine Learning		Current Work
	Graph-Based Integration	Decision Table	Apriori	Integrated Ontology
Class	97	50 (38 new)	50 (38 new)	135 (38 new)
Property	357	79 (49 new)	119(80 new)	453 (96 new)

4.5 Comparison

The graph-based ontology integration framework introduced in [18] only focuses on the related classes and properties acquisition, that may miss some core properties and classes. Hence, we applied machine learning methods to find out core properties for describing instances. The second column of Table 4 lists the number of classes and properties retrieved with the previous work - graph-based integration method. The next two columns list the number of classes and properties retrieved with the machine learning methods - Decision Table and Apriori. The last column is the final integrated ontology which merged the acquired ontological knowledge from two functional components: graph-based ontology integration and machine-learning-based ontology schema extraction, which includes Decision Table and Apriori algorithms.

With the graph-based ontology integration framework, we retrieved 97 classes and 357 properties, which are grouped into 49 and 38 groups, respectively. The final integrated ontology contains 135 classes and 453 properties that are grouped into 87 and 97 groups, respectively. Both of the Decision Table and the Apriori algorithms are performed on 50 selected top-level classes, among them 38 are not retrieved in the graph-based ontology integration. With the Decision Table, we extracted 79 properties, where 49 are not found in the graph-based ontology integration. The Apriori algorithm discovered 119 properties in total, where 80 properties are newly added. Based on the same data sets, Apriori can retrieve more properties than the Decision Table algorithm. Among the newly retrieved properties, 33 properties are retrieved from both Decision Table and Apriori.

By adding machine-learning-based ontology schema extraction component to the graph-based ontology integration, the final integrated ontology become more concrete with groups of related classes and properties, top-level classes, and core properties that are frequently used in instances. For each retrieved property, we automatically added property type definition and for the properties retrieved with the Apriori results, we automatically added domain information to indicate the relations between properties and classes.

Since most of the ontology matching tools fail to find alignments for the datasets that do not have a well designed ontology schema [9], we cannot use them to find alignments among DBpedia, Geonames, NYTimes, and LinkedMDB. The failure in the ontology alignment is caused by some ontologies that have ambiguous meaning of the concepts or the absence of corresponding concepts in the target dataset. However, our approach can find alignments for the poorly structured datasets by analyzing the contents of the interlinked instances.

4.6 Case Studies

In this section, we introduce some use cases with our integrated ontology. The class of db:Actor and mdb:actor are integrated into ex-onto:Actor, which can be used for discovering missing class information of the linked instances of actors. For instance, the db:Shingo_Katori is only described as a musical artist, but in fact he is also an actor and the DBpedia instance has a link to the mdb-actor:27092. Hence, we should add the class db-onto:Actor to the instance db:Shingo_Katori, because all the instances linked with mdb-actor should be an actor unless it is a wrong linkage.

If we want to link a person from different data sets, we can combine the class which indicates a person with some properties such as the birth date, the place of birth, and the name, etc. However, there exist various properties to describe the same kind of property. For example, we integrated 7 different properties that indicate the birthday of a person into the ex-prop:birthDate. Among them, only the property "db-onto:birthDate" has the default domain definition as db-onto:Person and has the highest frequency of usage, that appeared in 287,327 DBpedia instances. From the definitions of the properties and the number of instances which contain the corresponding properties, we can assume that the properties except "db-onto:birthDate" are mistakenly used when the data providers publish the DBpedia data. Therefore, we can suggest "db-onto:birthDate" as the standard property to represent the birthday of a person, and correct the other properties with this standard property.

Other than recommending standard properties, we also successfully integrated different property descriptions from diverse data sets. For instance, properties geo-onto:population, mdb:country_population, db-onto:populationTotal and other nine DBpedia properties are integrated into the property ex-prop: population. By combining the ex-onto:Country and ex-prop:population, we can detect the same country or countries with similar population.

5 Conclusion and Future Work

We proposed a semi-automatic ontology integration framework that can integrate heterogeneous ontologies by analyzing graph patterns of the interlinked instances and by applying machine learning methods. Grouping related classes and properties can reduce the heterogeneity of ontologies in the LOD cloud. The integrated ontology consists of related classes and properties, top-level classes and frequent core properties that can help Semantic Web application developers easily find related instances and query on various data sets. With the integrated ontology, we can also detect misuses of ontologies in the data sets and can recommend core properties for describing instances.

In future work, we plan to apply ontology reasoning methods to automatically detect and revise mistakes during the ontology merging process. Furthermore, we plan to automatically detect the undefined ranges of properties by analyzing the corresponding objects of properties in the data sets. We will test our framework with more data sets from the public linked data sets.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Proceedings of the Twentieth International Conference on Very Large Data Bases, pp. 487–499 (1994)
2. Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A.: OWL Web Ontology Language Reference. W3C Recommendation (2004), http://www.w3.org/TR/owl-ref/
3. Berners-Lee, T.: Linked Data - Design Issues (2006), http://www.w3.org/DesignIssues/LinkedData.html
4. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. International Journal on Semantic Web and Information Systems 5(3), 1–22 (2009)
5. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer, Heidelberg (2007)
6. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. MIT Press (1998)
7. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. Morgan & Claypool (2011)
8. Ichise, R.: An analysis of multiple similarity measures for ontology mapping problem. International Journal of Semantic Computing 4(1), 103–122 (2010)
9. Jain, P., Hitzler, P., Sheth, A.P., Verma, K., Yeh, P.Z.: Ontology alignment for linked open data. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010, Part I. LNCS, vol. 6496, pp. 402–417. Springer, Heidelberg (2010)
10. Kohavi, R.: The power of decision tables. In: Lavrač, N., Wrobel, S. (eds.) ECML 1995. LNCS, vol. 912, pp. 174–189. Springer, Heidelberg (1995)
11. Le, N.T., Ichise, R., Le, H.B.: Detecting hidden relations in geographic data. In: Proceedings of the 4th International Conference on Advances in Semantic Processing, pp. 61–68 (2010)
12. Manning, C.D., Raghavan, P., Schtze, H.: Introduction to Information Retrieval. Cambridge University Press (2008)
13. Meilicke, C., Völker, J., Stuckenschmidt, H.: Learning disjointness for debugging mappings between lightweight ontologies. In: Gangemi, A., Euzenat, J. (eds.) EKAW 2008. LNCS (LNAI), vol. 5268, pp. 93–108. Springer, Heidelberg (2008)
14. Parundekar, R., Knoblock, C.A., Ambite, J.L.: Discovering concept coverings in ontologies of linked data sources. In: Cudré-Mauroux, P., et al. (eds.) ISWC 2012, Part I. LNCS, vol. 7649, pp. 427–443. Springer, Heidelberg (2012)
15. Pedersen, T., Patwardhan, S., Michelizzi, J.: Wordnet: similarity: Measuring the relatedness of concepts. In: Proceedings of the Nineteenth National Conference on Artificial Intelligence, pp. 1024–1025. Association for Computational Linguistics (2004)
16. W3C OWL Working Group: OWL 2 Web Ontology Language Document Overview. W3C Recommendation (2012), http://www.w3.org/TR/owl2-overview/
17. Winkler, W.E.: Overview of record linkage and current research directions. Tech. rep., Statistical Research Division U.S. Bureau of the Census (2006)
18. Zhao, L., Ichise, R.: Graph-based ontology analysis in the linked open data. In: Proceedings of the Eighth International Conference on Semantic Systems, pp. 56–63. ACM (2012)
19. Zhao, L., Ichise, R.: Integrating ontologies using ontology learning approach. IEICE Transactions on Information and Systems E96-D(1), 40–50 (2013)

Logical Linked Data Compression

Amit Krishna Joshi, Pascal Hitzler, and Guozhu Dong

Kno.e.sis Center, Wright State University, Dayton, OH, U.S.A.
{joshi35,pascal.hitzler,guozhu.dong}@wright.edu

Abstract. Linked data has experienced accelerated growth in recent years. With the continuing proliferation of structured data, demand for RDF compression is becoming increasingly important. In this study, we introduce a novel lossless compression technique for RDF datasets, called Rule Based Compression (RB Compression) that compresses datasets by generating a set of new logical rules from the dataset and removing triples that can be inferred from these rules. Unlike other compression techniques, our approach not only takes advantage of syntactic verbosity and data redundancy but also utilizes semantic associations present in the RDF graph. Depending on the nature of the dataset, our system is able to prune more than 50% of the original triples without affecting data integrity.

1 Introduction

Linked Data has received much attention in recent years due to it's interlinking ability across disparate sources, made possible via machine processable non-proprietary RDF data [18]. Today, large number of organizations, including governments, share data in RDF format for easy re-use and integration of data by multiple applications. This has led to accelerated growth in the amount of RDF data being published on the web. Although the growth of RDF data can be viewed as a positive sign for semantic web initiatives, it also causes performance bottlenecks for RDF data management systems that store and provide access to data [12]. As such, the need for compressing structured data is becoming increasingly important.

Earlier RDF compression studies [3,6] have focused on generating a compact representation of RDF. [6] introduced a new compact format called *HDT* which takes advantage of the powerlaw distribution in term-frequencies, schema and resources in RDF datasets. The compression is achieved due to a compact form representation rather than a reduction in the number of triples. [13] introduced the notion of a lean graph which is obtained by eliminating triples which contain blank nodes that specify redundant information. [19] proposed a user-specific redundancy elimination technique based on rules. Similarly, [21] studied RDF graph minimization based on rules, constraints and queries provided by users. The latter two approaches are application dependent and require human input, which makes them unsuitable for compressing the ever growing set of linked datasets.

In this paper, we introduce a scalable lossless compression of RDF datasets using automatic generation of *decompression rules*. We have devised an algorithm

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 170–184, 2013.

to automatically generate a set of rules and split the database into two smaller disjoint datasets, viz., an *Active* dataset and a *Dormant* dataset based on those rules. The dormant dataset contains list of triples which remain uncompressed and to which no rule can be applied during decompression. On the other hand, the active dataset contains list of compressed triples, to which rules are applied for inferring new triples during decompression.

In order to automatically generate a set of rules for compression, we employ frequent pattern mining techniques [9,15]. We examine two possibilities for frequent mining - a) within each property (hence, intra-property) and b) among multiple properties (inter-property). Experiments reveal that RB compression performs better when inter-property transactions are used instead of intra-property transactions. Specifically, the contribution of this work is a rule-based compression technique with the following properties:

- The compression reduces the number of triples, without introducing any new subjects, properties or objects.
- The set of decompression rules, R, can be automatically generated using various algorithms.
- The compression is lossless.

A very preliminary and limited version of this paper appeared in [14].

2 Preliminaries

2.1 Frequent Itemset Mining

The concept of frequent itemset mining [1] (FIM) was first introduced for mining transaction databases. Over the years, frequent itemset mining has played an important role in many data mining tasks that aim to find interesting patterns from databases, including association rules and correlations, or aim to use frequent itemsets to construct classifiers and clusters [7]. In this study, we exploit frequent itemset mining techniques on RDF datasets for generating logical rules and subsequent compressing of RDF datasets.

Transaction Database. Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of distinct items. A set $X = \{i_1, i_2, \ldots, i_k\} \subseteq I$ is called an itemset, or a k-itemset if it contains k items. Let D be a set of transactions where each transaction, $T = (tid, X)$, contains a unique transaction identifier, tid, and an itemset X. Figure 1 shows a list of transactions corresponding to a list of triples containing the rdf:type[1] property. Here, subjects represent identifiers and the set of corresponding objects represent transactions. In this study, we use the following definitions for intra- and inter-property transactions.

Intra-property transactions. For a graph G containing a set of triples, an intra-property transaction corresponding to a property p is a set $T = (s, X)$ such that s is a subject and X is a set of objects, i.e. (s, p, o_x) is a triple in graph G; o_x is a member of X.

[1] *rdf:type* is represented by *a*.

Inter-property transactions. For a graph G containing a set of triples, an inter-property transaction is a set $T = (s, Z)$ such that s is a subject and each member of Z is a pair (p_z, o_z) of property and object, i.e. (s, p_z, o_z) is a triple in graph G.

s1 a 125 s4 a 125.
s1 a 22. s4 a 22.
s1 a 225. s4 a 225.
s1 a 60. s4 a 60.
s6 a 90. s6 a 22.
s5 a 125. s5 a 22.
s2 a 225. s2 a 125.
s2 a 22. s3 a 81.
s3 a 22.

(a) Triples

TID	rdf:type
S1	125,22,225,60
S2	125,22,225
S3	81,22
S4	125,22,225,60
S5	125,22
S6	90,22

(b) Transactions

Fig. 1. Triples and corresponding transactions

Item (k)	Frequent Patterns (F_k)
225	{([22, 225], 525786)}
60	{([22, 225, 60], 525786)}
189	{([22, 227, 83, 189], 60194)}
213	{([22, 227, 83, 189, 213], 60194)}
173	{([22, 103, 26, 304, 173], 57772)}
70	{([22, 70], 56372), ([22, 103, 26, 304, 173, 70], 31084), ([22, 202, 42, 70], 25288)}
13	{([22, 225, 60, 174, 13], 53120)}
235	{([22, 225, 60,174,235],52305), ([22, 225, 60, 202,42, 174,235],480)}
126	{([22, 191, 97, 222, 126], 49252)}

(a) Frequent Patterns

Item	Object
22	owl:Thing
227	dbp:Work
189	dbp:Film
213	schema:Movie
103	dbp:Person
26	schema:Person
304	foaf:Person
173	dbp:Artist
225	dbp:Place
60	schema:Place

(b) object mappings

Fig. 2. Sample frequent patterns generated for *DBpedia Ontology Types* dataset. Each item represents a numerically encoded object. An item can be associated with multiple frequent patterns as seen for item 70.

Support and Frequent Itemset. The *support* of an itemset X, denoted by $\sigma(X)$, is the number of transactions in D containing X. Itemset X is said to be *frequent* if $\sigma(X) \geq \sigma_{min}$ (σ_{min} is a minimum support threshold).

Itemset Mining. A frequent itemset is often referred to as a *frequent pattern*. Numerous studies have been done and various algorithms [1,2,9,22,23] have been proposed to mine frequent itemsets. In this study, we use the *FP-Growth* [9] algorithm for generating frequent itemsets. We represent the output of FP-Growth as a set of pairs (k, F_k), where k is an item, and F_k, a set of frequent patterns corresponding to k. Each frequent pattern is a pair of the form (v, σ_v). v is an itemset of a frequent pattern and σ_v is a support of this frequent pattern.

Definition 1. *Let D be a transaction database over a set I of items, and σ_{min} a minimum support threshold. The set of frequent itemsets in D with respect to σ_{min} is denoted by $F(D, \sigma_{min}) := \{X \subseteq I | \sigma(X) \geq \sigma_{min}\}$*

Figure 2(a) shows several frequent patterns for *DBpedia Ontology Types* dataset containing only the rdf:type property.[2] To generate such frequent patterns, we first create a transaction database as shown in Figure 1 and then use parallel FP-Growth to compute frequent patterns. Please refer to [9,15] for details about the FP-Growth algorithm and its implementation. Figure 3 shows the list of inter-property frequent patterns for one of the linked open datasets.

Item	Frequent Patterns
6:114	{([1:101, 5:113, 6:114],748384),
	([1:101, 11:8912626, 5:113, 6:114],230746)}
5:102	{([1:101, 5:102],1042692),
	([1:101, 11:8912626, 5:102],225428)}
5:176	{([1:101, 5:176],1695814),
	([1:101, 11:8912626, 5:176],1044079)}
6:109	{([1:101, 5:108, 6:109],2792865),
	([1:101, 5:108, 6:109, 11:8912626],166815)}

Fig. 3. Frequent patterns generated for the Geonames dataset. Each item is a pair of property and object $(p : o)$.

2.2 Association Rule Mining

Frequent itemset mining is often associated with *association rule mining*, which involves generating association rules from the frequent itemset with constraints of minimal confidence (to determine if a rule is interesting or not). However, in this study, we do not require mining association rules using confidence values. Instead, we split the given database into two disjoint databases, say A and B, based on the frequent patterns. Those transactions which contain one or more of the top N frequent patterns are inserted into dataset A while the other transactions are inserted into dataset B. Compression can be performed by creating a set of rules using top N frequent patterns and removing those triples from the dataset which can be inferred by applying rules to some other triples in the same dataset.

Multi-dimensional Association Rules. Although association mining was originally studied for mining transactions for only one attribute (ex:Product), much research has been performed to extend it across multiple attributes [16,17,28,29]. In this study, RDF datasets are viewed as multi-dimensional transaction databases by treating each property as an attribute and a subject as an identifier. Similar to intra-transaction and inter-transaction associations [17], we define intra-property and inter-property associations for RDF datasets. Intra-property association refers to an association among different object values for

[2] `http://downloads.dbpedia.org/preview.php?file=3.7_sl_en_sl_instance_types_en.nt.bz2`

a given property while inter-property association refers to association between multiple properties.

3 Rule Based Compression

In this section, we introduce two RB compression algorithms - one using intra-property transactions and the other using inter-property transactions. In addition, we provide an algorithm for *delta compression* to deal with incremental compression when a set of triples needs to be added to existing compressed graphs. Specifically, we investigate how to

- generate a set of decompression rules, R
- decompose the graph G to G_A and G_D, such that the requirements of RB compression holds true
- maximize the reduction in number of triples

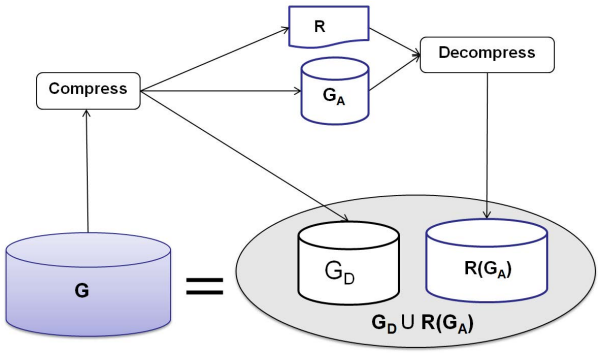


Fig. 4. Rule Based Compression, $G = G_D \cup R(G_A)$

Figure 4 depicts the high level overview of Rule Based Compression technique. We consider an RDF Graph G containing $|G|$ non-duplicate triples. Lossless compression on graph G can be obtained by splitting the given graph G into an *Active Graph,* G_A, and a *Dormant Graph,* G_D, such that: $G = R(G_A) \cup G_D$ where R represents the set of *decompression rules* to be applied to the active graph G_A during decompression. $R(G_A)$ is the graph resulting from this application.

Since the compression is lossless, we have $|G| = |R(G_A)| + |G_D|$.

Definition 2. *Let G be an RDF graph containing a set T of triples. An RB compression is a 3-tuple (G_A, G_D, R), where $G_D \subset G$ is a dormant graph containing some triples $T_D \subset T$, G_A is an active graph containing $T_A \subset T - T_D$ triples and R is a set of decompression rules that is applied to G_A (denoted by $R(G_A)$) producing a graph containing exactly the set $T - T_D$ of triples.*

G_D is referred to as dormant since it remains unchanged during decompression (no rule can be applied to it during decompression).

3.1 Intra-property RB Compression

Algorithm 1 follows a divide and conquer approach. For each property in a graph G, we create a new dataset and mine frequent patterns on this dataset. Transactions are created per subject within this dataset. Each transaction is a list of objects corresponding to a subject as shown in Figure 1. Using frequent patterns, a set of rules is generated for each property and later aggregated. Each rule contains a property p, an object item k, and a frequent pattern itemset v associated with k. This rule will be used to expand compressed data given in G_A as follows:

$$\forall x.triple(x,p,k) \rightarrow \bigwedge_{i=1}^{n} triple(x,p,v_i) \qquad \text{where, } v = v_1, v_2, ..., v_n$$

Algorithm 1. Intra-property RB compression

Require: G

$R \leftarrow \phi$, $G_D \leftarrow \phi$, $G_A \leftarrow \phi$
for each property, p that occurs in G **do**
create a transaction database D from a set of intra-property transactions. Each transaction (s,t) contains a subject s as identifier and t a set of corresponding objects.
generate $\{(k, F_k)\}$ set of frequent patterns
  for all (k, F_k) **do**
    select v_k such that
      $\sigma(v_k) = argmax_v\{\sigma(v) | v \text{ occurs in } F_k, |v| > 1\}$
    $R \leftarrow R \cup (k \rightarrow v_k)$ ▷ add a new rule
  end for
  for each $(s,t) \in D$ **do**
    for each $(k \rightarrow v_k) \in R$ **do**
      if $t \cap v_k = v_k$ **then**
        $G_A \leftarrow G_A \cup (s,p,k)$ ▷ add single triple
        $t \leftarrow t - v_k$
      end if
    end for
    for each $o \in t$ **do**
      $G_D \leftarrow G_D \cup (s,p,o)$
    end for
  end for
end for

For illustration, here's one such decompression rule we obtained during an experiment on DBpedia dataset:

$\forall x$.triple(x, rdf:type, foaf:Person) $\rightarrow$
triple(x, rdf:type, schema:Person)
$\wedge$ triple(x, rdf:type, dbp:Person)
$\wedge$ triple(x, rdf:type, owl:Thing)

This triple is attached to the active graph G_A so that all triples that can be inferred from it are removed. Other triples which cannot be inferred, are placed

in dormant graph G_D. The process is repeated for all properties, appending results to already existing rules R, active graph G_A and dormant graph G_D.

3.2 Inter-property RB Compression

In Algorithm 2, we mine frequent patterns across different properties. Transactions used in this algorithm are created by generating a list of all possible pairs of properties and objects for each subject. Thus, each item of a transaction is a pair $(p : o)$. We follow similar approach as before for generating frequent patterns and rules. Each rule contains a key pair (p_k, o_k) and a corresponding frequent pattern v as a list of items $(p : o)$.

Algorithm 2. Inter-property RB compression

Require: G

$R \leftarrow \phi$, $G_D \leftarrow \phi$, $G_A \leftarrow \phi$
create a transaction database D from a set of inter-property transactions. Each transaction, (s, t) contains a subject s as identifier and t a set of (p, o) items.
generate $\{(k, F_k)\}$ set of frequent patterns
for all (k, F_k) **do**
    select v_k such that
        $\sigma(v_k) = \{argmax_v \sigma(v) | v \text{ occurs in } F_k, |v| > 1\}$
    $R \leftarrow R \cup (k \rightarrow v_k)$ ▷ add a new rule
end for
for each $(s, t) \in D$ **do**
    for each $(k \rightarrow v_k) \in R$ **do**
        if $t \cap v_k = v_k$ **then**
            $G_A \leftarrow G_A \cup (s, p_k, o_k)$ ▷ add single triple
            $t \leftarrow t - v_k$
        end if
    end for
    for each $(p, o) \in t$ **do**
        $G_D \leftarrow G_D \cup (s, p, o))$
    end for
end for

The procedure is similar to one described in 3.1 once frequent patterns and rules are generated.

$$\forall x.triple(x, p_k, o_k) \rightarrow \bigwedge_{i=1}^{n} triple(x, p_i, o_i)$$

For illustration, here's one such decompression rule we obtained during an experiment on Geonames dataset:

$\forall x$.triple(x, geo:featureCode, geo:V.FRST) $\rightarrow$
triple(x, rdf:type, geo:Feature)
$\wedge$ triple(x, geo:featureClass, geo:V)

3.3 Optimal Frequent Patterns

In this section, we describe optimal rule generation strategy for achieving better compression. In Algorithm 1 and Algorithm 2, we generate frequent patterns and keep only one frequent pattern v per k. By selecting only one frequent pattern per item, it's guaranteed that no circular reference or recursion occurs during decompression. As such, for any given triple in a compressed graph, only one rule can be applied.

The choice of v for k is determined based on whether v has the maximum support. In this section, we present our findings for optimal v pattern selection based on both support value and itemset length. To illustrate this finding, please consider a sample FP-Growth output obtained by mining one of the datasets as shown in Figure 2(a) in section 2.1. If we look at frequent pattern sets for $k = 70$, we have:

1. $(v_1, \sigma_1) = ([22, 70], 56372)$
2. $(v_2, \sigma_2) = ([22, 103, 26, 304, 173, 70], 31084)$
3. $(v_3, \sigma_3) = ([22, 202, 42, 70], 25288)$

The following rule can be applied to select the optimal frequent pattern: select the pattern v_i that maximizes $(|v_i| - 1) \times \sigma_i)$. We call $(|v_i| - 1) \times \sigma_i)$, denoted by $\rho(v_i)$, the *Redundant Triple Density*, signifying the total number of triples that can be removed by using a rule: $(k \rightarrow v_k)$. It is apparent that selecting v_2 during rule generation leads to higher compression than selecting v_1 or v_3.

We call $(|v_i|) \times \sigma_i)$ the *Triple Density* signifying the total number of triples that are associated with this rule.

3.4 Delta Compression

One of the important properties of RB compression is that incremental compression can be achieved on the fly without much computation. Let's say, we consider an RDF graph G, which has undergone RB-Compression resulting in G_A active graph, G_D dormant graph and a set R of decompression rules. If a new set of triples corresponding to a subject s, denoted by ΔT_s, needs to be added to graph G, delta compression can be achieved by using the results from the last compression. Each delta compression updates the existing active and dormant graphs. Hence, there is no need for full RB-Compression every time a set of triples is added.

Algorithm 3 provides a delta compression algorithm when ΔT_s needs to be added. The algorithm can be extended to include a set of subjects, S. It should be noted that we do not create new rules for a new set of triples. As such, the compressed version might not be optimal. A full compression is recommended if a large number of new triples needs to be added or if large number of delta compression have already been performed.

If a triple needs to be removed, an extra check needs to be performed to see if the removal violates any existing rules. Such removal might require moving some of the inferred triples from the active graph to the dormant graph.

Algorithm 3. Delta Compression

Require: G_A, G_D, R , ΔT_s
1: Extract all triples, T_D, corresponding to s subject from G_D
2: $T \leftarrow T_D \cup \Delta T_s$
3: **for all** $t \in T$ **do**
4: **if** $R(t) \subseteq T$ **then**
5: $G_A \leftarrow G_A \cup t$ ▷ insert into active graph
6: $T \leftarrow T - R(t)$
7: **end if**
8: **end for**
9: **for all** $t \in T$ **do**
10: $G_D \leftarrow G_D \cup t$ ▷ insert into dormant graph
11: **end for**

4 Decompression

Decompression can be performed either sequentially or in parallel. Sequential decompression requires applying R decompression rules to triples in G_A active graph and merging these inferred triples with the triples in G_D dormant graph. Since each triple in a compressed graph can belong to at most one rule, it's complexity is $O(|R|.|G_A|)$. The number of rules is negligible compared to the number of triples in the active graph.

For parallel decompression, an active graph can be split into multiple smaller graphs so that each small dataset can perform decompression. This allows generation of inferred triples in parallel. Since rules are not ordered, inferred triples can be added to an uncompressed graph whenever they are generated. Finally, all triples of the dormant graph are merged into this uncompressed graph.

5 Experiments

This section shows experimental results of the compression performed by our system. Our experiment is conducted on several linked open datasets as well as synthetic benchmark datasets of varying sizes. The smallest dataset consists of 130K triples while the largest dataset consists of 119 million triples.

5.1 RB Compression - Triple Reduction

Table 1 shows a comparison between the outputs of the two algorithms we discussed in Section 3 for nine different linked open datasets. The compression ratio, r is defined as the ratio of the number of triples in compressed dataset to that in uncompressed dataset. It is evident from the results that compression based on inter-property frequent patterns is far better than compression using intra-property frequent patterns. Details including the number of predicates and transactions derived during experiments are also included in the table. It can be seen that the best RB compression (inter-property) can remove more than 50% of triples for the CN datasets and DBpedia rdftype dataset.

Table 1. Compression ratio for various linked open datasets

Dataset	triples (K)	predicate	transaction (K)	compression ratio	
				intra-property	inter-property
Dog Food	130	132	12	0.98	0.82
CN 2012	137	26	14	0.82	0.43
ArchiveHub	431	141	51	0.92	0.71
Jamendo	1047	25	336	0.99	0.82
LinkedMdb	6147	222	694	0.97	0.75
rdftypes	9237	1	9237	0.19	0.19
RDF About	17188	108	3132	0.97	0.84
DBLP	46597	27	2840	0.96	0.86
Geonames	119416	26	7711	0.97	0.71

5.2 RB Compression - Performance

In addition to the compression ratio, the following metrics are measured to evaluate the performance of the system: a) time it takes to perform RB compression and b) time it takes to perform full decompression. Figure 5 shows the

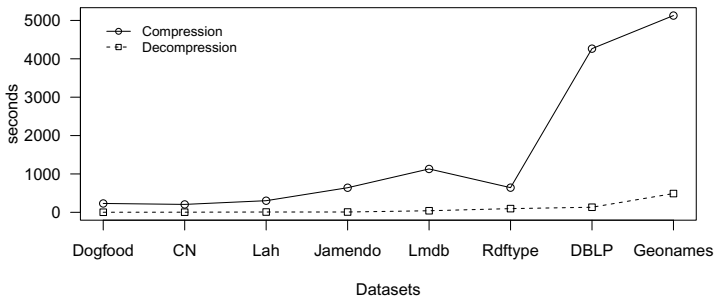


Fig. 5. Compression vs Decompression time for various linked open datasets

comparison between total time required for compression and the total time required for the full decompression. In general, RB compression time increases with the increase in triple size. However, if the total number of predicates in a dataset is very low, as in the case of DBpedia rdftypes dataset, compression time could be significantly lower. Decompression is faster by several order of magnitudes compared to the compression. This can be attributed to the fact that each triple is associated with a maximum of one rule and the number of rules are very few compared to the triple size. In addition, we apply rules only to triples in the Active Graph.

5.3 RB Compression on Benchmark Dataset

In this experiment, we ran RB Compression against one of the mainstream benchmark datasets, LUBM [8]. LUBM consists of a university domain ontology and provides a method for generating synthetic data of varying size.

Table 2 provides details on various LUBM datasets[3] we used for the experiment. Not surprisingly, these results show that compression time on dataset increases with the increase in dataset size. However, the compression ratio remained nearly constant for all the synthetic dataset. Decompression time proved to be far lesser than the time required for compression as seen in Figure 6. It took only 200 seconds for the decompression of the LUBM 1000 dataset compared to 11029 second for the compression.

Table 2. Compression ratio and time for various LUBM datasets

Dataset	**triples** (K)	**transaction** (K)	**compression** ratio	**Time** sec
LUBM 50	6654	1082	0.763	715
LUBM 100	13405	2179	0.757	1485
LUBM 200	26696	4341	0.757	2513
LUBM 500	66731	10847	0.757	6599
LUBM 1000	133573	21715	0.757	11029

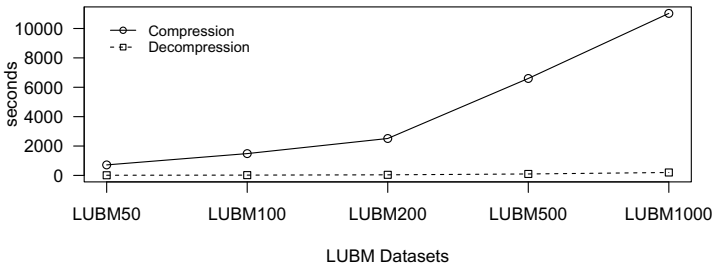


Fig. 6. Compression vs Decompression time for various LUBM datasets

5.4 Comparison Using Compressed Dataset Size

In addition to evaluating our system based on triple count, we examine the compression based on the storage size of the compressed datasets and compare it against other compression systems. This is important since none of the existing compression systems has the ability to compress RDF datasets by removing triples. [5] compared different universal compressors and found that bzip2[4] is one of the best universal compressors. For this study, we compress the input

[3] LUBM datasets created with index and seed set to 0.
[4] `http://bzip2.org`

dataset (in N-Triples format) and the resulting dataset using bzip2 and provide a quantitative comparison (see Table 3). An advantage of semantic compression such as RB Compression is that one can still apply syntactic compression (e.g. HDT) to the results. HDT [6] achieves a greater compression for most of the datasets we experimented on. Such high performance can be attributed to its ability to take advantage of the highly skewed RDF data. Since any generic RDF dataset can be converted to HDT compact form, we ran HDT on the compressed dataset resulting from RB Compression. The experimental results are shown in Table 3. We see that this integration does not always lead to a better compression. This is due to the overhead of header and dictionary that HDT creates for both active and dormant dataset[5].

Table 3. Comparison of various compression techniques based on dataset size

Dataset	Size	compressed	**compressed size using bzip2**		
			HDT	inter-property	HDT + inter-
DogFood	23.4 MB	1.5 MB	1088 K	1492 K	1106 K
CN 2012	17.9 MB	488 K	164 K	296 K	144 K
Archive Hub	71.8 MB	2.5MB	1.8 MB	1.9 MB	1.7MB
Jamendo	143.9 MB	6 MB	4.4MB	5.6 MB	4.6 MB
LinkedMdb	850.3 MB	22 MB	16 MB	22.6 MB	14.5MB
DBpedia rdftypes	1.2 GB	45 MB	11 MB	17.9 MB	10.1 MB
DBLP	7.5 GB	265 MB	201 MB	239 MB	205 MB
Geonames	13 GB	410 MB	304 MB	380 MB	303 MB

6 Soundness and Completeness

Although it should already be rather clear from our definitions and algorithms that our compression is *lossless* in the sense that we can recover all erased triples by using the newly introduced rules—let us dwell on this point for a little while.

First of all, it is worth mentioning that we cannot only recreate all erased triples by exhaustive forward-application of the rules—a fact that we could reasonable refer to as *completeness* of our approach. Rather, our approach is also *sound* in the sense that *only* previously erased triples are created by application of the rules. I.e., our approach does *not* include an inductive component, but is rather restricted to *detecting patterns which are explicitly and exactly represented in the dataset*. Needless to say, the recreation of erased triples using a forward-chaining application of rules can be rephrased as using a deductive reasoning system as decompressor.

It is also worth noting that the rules which we introduce, which are essentially of the form $\text{triple}(x, p, k) \rightarrow \text{triple}(x, p, v)$, can also be expressed in the OWL [10]

[5] If both these graphs are merged and HDT is performed, the resulting size will be always lesser than that obtained when only HDT is used for compression.

Web ontology Language. Indeed, a triple such as (x, p, k) can be expressed in OWL, e.g., in the form[6] $k(x)$ if p is `rdf:type`, or in the form $p(x, k)$ if p is a newly introduced property. The rule above then becomes $k \sqsubseteq v$ for p being `rdf:type`, and it becomes $\exists p.\{k\} \sqsubseteq \exists p.\{v\}$ in the case of the second example.

The observation just made that our compression rules are expressible in OWL. From this perspective, our approach to lossless compression amounts to the creation of schema knowledge which is completely faithful (in the sound and complete sense) to the underlying data. I.e., it amounts to the introduction of *uncontroversial* schema knowledge to Linked Data sets. It is rather clear that this line of thinking opens up a plethora of exciting follow-up work, which we intend to pursue.

7 Related Work

To the best of our knowledge, this is the first work that investigates practical rule based logical compression of RDF datasets which removes triples to achieve compression. Most of the existing compression techniques focus on compact representation of RDF data as a means of compression. Turtle, a sub-language of N3, is one such compact and natural text representation for RDF data. [5] has explored various compression techniques for RDF datasets and observed that most RDF datasets are highly compressible due to it's power-law distribution in term-frequencies, schemas and resources. [6] introduced a more compact representation format, HDT, by decomposing an RDF data source into Header, Dictionary and Triples. A specific compressed version of HDT, HDT-compressed, outperforms most of the universal compressors [6]. [19,21] studied the problem of redundancy elimination on RDF graphs in the presence of rules, constraints and queries. [24] uses distributed dictionary encoding with MapReduce to compress large RDF datasets.

Work on frequent itemset mining [1,9,15,26,20,27] provides a foundation for our algorithms. [4] explored pattern mining based compression schemes for web graphs specifically designed to accomodate community queries. [25] used association rule mining techniques for generating ontology based on rdf:type statements.

8 Conclusion

In this paper, we have introduced a novel lossless compression technique called Rule Based Compression that efficiently compresses RDF datasets using logical rules. The key idea is to split the original dataset into two disjoint datasets A and B, such that dataset A adheres to certain logical rules while B does not. Dataset A can be compressed since we can prune those triples that can be inferred by applying rules on some other triples in the same dataset. We have provided two algorithms based on frequent pattern mining to demonstrate the compression capability of our rule based compression. Experimental results show

[6] We use description logic notation for convenience, see [11].

that in some datasets, RB Compression can remove more than half the triples without losing data integrity. This finding is promising and should be explored further for achieving better compression. In future work, we will investigate the use of RB Compression in instance alignment and automated schema generation.

Acknowledgments. This work was supported by the National Science Foundation under award 1143717 "III: EAGER – Expressive Scalable Querying over Linked Open Data" and 1017225 "III: Small: TROn – Tractable Reasoning with Ontologies."

References

1. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, SIGMOD 1993, pp. 207–216. ACM (1993)
2. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proceedings of the 20th International Conference on Very Large Data Bases, VLDB 1994, pp. 487–499. Morgan Kaufmann Publishers Inc. (1994)
3. Álvarez-García, S., Brisaboa, N.R., Fernández, J.D., Martínez-Prieto, M.A.: Compressed k2-triples for full-in-memory RDF engines. In: AMCIS (2011)
4. Buehrer, G., Chellapilla, K.: A scalable pattern mining approach to web graph compression with communities. In: Proceedings of the 2008 International Conference on Web Search and Data Mining, WSDM 2008, pp. 95–106. ACM (2008)
5. Fernández, J.D., Gutierrez, C., Martínez-Prieto, M.A.: RDF compression: Basic approaches. In: Proceedings of the 19th International Conference on World Wide Web, WWW 2010, pp. 1091–1092. ACM (2010)
6. Fernández, J.D., Martínez-Prieto, M.A., Gutierrez, C.: Compact representation of large RDF data sets for publishing and exchange. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010, Part I. LNCS, vol. 6496, pp. 193–208. Springer, Heidelberg (2010)
7. Goethals, B.: Survey on frequent pattern mining. Tech. rep. (2003)
8. Guo, Y., Pan, Z., Heflin, J.: Lubm: A benchmark for owl knowledge base systems. Journal of Web Semantics 3(2-3), 158–182 (2005)
9. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, SIGMOD 2000, pp. 1–12. ACM (2000)
10. Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P.F., Rudolph, S. (eds.): OWL 2 Web Ontology Language: Primer. W3C Recommendation (October 27, 2009), `http://www.w3.org/TR/owl2-primer/`
11. Hitzler, P., Krötzsch, M., Rudolph, S.: Foundations of Semantic Web Technologies. Chapman & Hall/CRC (2009)
12. Huang, J., Abadi, D.J., Ren, K.: Scalable SPARQL querying of large RDF graphs. PVLDB 4(11), 1123–1134 (2011)
13. Iannone, L., Palmisano, I., Redavid, D.: Optimizing RDF storage removing redundancies: An Algorithm. In: Ali, M., Esposito, F. (eds.) IEA/AIE 2005. LNCS (LNAI), vol. 3533, pp. 732–742. Springer, Heidelberg (2005)

14. Joshi, A.K., Hitzler, P., Dong, G.: Towards logical linked data compression. In: Proceedings of the Joint Workshop on Large and Heterogeneous Data and Quantitative Formalization in the Semantic Web, LHD+SemQuant 2012, at the 11th International Semantic Web Conference, ISWC 2012 (2012)
15. Li, H., Wang, Y., Zhang, D., Zhang, M., Chang, E.Y.: PFP: Parallel FP-Growth for query recommendation. In: Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys 2008, pp. 107–114. ACM (2008)
16. Li, Q., Feng, L., Wong, A.K.Y.: From intra-transaction to generalized intertransaction: Landscaping multidimensional contexts in association rule mining. Inf. Sci. 172(3-4), 361–395 (2005)
17. Lu, H., Feng, L., Han, J.: Beyond intratransaction association analysis: mining multidimensional intertransaction association rules. ACM Trans. Inf. Syst. 18(4), 423–454 (2000)
18. Manola, F., Miller, E., McBride, B.: RDF primer (2004), http://www.w3.org/TR/rdf-primer/
19. Meier, M.: Towards rule-based minimization of RDF graphs under constraints. In: Calvanese, D., Lausen, G. (eds.) RR 2008. LNCS, vol. 5341, pp. 89–103. Springer, Heidelberg (2008)
20. Özdogan, G.Ö., Abul, O.: Task-parallel FP-growth on cluster computers. In: Gelenbe, E., Lent, R., Sakellari, G., Sacan, A., Toroslu, H., Yazici, A. (eds.) Computer and Information Sciences. LNEE, vol. 62, pp. 383–388. Springer, Heidelberg (2010)
21. Pichler, R., Polleres, A., Skritek, S., Woltran, S.: Redundancy elimination on RDF graphs in the presence of rules, constraints, and queries. In: Hitzler, P., Lukasiewicz, T. (eds.) RR 2010. LNCS, vol. 6333, pp. 133–148. Springer, Heidelberg (2010)
22. Savasere, A., Omiecinski, E., Navathe, S.B.: An efficient algorithm for mining association rules in large databases. In: Proceedings of the 21st International Conference on Very Large Data Bases, VLDB 1995, pp. 432–444. Morgan Kaufmann Publishers Inc. (1995)
23. Srikant, R., Vu, Q., Agrawal, R.: Mining association rules with item constraints. In: KDD, pp. 67–73 (1997)
24. Urbani, J., Maassen, J., Drost, N., Seinstra, F.J., Bal, H.E.: Scalable RDF data compression with MapReduce. Concurrency and Computation: Practice and Experience 25(1), 24–39 (2013)
25. Völker, J., Niepert, M.: Statistical schema induction. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) ESWC 2011, Part I. LNCS, vol. 6643, pp. 124–138. Springer, Heidelberg (2011)
26. Zaïane, O.R., El-Hajj, M., Lu, P.: Fast parallel association rule mining without candidacy generation. In: Proceedings of the 2001 IEEE International Conference on Data Mining, ICDM 2001, pp. 665–668. IEEE Computer Society (2001)
27. Zaki, M.J., Parthasarathy, S., Ogihara, M., Li, W.: New algorithms for fast discovery of association rules. In: KDD, pp. 283–286 (1997)
28. Zhang, H., Zhang, B.: Generalized association rule mining algorithms based on multidimensional data. In: Xu, L.D., Min Tjoa, A., Chaudhry, S.S. (eds.) CONFENIS 2007. IFIP, vol. 254, pp. 337–342. Springer, Boston (2007)
29. Zhou, A., Zhou, S., Jin, W., Tian, Z.: Generalized multidimensional association rules. J. Comput. Sci. Technol. 15(4), 388–392 (2000)

Access Control for HTTP Operations on Linked Data

Luca Costabello[1,⋆], Serena Villata[1,⋆⋆,⋆],
Oscar Rodriguez Rocha[2,⋆], and Fabien Gandon[1,⋆]

[1] INRIA Sophia Antipolis, France
firstname.lastname@inria.fr
[2] Politecnico di Torino, Italy
oscar.rodriguezrocha@polito.it

Abstract. Access control is a recognized open issue when interacting with RDF using HTTP methods. In literature, authentication and authorization mechanisms either introduce undesired complexity such as SPARQL and ad-hoc policy languages, or rely on basic access control lists, thus resulting in limited policy expressiveness. In this paper we show how the Shi3ld attribute-based authorization framework for SPARQL endpoints has been progressively converted to protect HTTP operations on RDF. We proceed by steps: we start by supporting the SPARQL 1.1 Graph Store Protocol, and we shift towards a SPARQL-less solution for the Linked Data Platform. We demonstrate that the resulting authorization framework provides the same functionalities of its SPARQL-based counterpart, including the adoption of Semantic Web languages only.

1 Introduction

In scenarios such as Linked Enterprise Data, access control becomes crucial, as not all triples are openly published on the Web. Solutions proposed in literature protect either SPARQL endpoints or generic RDF documents and adopt Role-based (RBAC) [20] or Attribute-based (ABAC) [18] models. The Semantic Web community is recently emphasizing the need for a substantially *"Web-like"* interaction paradigm with Linked Data. For instance, the W3C Linked Data Platform[1] initiative promotes the use of read/write HTTP operations on triples, thus providing a *basic profile* for Linked Data servers and clients. Another example is the SPARQL 1.1 Graph Store Protocol[2], a set of guidelines to interact with RDF graphs with HTTP operations. Defining an access control model for these scenarios is still an open issue[3]. Frameworks targeting HTTP access to

⋆ We are grateful to Olivier Corby for helpful comments and suggestions.
⋆⋆ The second author acknowledges support of the DataLift Project ANR-10-CORD-09 founded by the French National Research Agency.

[1] http://www.w3.org/TR/ldp/
[2] http://www.w3.org/TR/sparql11-http-rdf-update/
[3] http://www.w3.org/2012/ldp/wiki/AccessControl

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 185–199, 2013.

RDF resources rely on access control lists, thus offering limited policy expressiveness [13,14,17,19], e.g., no location-based authorization. On the other hand, existing access control frameworks for SPARQL endpoints [1,4,10] add complexity rooted in the query language and in the SPARQL protocol, and they often introduce ad-hoc policy languages, thus requiring adaptation to the HTTP-only scenario.

In this paper, we answer the research question: *How to design an authorization framework for HTTP-based interaction with Linked Data?* This research question breaks down into the following sub-questions: (i) how to define an authorization model featuring expressive policies based on standard Web languages only, and (ii) how to adopt this model in HTTP-based interaction with Linked Data scenarios like the Graph Store Protocol (GSP) and the Linked Data Platform (LDP).

We adapt the Shi3ld authorization framework for SPARQL [4] to a SPARQL-less scenario. We choose Shi3ld because its features satisfy our requirements: (i) it adopts attribute-based access policies ensuring expressiveness, and (ii) it exclusively uses Semantic Web languages for policy definition and enforcement.

We illustrate Shi3ld-GSP, an intermediate version designed for the SPARQL 1.1 Graph Store HTTP Protocol. We progressively shift to the Linked Data Platform context, a scenario where SPARQL is no longer present. We have developed two solutions for this scenario: (i) an authorization module embedding a hidden SPARQL engine, and (ii) a framework where we completely get rid of SPARQL. In the latter case, the Shi3ld framework adopts a SPARQL-less subgraph matcher which grants access if client attributes correspond to the declared policy graphs. For each framework, we evaluate the response time and we show how the authorization procedure impacts on HTTP operations on RDF data.

The key features of our attribute-based authorization framework for HTTP-based interaction with Linked Data are (i) the use of Web languages only, i.e., HTTP methods and RDF, without ad-hoc languages for policies definition, (ii) the adoption of attribute-based access conditions enabling highly expressive policies, and (iii) the adaptation to the GSP and LDP scenarios as a result of a progressive disengagement from SPARQL. Moreover, Shi3ld is compatible and complementary with the WebID authentication framework[4].

In this paper, we focus on *authorization* only, without addressing the issues related to authentication and identity on the Web. Although we discuss state-of-the-art anti-spoofing techniques for attribute data, the present work does not directly address the issue.

The remainder of this paper is organized as follows. Section 2 summarizes the related work, and highlights the requirements of an authorization model for our scenario. Section 3 describes the main insights of Shi3ld, and presents the three proposed solutions to adapt the framework to HTTP operations on RDF. An experimental evaluation of response time overhead is provided in Section 4.

[4] `http://www.w3.org/2005/Incubator/webid/spec/`

2 Related Work

Many access control frameworks rely on *access control lists* (ACLs) that define which users can access the data. This is the case of the Web Access Control vocabulary (WAC)[5], that grants access to a whole RDF document. Hollenbach et al. [13] present a system where providers control access to RDF documents using WAC. In our work, we look for more expressive policies that can be obtained without leading to an increased complexity of the adopted language or model.

Similarly to ACLs, other approaches specify *who* can access the data, e.g., to which roles access is granted. Among others, Giunchiglia et al. [12] propose a Relation Based Access Control model (*RelBAC*) based on description logic, and Finin et al. [9] study the relationship between OWL and RBAC [20]. They briefly discuss possible ways of going beyond RBAC such as Attribute Based Access Control, a model that grants access according to client attributes, instead of relying on access control lists.

The ABAC model is adopted in the Privacy Preference Ontology (PPO)[6] [19], built on top of WAC, where consumers require access to a given RDF file, e.g., a FOAF profile, and the framework selects the part of the file the consumer can access, returning it. In our work, we go beyond the preference specification based on FOAF profiles. Shi3ld [4] adopts ABAC for protecting the accesses to SPARQL endpoints using Semantic Web languages only.

Other frameworks introduce a *high level syntax* for expressing policies. Abel et al. [1] present a context-dependent access control system for RDF stores, where policies are expressed using an ad-hoc syntax and mapped to existing policy languages. Flouris et al. [10] present an access control framework on top of RDF repositories using a high level specification language to be translated into a SPARQL/SerQL/SQL query to enforce the policy. Muhleisen et al. [17] present a policy-enabled server for Linked Data called PsSF, where policies are expressed using a descriptive language based on SWRL[7]. Shen and Cheng [21] propose a Context Based Access Control model (SCBAC) where policies are expressed using SWRL. Based on the Oracle Relational database, the Oracle triple store protects RDF granting or revoking operations on database views. If tighter security is requested, triple-level access control can be enforced by relying on Oracle Label Security or Oracle Virtual Private Database[8].

Access control models may consider not only the information about the consumer who is accessing the data, but also the *context* of the request, e.g., time, location. Covington et al. [5] present an approach where the notion of role proposed in RBAC is used to capture the environment in which the access requests are made. Cuppens and Cuppens-Boulahia [6] propose an Organization Based Access Control model (OrBAC) that contains contextual conditions. Toninelli et al. [22] use context-awareness to control access to resources, and semantic

[5] `http://www.w3.org/wiki/WebAccessControl`
[6] `http://vocab.deri.ie/ppo`
[7] `http://www.w3.org/Submission/SWRL/`
[8] `http://bit.ly/oracle-RDF-access-control`

technologies for policy specification. Corradi et al. [3] present UbiCOSM, a security middleware adopting context as a basic concept for policy specification and enforcement.

Table 1 summarizes the main characteristics of the related work described above[9]: application in the Web scenario, adopted AC model, policy language, protection granularity, permission model, context-awareness, conflict verification among policies, response time evaluation. None of the presented approaches satisfies all the features that we require for protecting HTTP operations on Linked Data, i.e., absence of ad-hoc policy languages, CRUD (Create, Read, Update, Delete) permission model, protection granularity at resource-level, and expressive access control model to go beyond basic access control lists.

Table 1. A summarizing comparison of the related work

	Web-based	AC model	Policy language	Protection granularity	Permissio model	Context Awareness	Conflict verificatio	Eval.
WAC[5]	YES	RBAC	RDF	RDF document	R/W	N/A	N/A	N/A
Abel et al. [1]	YES	ABAC	Custom	triples	R	YES	N/A	YES
Finin et al. [9]	YES	RBAC	OWL/RDF	resources	N/A	N/A	N/A	N/A
RelBAC [12]	YES	relation	DL	resources	N/A	N/A	N/A	N/A
Hollenbach[13]	YES	RBAC	RDF	RDF document	R/W	N/A	N/A	YES
Flouris et al. [10]	YES	RBAC	Custom	triples	R	N/A	YES	YES
PeLDS [17]	YES	RBAC	SWRL	RDF document	R/W	N/A	N/A	YES
PPO [19]	YES	ABAC	RDF, SPARQL	RDF doc(part)	R/W	N/A	N/A	N/A
SCBAC [21]	YES	context	SWRL	resources	N/A	YES	YES	N/A
Shi3ld-SPARQL[4]	YES	ABAC	RDF, SPARQL	named graphs	CRUD	YES	N/A	YES
Covington [5]	NO	RBAC	Custom	resources	R/W	YES	YES	N/A
CSAC [14]	NO	gen. RBAC	XML	resources	R	YES	N/A	N/A
Proteus[22]	NO	context	DL	Resources	N/A	YES	YES	YES
OrBAC [6]	NO	organizatio	Datalog	resources	R/W	YES	YES	N/A
UbiCOSM [3]	NO	context	RDF	resources	N/A	YES	YES	YES

3 Restricting HTTP Operations on Linked Data

Before discussing how we modified the Shi3ld original proposition [4] to obtain a SPARQL-less access control framework for HTTP operations on Linked Data, we provide an overview of the original Shi3ld authorization model for SPARQL endpoints. Shi3ld [4] presents the following key features:

Attribute-Based Paradigm. Shi3ld is an *attribute-based* authorization framework, i.e., authorization check is performed against a set of attributes sent by the client with the query that targets the resource. Relying on attributes provides broad access policy expressiveness, beyond access control lists. That means, among all, creating location-based and temporal-based access policies.

Semantic Web Languages Only. Shi3ld uses access policies defined with Semantic Web languages only, and no additional policy language needs to be defined. In particular, the access conditions specified in the policies are SPARQL ASK queries.

CRUD Permission Model. Access policies are associated to specific permissions over the protected resource. It is therefore possible to specify rules satisfied only when the access is in *create*, *read*, *update* and *delete* mode.

[9] We use N/A when the feature is not considered in the work.

Granularity. The proposed degree of granularity is represented by named graphs, allowing protection from triples up to whole dataset.

The HTTP-based interaction with Linked Data requires some major modifications to the above features. Although we keep the attribute-based paradigm and the CRUD permission model, the new versions of Shi3ld satisfy also the following requirements:

Protection of HTTP Access to Resources. Protected resources are retrieved and modified by clients using HTTP methods only, without SPARQL querying[10].

RDF-Only Policies. In the SPARQL-less scenario, access conditions are RDF triples with no embedded SPARQL.

Granularity. The atomic element protected by Shi3ld is a *resource*.

In this paper, we rely on the definition of resource provided by the W3C Linked Data Platform Working Group: LDP resources are HTTP resources queried, created, modified and deleted via HTTP requests processed by LDP servers[11]. Linked Data server administrators adopting Shi3ld must define a number of *access policies* and associate them to protected resources. Access policies and their components are formally defined as follows:

Definition 1. *(Access Policy) An Access Policy (P) is a tuple of the form* $P = \langle ACS, AP, R \rangle$ *where (i)* ACS *is a set of Access Conditions to satisfy, (ii)* AP *is an Access Privilege, and (iii)* R *is the resource protected by* P*.*

Definition 2. *(Access Condition) An Access Condition (AC) is a set of attributes that need to be satisfied to interact with a resource.*

Definition 3. *(Access Privilege) An Access Privilege (AP) is the set of allowed operations on the protected resource,* $AP = \{Create, Read, Update, Delete\}$*.*

The lightweight vocabularies used by Shi3ld are `s4ac`[12] for defining the policy structure, and `prissma`[13] for the client attributes[14]. Client attributes include user profile information, device features, environment data, or any given combination of these dimensions, in compliance with the widely-accepted definition by Dey [7] and the work by Fonseca et al.[15]. We delegate refinements and extensions to domain specialists, in the light of the Web of Data philosophy.

[10] This is in compliance with the LDP specifications.

[11] An LDP server is an "*application program that accepts connections in order to service requests by sending back responses*" as specified by HTTP 1.1 definition.

[12] `http://ns.inria.fr/s4ac`

[13] `http://ns.inria.fr/prissma`

[14] Although this vocabulary provides classes and properties to model context-aware attributes, it is not meant to deliver yet another contextual model: instead, well-known Web of Data vocabularies and recent W3C recommendations are reused. For more details, see Costabello et al. [4].

[15] `http://www.w3.org/2005/Incubator/model-based-ui/XGR-mbui/`

The main classes and properties of these vocabularies are visualized in Figure 1. Shi3ld offers a double notation for defining access conditions: with embedded SPARQL (Figure 2a) for SPARQL-equipped scenarios, and in full RDF (Figure 2b), adopted in SPARQL-less environments.

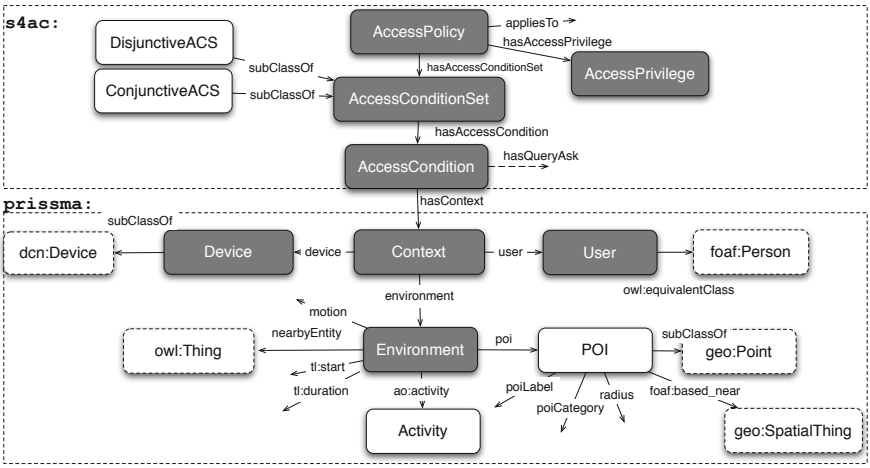


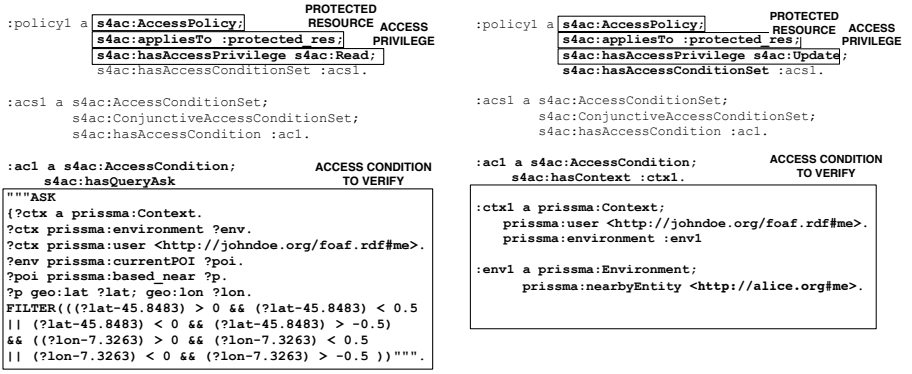
Fig. 1. Interplay of `s4ac` and `prissma` vocabularies for Shi3ld access policies

Figure 2 presents two sample access policies, expressed with and without SPARQL. The policy visualized in Figure 2a allows *read-only* access to the protected resource exclusively by a specific user and from a given location. The policy in Figure 2b authorizes the *update* of the resource by the given user, only if he is currently near Alice.

Whenever an HTTP query is performed on a resource, Shi3ld runs the authorization algorithm to check if the policies that protect the resource are satisfied or not. The procedure verifies the matching between the client attributes sent with the query and the access policies that protect the resource.

Shi3ld deals with authorization only. Nevertheless, authentication issues cannot be ignored as the trustworthiness of client attributes is critical for a reliable access control framework. Shi3ld supports heterogeneous authentication strategies, since the attributes attached to each client request include heterogeneous data, ranging from user identity to environment information fetched by device sensors (e.g. location). The trustworthiness of user identity is achieved thanks to the WebID[4] compatibility: in Shi3ld, user-related attributes are modelled with the `foaf` vocabulary[16], thus easing the adoption of WebID. Authenticating the attributes fetched by client sensors is crucial to prevent tampering. Hulsebosch et al. [14] provide a survey of verification techniques, such as heuristics relying on location history and collaborative authenticity checks. A promising approach is

[16] `http://xmlns.com/foaf/spec/`



(a) SPARQL-based (b) SPARQL-less

Fig. 2. Shi3ld access policies, expressed with and without SPARQL

mentioned in Kulkarni and Tripathi [16], where client sensors are authenticated beforehand by a trusted party. To date, no tamper-proof strategy is implemented in Shi3ld, and this is left for future work.

Moreover, sensible data, such as current location must be handled with a privacy-preserving mechanism. Recent surveys describe strategies to introduce privacy mainly in location-based services [8,15]. Shi3ld adopts an *anonymity-based* solution [8] and delegates attribute anonymisation to the client side, thus sensitive information is not disclosed to the server. We rely on partially encrypted RDF graphs, as proposed by Giereth [11]. Before building the RDF attribute graph and sending it to the Shi3ld-protected repository, a partial RDF encryption is performed, producing RDF-compliant results, i.e., the encrypted graph is still RDF (we use SHA-1 cryptographic hash function to encrypt RDF literals). On server-side, every time a new policy is added to the system, the same operation is performed on the attributes included in access policies. As long as literals included in access conditions are hashed with the same function used on the client side, the Shi3ld authorization procedure still holds[17].

We now describe the steps leading to a SPARQL-less authorization framework for HTTP operations on Linked Data. Our first proposal is a Shi3ld authorization framework for the SPARQL 1.1 Graph Store Protocol (Section 3.1). In Sections 3.2 and 3.3 we describe two scenarios tailored to the Linked Data Platform specifications, the second being completely SPARQL-less. Our work is grounded on the analogies between SPARQL 1.1 functions and the HTTP protocol semantics, as suggested by the SPARQL Graph Store Protocol specification[2].

[17] The adopted technique does not guarantee full anonymity [15]. Nevertheless, the problem is mitigated by the short persistence of client-related data inside Shi3ld cache: client attributes are deleted after each authorization evaluation. Encryption is not applied to location coordinates and timestamps, as this operation prevents geo-temporal filtering.

3.1 Shi3ld for SPARQL Graph Store Protocol

The SPARQL 1.1 HTTP Graph Store Protocol[2] provides an alternative interface to access RDF stored in SPARQL-equipped triple stores. The recommendation describes a mapping between HTTP methods and SPARQL queries, thus enabling HTTP operations on triples. The Graph Store Protocol can be considered as an intermediate step towards an HTTP-only access to RDF datastores, since it still needs a SPARQL endpoint.

Figure 3a shows the architecture of the authorization procedure of Shi3ld for GSP-compliant SPARQL endpoints (Shi3ld-GSP). Shi3ld-GSP acts as a module protecting a stand-alone SPARQL 1.1 endpoint, equipped with a Graph Store Protocol module. First, the client performs an HTTP operation on a resource. This means that an RDF attribute graph is built on the client, serialized and sent with the request in the HTTP `Authorization` header[18]. Attributes are saved into the triple store with a SPARQL 1.1 query. Second, Shi3ld selects the access policies that protect the resource. The access conditions (SPARQL `ASK` queries, as in Figure 2a) included in the policies are then executed against the client attribute graph. Finally, the results are logically combined according to the type of access condition set (disjunctive or conjunctive) defined by each policy. If the result returns *true*, the HTTP query is forwarded to the GSP SPARQL engine, which in turns translates it into a SPARQL query. If the access is not granted, a HTTP 401 message is delivered to the client.

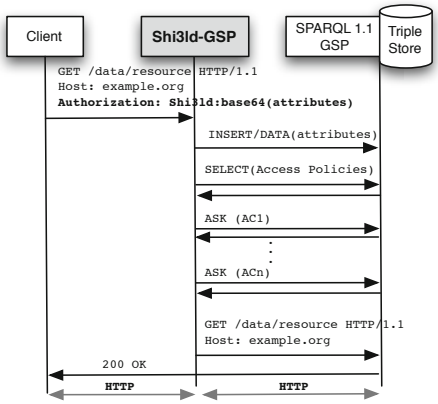
3.2 Shi3ld-LDP with Internal SPARQL Engine

The Linked Data Platform initiative proposes a simplified configuration for Linked Data servers and Web-like interaction with RDF resources. Compared to the GSP case, authorization frameworks in this scenario must deal with a certain number of changes, notably the absence of SPARQL and potentially the lack of a graph store.

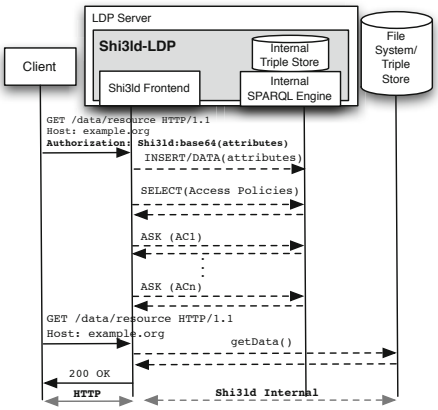
We adapt Shi3ld to work under these restrictions (Shi3ld-LDP). The framework architecture is shown in Figure 3b. Shi3ld-LDP protects HTTP operations, but it does not communicate with an external SPARQL endpoint, i.e. there are no intermediaries between the RDF repository (the filesystem or a triple store) and Shi3ld. To re-use the authorization procedure previously described, we integrate an internal SPARQL engine into Shi3ld, along with an internal triple store. Although SPARQL is still present, this is perfectly legitimate in a Linked Data Platform scenario, since the use of the query language is limited to Shi3ld internals and is not exposed to the outside world[19]. Despite the architectural changes, the Shi3ld model remains unchanged. Few modifications occur to the authorization procedure as described in Figure 3a: clients send HTTP requests to the desired resource. HTTP headers contain the attribute graph, serialized as previously described in Section 3.1. Instead of relying on an external SPARQL

[18] We extend the header with the ad-hoc `Shi3ld` option. Other well-known proposals on the web re-use this field, e.g. the OAuth authorization protocol.

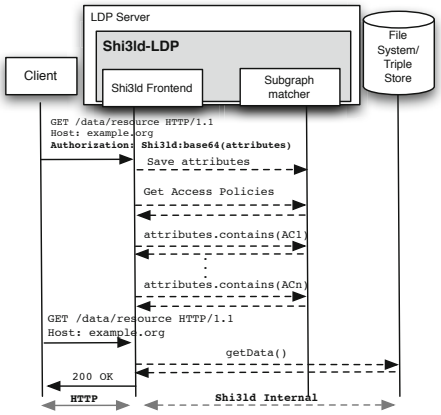
[19] SPARQL is still visible in access policies (Figure 2a).



(a) Shi3ld-GSP



(b) Shil3d-LDP (internal SPARQL engine)



(c) Shi3ld-LDP (SPARQL-less)

Fig. 3. Shi3ld Configurations

endpoint, attributes are now saved internally, using an `INSERT DATA` query. The access policies selection and the access conditions execution remain substantially unchanged, but the whole process is transparent to the platform administrator, as the target SPARQL endpoint is embedded in Shi3ld.

3.3 SPARQL-Less Shi3ld-LDP

To fulfill the Linked Data Platform recommendations, thus achieving a full-fledged *basic profile* for authorization frameworks, we drop SPARQL from the Shi3ld-LDP framework described in Section 3.2. Ditching SPARQL allows RDF-only access policies definition, and a leaner authorization procedure. To obtain a SPARQL-less framework, we re-use the access policy model and the logical steps of the previously described authorization procedure, although conveniently

adapted (Figure 3c). First, Shi3ld-LDP policies adopt RDF only, as shown in Figure 2b: attribute conditions previously expressed with SPARQL ASK queries (Figure 2a) are expressed now as RDF graphs. Second, the embedded SPARQL engine used in Section 3.2 has been replaced: its task was testing whether client attributes verify the conditions defined in each access policy. This operation boils down to a *subgraph matching problem*. In other words, we must check if the access conditions (expressed in RDF) are contained into the attribute graph sent with the HTTP client query. Such subgraph matching procedure can be performed without introducing SPARQL in the loop. To steer clear of SPARQL, without re-inventing yet another subgraph matching procedure, we scrap the SPARQL interpreter from the SPARQL engine [2] used in Section 3.2, keeping only the underlying subgraph matching algorithm[20].

To understand the SPARQL-less policy verification procedure and the complexity hidden by the SPARQL layer, we now provide a comprehensive description of the adopted subgraph matching algorithm, along with an overview of the RDF indexes used by the procedure. The algorithm checks whether a query graph Q (the access condition) is contained in the reference graph R (the client attributes sent with the query).

The reference graph R is stored in two key-value indexes (see example in Figure 4): index I_s stores the associations between property types and property subjects, and index I_o stores the associations between property types and property objects. Each RDF property type of R is therefore associated to a list of property subjects S_p and a list of property objects O_p. S_p contains URIs or blank nodes, O_p contains URIs, typed literals and blank nodes. Blank nodes are represented as anonymous elements, and their IDs are ignored.

The query graph Q, i.e., the access condition attributes, is serialized in a list L of subject-property-object elements $\{s_i, p_i, o_i\}$[21]. Blank nodes are added to the serialization as anonymous s_i or o_i elements.

The matching algorithm works as follows: for each subject-property-object $\{s_i, p_i, o_i\}$ in L, it looks up the indexes I_s and I_o using p_i as key. It then retrieves the list of property subjects S_p and the list of property objects O_p associated to p_i. Then, it searches for a subject in S_p matching with s_i, and an object in O_p matching with o_i. If both matches are found, $\{s_i, p_i, o_i\}$ is matched and the procedure moves to the next elements in L. If no match is found in either I_s or I_o, the procedure stops. Subgraph matching is successful if all L items are matched in the R index. Blank nodes act as wildcards: if a blank node is found in $\{s_i, p_i, o_i\}$ as object o_i or subject s_i, and O_p or S_p contains one or more blank nodes, the algorithm continues the matching procedure recursively, backtracking in case of mismatch and therefore testing all possible matchings. The example in Figure 4 shows a matching step of the algorithm, i.e., the successful matching of the triple "`_:b2 p:nearbyEntity http://alice.org/me`" against the client

[20] Third-party SPARQL-less Shi3ld-LDP implementations might adopt other off-the-shelf subgraph matching algorithms.

[21] A preliminary step replaces the query graph Q intermediate nodes into blank nodes. Blank nodes substitute SPARQL variables in the matching procedure.

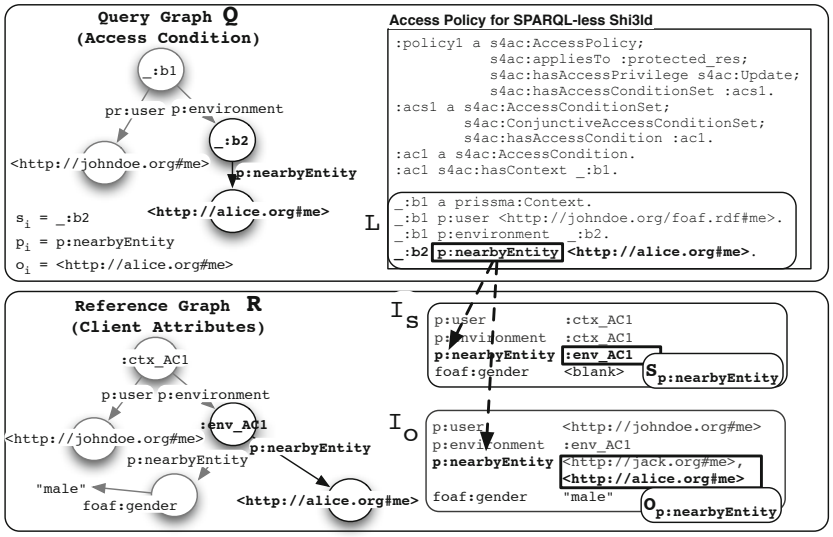


Fig. 4. Example of subgraph matching used in the SPARQL-less Shi3ld-LDP

attributes indexes I_s and I_o. The highlighted triple is successfully matched against the client attributes R.

Note that policies might contain location and temporal constraints: Shi3ld-GSP (Section 3.1) and Shi3ld-LDP with internal SPARQL endpoint (Section 3.2) handle these conditions by translating RDF attributes into SPARQL `FILTER` clauses. The subgraph matching algorithm adopted by SPARQL-less Shi3ld-LDP does not support geo-temporal authorization evaluation yet.

The three Shi3ld configurations described in this Section use the `Authorization` header to send client attributes. Even if there is no limit to the size of each header value, it is good practice to limit the size of HTTP requests, to minimize latency. Ideally, HTTP requests should not exceed the size of a TCP packet (1500 bytes), but in real world finding requests that exceed 2KB is not uncommon, as a consequence of cookies, browser-set fields and URL with long query strings[22]. To keep size as small as possible, before base-64 encoding, client attributes are serialized in turtle (less verbose that N-triples and RDF/XML). We plan to test the effectiveness of common lossless compression techniques to reduce the size of client attributes as future work. Furthermore, instead of sending the complete attribute graph in all requests, a server-side caching mechanism would enable the transmission of attribute graph deltas (i.e. only newly updated attributes will be sent to the server). Sending differences of RDF graphs is an open research topic[23], and it is out of the scope of this paper.

[22] `https://developers.google.com/speed/docs/best-practices/request`
[23] `http://www.w3.org/2001/sw/wiki/How_to_diff_RDF`

4 Evaluation

We implemented the three scenarios presented in Section 3 as Java standalone web services[24]. The Shi3ld-GSP prototype works with the Fuseki GSP-compliant SPARQL endpoint[25]. The Shi3ld-LDP prototype with internal SPARQL endpoint embeds the KGRAM/Corese[26] engine [2]. Our test campaign assesses the impact of Shi3ld on HTTP query response time[24]. We evaluate the prototypes on an Intel Xeon E5540, Quad Core 2.53 GHz machine with 48GB of memory. In our test configuration, Shi3ld-GSP protects a Fuseki SPARQL server, while Shil3d-LDP scenarios secure RDF resources saved on the filesystem. First, we investigate the relationship between response time and the number of access conditions to verify. Second, we test how access conditions complexity impacts on response time. Our third test studies the response time with regard to different HTTP methods. We execute five independent runs of a test query batch consisting in 50 HTTP operations (tests are preceded by a warmup run). Each query contains client attributes serialized in turtle (20 triples). The base-64 turtle serialization of the client attributes used in tests[24] is 1855 bytes long (including prefixes). Tests do not consider client-side literal anonymization (Section 3).

Our first test shows the impact of the access conditions number on HTTP GET response time (Figure 5a and 5b). Each policy contains one access condition, each including 5 triples. We progressively increased the number of access conditions protecting the target RDF resource. Not surprisingly, the number of access conditions defined on the protected resource impacts on response time. In Figure 5a we show the results for Shi3ld-LDP scenarios: data show a linear relationship between response time and access conditions number. We tested the system up to 100 access conditions, although common usage scenarios have a smaller number of conditions defined for each resource. For example, the 5 access condition case is approximately 3 times slower than unprotected access. Nevertheless, ditching SPARQL improved performance: Figure 5a shows that the SPARQL-less configuration is in average 25% faster than its SPARQL-based counterpart, due to the absence of the SPARQL interpreter. As predicted, the delay introduced by Shi3ld-GSP is higher, e.g., 7 times slower for resources protected by 5 access policies (Figure 5b). This is mainly due to the HTTP communication between the Shi3ld-GSP module and Fuseki. Further delay is introduced by the Fuseki GSP module, that translates HTTP operations into SPARQL queries. Moreover, unlike Shi3ld-LDP scenarios, Shi3ld-GSP uses a shared RDF store for protected resources and access control-related data (client attributes and access policies). This increases the execution time of SPARQL queries, thus determining higher response time: in Figure 5b, we show the behaviour of Shi3ld-GSP with two Fuseki server configurations: empty and with approximately 10M triples, stored in 17k graphs (we chose the "4-hop expansion Timbl crawl" part of the

[24] Binaries, code and complete evaluation results are available at: `http://wimmics.inria.fr/projects/shi3ld-ldp`

[25] `http://jena.apache.org/documentation/serving_data`

[26] `http://tinyurl.com/corese-engine`

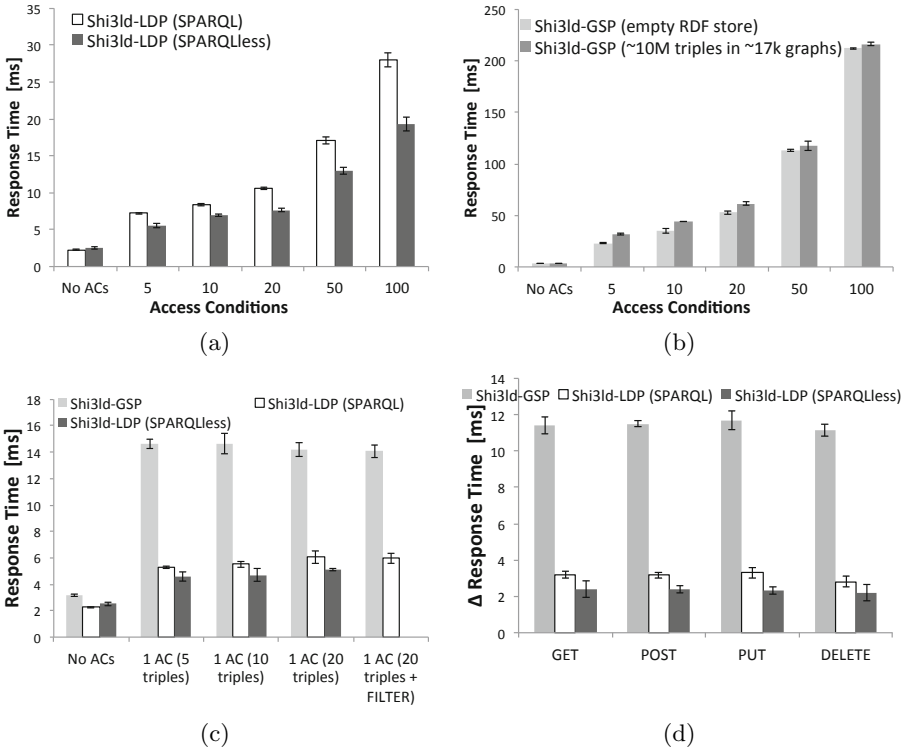


Fig. 5. Shi3ld response time evaluation. The "No ACs" column shows performance without access control.

Billion Triple Challenge 2012 Dataset[27]). Results show an average response time difference of 14%, with a 27% variation for the 5 access condition case (Figure 5b). The number and the distribution of triples in the RDF store influence Shi3ld-GSP response time. Results might vary when Shi3ld-GSP is coupled with SPARQL endpoints adopting different indexing strategies or with different triple number and graph partitioning.

In Figure 5c, we show the impact of access conditions complexity on HTTP GET response time. The requested resource is protected by a single access condition, with growing complexity: we added up to 20 triples, and we assess an access condition containing a FILTER clause (for SPARQL-based scenarios only). Results show no relevant impact on response time: this is because of the small size of the client attributes graph, over which access conditions are evaluated (in our tests, client attributes include 20 triples). Although attribute graph varies according to the application domain, it is reasonable that size will not exceed tens of triples.

The third test (Figure 5d) shows the delay introduced by Shi3ld for each HTTP operation. The figure displays the difference between response time with

[27] `http://km.aifb.kit.edu/projects/btc-2012/`

and without access control. We executed HTTP GET, POST, PUT and DELETE methods. Each HTTP method is associated to a 5-triple access condition. As predicted, the delay introduced by Shi3ld is independent from the HTTP method.

In Section 2, we addressed a qualitative comparison with respect to the related work. On the other hand, addressing a quantitative evaluation is a tricky point: among the list in Table 1, only few works explicitly designed for the Web come with an evaluation campaign [1,4,10,13,17]. Moreover, although some of these works provide a response time evaluation, the experimental conditions vary, making the comparison difficult.

5 Conclusions

We described an authorization framework for HTTP operations on Linked Data. The framework comes in three distinct configurations: Shi3ld-GSP, for the SPARQL 1.1 Graph Store Protocol, and Shi3ld-LDP for the Linked Data Platform (with and without the internal SPARQL endpoint). Our solutions feature attribute-based access control policies expressed with Web languages only. Evaluation confirms that Shi3ld-GSP is slower than the Shi3ld-LDP counterparts, due to the HTTP communication with the protected RDF store. Shi3ld-LDP with internal SPARQL endpoint introduces a 3x delay in response time (when resources are protected by 5 access conditions). Nevertheless, under the same conditions, the SPARQL-less solution exhibits a 25% faster response time. We show that response time grows linearly with the number of access conditions, and the complexity of each access condition does not impact on the delay.

Future work includes ensuring the trustworthiness of attributes sent by the client. Furthermore, a caching mechanism for client attributes must be introduced, to speed up the authorization procedure. The caching mechanism must be coupled with an efficient strategy to send attributes updates, to reduce the average size of HTTP requests. Finally, an effective administration interface to define access policies has to be designed, as user interaction issues should not be underestimated.

References

1. Abel, F., De Coi, J.L., Henze, N., Koesling, A.W., Krause, D., Olmedilla, D.: Enabling Advanced and Context-Dependent Access Control in RDF Stores. In: Aberer, K., et al. (eds.) ISWC/ASWC 2007. LNCS, vol. 4825, pp. 1–14. Springer, Heidelberg (2007)
2. Corby, O., Faron-Zucker, C.: The KGRAM Abstract Machine for Knowledge Graph Querying. In: Procs. of WI, pp. 338–341. IEEE (2010)
3. Corradi, A., Montanari, R., Tibaldi, D.: Context-Based Access Control Management in Ubiquitous Environments. In: Procs. of NCA, pp. 253–260. IEEE (2004)
4. Costabello, L., Villata, S., Gandon, F.: Context-Aware Access Control for RDF Graph Stores. In: Procs. of ECAI, pp. 282–287 (2012)
5. Covington, M.J., Long, W., Srinivasan, S., Dey, A.K., Ahamad, M., Abowd, G.D.: Securing Context-aware Applications using Environment Roles. In: Procs. of SACMAT, pp. 10–20. ACM (2001)

6. Cuppens, F., Cuppens-Boulahia, N.: Modeling Contextual Security Policies. Int. J. Inf. Sec. 7(4), 285–305 (2008)
7. Dey, A.K.: Understanding and using context. Personal and Ubiquitous Computing 5, 4–7 (2001)
8. Duckham, M.: Moving Forward: Location Privacy and Location Awareness. In: Procs. of SPRINGL, pp. 1–3. ACM (2010)
9. Finin, T.W., Joshi, A., Kagal, L., Niu, J., Sandhu, R.S., Winsborough, W.H., Thuraisingham, B.M.: R*OWL*BAC: representing role based access control in *OWL*. In: Procs. of SACMAT, pp. 73–82. ACM (2008)
10. Flouris, G., Fundulaki, I., Michou, M., Antoniou, G.: Controlling Access to RDF Graphs. In: Berre, A.J., Gómez-Pérez, A., Tutschku, K., Fensel, D. (eds.) FIS 2010. LNCS, vol. 6369, pp. 107–117. Springer, Heidelberg (2010)
11. Giereth, M.: On Partial Encryption of RDF-Graphs. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 308–322. Springer, Heidelberg (2005)
12. Giunchiglia, F., Zhang, R., Crispo, B.: Ontology Driven Community Access Control. In: Procs. of SPOT (2009)
13. Hollenbach, J., Presbrey, J., Berners-Lee, T.: Using RDF Metadata to Enable Access Control on the Social Semantic Web. In: Procs. of CK (2009)
14. Hulsebosch, R., Salden, A., Bargh, M., Ebben, P., Reitsma, J.: Context Sensitive Access Control. In: Procs. of SACMAT, pp. 111–119. ACM (2005)
15. Krumm, J.: A Survey of Computational Location Privacy. Personal Ubiquitous Comput. 13(6), 391–399 (2009)
16. Kulkarni, D., Tripathi, A.: Context-aware Role-based Access Control in Pervasive Computing Systems. In: Procs. of SACMAT, pp. 113–122. ACM (2008)
17. Muhleisen, H., Kost, M., Freytag, J.C.: SWRL-based Access Policies for Linked Data. In: Procs. of SPOT (2010)
18. Priebe, T., Fernández, E.B., Mehlau, J.I., Pernul, G.: A Pattern System for Access Control. In: Procs. of DBSec, pp. 235–249. Kluwer (2004)
19. Sacco, O., Passant, A., Decker, S.: An Access Control Framework for the Web of Data. In: Proc. of TrustCom, pp. 456–463. IEEE (2011)
20. Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-Based Access Control Models. IEEE Computer 29(2), 38–47 (1996)
21. Shen, H., Cheng, Y.: A Semantic Context-Based Model for Mobile Web Services Access Control. I. J. Computer Network and Information Security 1, 18–25 (2011)
22. Toninelli, A., Montanari, R., Kagal, L., Lassila, O.: A Semantic Context-Aware Access Control Framework for Secure Collaborations in Pervasive Computing Environments. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 473–486. Springer, Heidelberg (2006)

Bio2RDF Release 2: Improved Coverage, Interoperability and Provenance of Life Science Linked Data

Alison Callahan[1,*], José Cruz-Toledo[1,*], Peter Ansell[2], and Michel Dumontier[1]

[1] Department of Biology, Carleton University, Ottawa, Canada
{acallaha,jctoledo}@connect.carleton.ca,
michel_dumontier@carleton.ca
[2] eResearch Lab, School of ITEE, University of Queensland, Brisbane, Australia
ansell.peter@gmail.com

Abstract. Bio2RDF currently provides the largest network of Linked Data for the Life Sciences. Here, we describe a significant update to increase the overall quality of RDFized datasets generated from open scripts powered by an API to generate registry-validated IRIs, dataset provenance and metrics, SPARQL endpoints, downloadable RDF and database files. We demonstrate federated SPARQL queries within and across the Bio2RDF network, including semantic integration using the Semanticscience Integrated Ontology (SIO). This work forms a strong foundation for increased coverage and continuous integration of data in the life sciences.

Keywords: Semantic Web, RDF, Linked Data, Life Sciences, SPARQL.

1 Introduction

With the advent of the World Wide Web, journals have increasingly augmented their peer-reviewed journal publications with downloadable experimental data. While the increase in data availability should be cause for celebration, the potential for biomedical discovery across all of these data is hampered by access restrictions, incompatible formats, lack of semantic annotation and poor connectivity between datasets [1]. Although organizations such as the National Center for Biotechnology Information (NCBI) and the European Bioinformatics Institute (EBI) have made great strides to extract, capture and integrate data, the lack of formal, machine-understandable semantics results in ambiguity in the data and the relationships between them. With over 1500 biological databases, it becomes necessary to implement a more sophisticated scheme to unify the representation of diverse biomedical data so that it becomes easier to integrate and explore [2]. Importantly, there is a fundamental need to capture the provenance of these data in a manner that will support experimental design and reproducibility in scientific research. Providing data also presents real practical challenges, including ensuring persistence, availability, scalability, and providing the right tools to facilitate data exploration including query formulation.

[*] These authors contributed equally to this work.

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 200–212, 2013.

The Resource Description Framework (RDF) provides an excellent foundation to build a unified network of linked data on the emerging Semantic Web. While an increasing number of approaches are being proposed to describe and integrate specific biological data [3-5], it is the lack of coordinated identification, vocabulary overlap and alternative formalizations that challenges the promise of large-scale integration [6]. Formalization of data into ontologies using the Web Ontology Language (OWL) have yielded interesting results for integration, classification, consistency checking and more effective query answering with automated reasoning [7-11]. However, these efforts build the ontology in support of the task and there is little guarantee that the formalization will accommodate future data or support new applications. Alternatively, integration of data may be best facilitated by independent publication of datasets and their descriptions and subsequent coordination into integrative ontologies or community standards. This approach provides maximum flexibility for publishing original datasets with publisher provided descriptors in that they are not constrained by limited standards, but provides a clear avenue for future integration into a number of alternative standards.

Bio2RDF is a well-recognized open-source project that provides linked data for the life sciences using Semantic Web technologies. Bio2RDF scripts convert heterogeneously formatted data (e.g. flat-files, tab-delimited files, dataset specific formats, SQL, XML *etc*.) into a common format – RDF. Bio2RDF follows a set of basic conventions to generate and provide Linked Data which are guided by Tim Berners-Lee's design principles [1], the Banff Manifesto [2] and the collective experience of the Bio2RDF community. Entities, their attributes and relationships are named using a simple convention to produce Internationalized Resource Identifiers (IRIs) while statements are articulated using the lightweight semantics of RDF Schema (RDFS) and Dublin Core. Bio2RDF IRIs are resolved through the Bio2RDF Web Application, a servlet that answers Bio2RDF HTTP requests by formulating SPARQL queries against the appropriate SPARQL endpoints.

Although several efforts for provisioning linked life data exist such as Neurocommons [12], LinkedLifeData [13], W3C HCLS[3], Chem2Bio2RDF [14] and BioLOD, Bio2RDF stands out for several reasons: i) Bio2RDF is open source and freely available to use, modify or redistribute, ii) it acts on a set of basic guidelines to produce syntactically interoperable linked data across all datasets, iii) does not attempt to marshal data into a single global schema, iv) provides a federated network of SPARQL endpoints and v) provisions the community with an expandable global network of mirrors that host RDF datasets. Thus, Bio2RDF uniquely offers a community-focused resource for creating and enhancing the quality of biomedical data on the Semantic Web.

Here, we report on a second coordinated release of Bio2RDF Release 2 (R2), which yields substantial increases in syntactic and semantic interoperability across refactored Bio2RDF datasets. We address the problem of IRI inconsistency arising from independently generated scripts through an API over a dataset registry

[1] http://www.w3.org/DesignIssues/Principles.html

[2] https://sourceforge.net/apps/mediawiki/bio2rdf/index.php?title=Banff_Manifesto

[3] http://www.w3.org/blog/hcls/

to generate validated IRIs. We further generate provenance and statistics for each dataset, and provide public SPARQL endpoints, downloadable database files and RDF files. We demonstrate federated SPARQL queries within and across the Bio2RDF network, including queries that make use of the Semanticscience Integrated Ontology (SIO) [4], which provides a simple model with a rich set of relations to coordinate ontologies, data and services.

2 Methods

In the following section we will discuss the procedures and improvements used to generate Bio2RDF R2 compliant Linked Open Data including entity naming, dataset provenance and statistics, ontology mapping, query and exploration.

2.1 Entity Naming

For data with a source assigned identifier, entities are named as follows:

`http://bio2rdf.org/`*`namespace`*`:`*`identifier`*

where 'namespace' is the preferred short name of a biological dataset as found in our dataset registry and the 'identifier' is the unique string used by the source provider to identify any given record. For example, the HUGO Gene Nomenclature Committee identifies the human prostaglandin E synthase gene (PIG12) with the accession number "9599". This dataset is assigned the namespace "hgnc" in our dataset registry, thus, the corresponding Bio2RDF IRI is

`http://bio2rdf.org/hgnc:9599`

For data lacking a source assigned identifier, entities are named as follows:

`http://bio2rdf.org/`*`namespace`*`_resource:`*`identifier`*

where 'namespace' is the preferred short name of a biological dataset as found in our dataset registry and 'identifier' is uniquely created and assigned by the Bio2RDF script. This pattern is often used to identify objects that arise from the conversion of n-ary relations into an object with a set of binary relations. For example, the Comparative Toxicogenomics Database (CTD) describes associations between diseases and drugs, but does not specify identifiers for these associations, and hence we assign a new stable identifier for each, such as

`http://bio2rdf.org/ctd_resource:C112297D029597`

for the chemical-disease association between 10,10-bis(4-pyridinylmethyl)-9(10H)-anthracenone (mesh:C112297) and the Romano-Ward Syndrome (mesh:D029597).

Finally, dataset-specific types and relations are named as follows:

`http://bio2rdf.org/`*`namespace`*`_vocabulary:`*`identifier`*

[4] `http://code.google.com/p/semanticscience/wiki/SIO`

where 'namespace' is the preferred short name of a biological dataset as found in our dataset registry and 'identifier' is uniquely created and/or assigned by the Bio2RDF script. For example, the NCBI's HomoloGene resource provides groups of homologous eukaryotic genes and includes references to the taxa from which the genes were isolated. Hence, the Homologene group is identified as a class

```
http://bio2rdf.org/homologene_vocabulary:HomoloGene_Group
```

while the taxonomic relation is specified with:

```
http://bio2rdf.org/homologene_vocabulary:has_taxid
```

2.2 Open Source Scripts

In 2012, we consolidated the set Bio2RDF open source[5] scripts into a single GitHub repository (bio2rdf-scripts[6]). GitHub facilitates collaborative development through project forking, pull requests, code commenting, and merging. Thirty PHP scripts, one Java program and a Ruby gem are now available for any use (including commercial), modification and redistribution by anyone wishing to generate BioRDF data, or to improve the quality of RDF conversions currently used in Bio2RDF.

2.3 Programmatically Accessible Resource Registry

In order to ensure consistency in IRI assignment by different scripts, we established a common resource registry that each script must make use of. The resource registry specifies a unique namespace for each of the datasets (a.k.a. namespace; *e.g.* 'pdb' for the Protein Data Bank), along with synonyms (*e.g.* ncbigene, entrez gene, entrezgene/locuslink for the NCBI's Gene database), as well as primary and secondary IRIs used within the datasets (*e.g.* http://purl.obolibrary.org/obo/, http://purl.org/obo/owl/, http://purl.obofoundry.org/namespace, *etc.*) when applicable. The use of the registry in this way ensures a high level of syntactic interoperability between the generated linked data sets.

2.4 Provenance

Bio2RDF scripts now generate provenance using the Vocabulary of Interlinked Datasets (VoID), the Provenance vocabulary (PROV) and Dublin Core vocabulary. As illustrated in Fig. 1, each item in a dataset is linked using void:inDataset to a provenance object (typed as void:Dataset). The provenance object represents a Bio2RDF dataset, in that it is a version of the source data whose attributes include a label, the creation date, the creator (script URL), the publisher (Bio2RDF.org), the Bio2RDF license and rights, the download location for the dataset and the SPARQL endpoint in which the resource can be found. Importantly, we use the W3C PROV relation 'wasDerivedFrom' to link this Bio2RDF dataset to the source dataset, along with its licensing and source location.

---

[5] http://opensource.org/licenses/MIT

[6] https://github.com/bio2rdf/bio2rdf-scripts

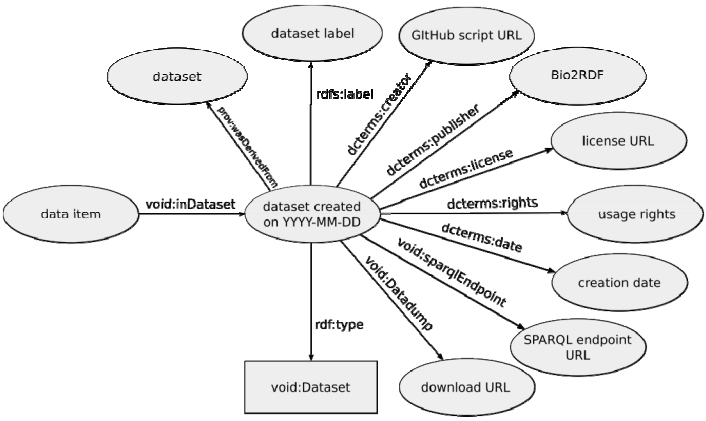


Fig. 1. The Bio2RDF R2 provenance model

2.5 Dataset Metrics

A set of nine dataset metrics are computed for each dataset that summarize their contents:

- total number of triples
- number of unique subjects
- number of unique predicates
- number of unique objects
- number of unique types
- number of unique objects linked from each predicate
- number of unique literals linked from each predicate
- number of unique subjects and objects linked by each predicate
- unique subject type-predicate-object type links and their frequencies

These metrics are serialized as RDF using our own vocabulary using the namespace `http://bio2rdf.org/dataset_vocabulary`), and subsequently loaded into a named graph at each dataset SPARQL endpoint with the following pattern:

`http://bio2rdf.org/bio2rdf-`*`namespace`*`-statistics`

where *namespace* is the preferred short name for the Bio2RDF dataset. While the values for metrics 1-4 are provided via suitably named datatype properties, metrics 5-9 require a more complex, typed object. For instance, a SPARQL query to retrieve all type-predicate-type links and their frequencies from the CTD endpoint is:

```
PREFIX statistics: <http://bio2rdf.org/dataset_vocabulary:>
SELECT *
FROM <http://bio2rdf.org/bio2rdf-ctd-statistics>
WHERE {
 ?endpoint a statistics:Endpoint.
```

```
 ?endpoint statistics:has_type_relation_type_count ?c.
  ?c statistics:has_subject_type ?subjectType.
  ?c statistics:has_subject_count ?subjectCount.
  ?c statistics:has_predicate ?predicate.
  ?c statistics:has_object_type ?objectType.
  ?c statistics:has_object_count ?objectCount.
}
```

Furthermore, to support context-sensitive SPARQL query formulation using SparQLed [15], we generated the data graph summaries using the Dataset Analytics Vocabulary[7]. These are stored in each endpoint in the graph named http://sindice.com/analytics .

2.6 Bio2RDF to SIO Ontology Mapping

Since each Bio2RDF dataset is expressed in terms of a dataset-specific vocabulary for its types and relations, it becomes rather challenging to compose federated queries across both linked datasets as well as datasets that overlap in their content. To facilitate dataset-independent querying, Bio2RDF dataset-specific vocabulary were mapped to the Semanticscience Integrated Ontology (SIO), which is also being used to map vocabularies used to describe SADI-based semantic web services. Dataset specific types and relations were extracted using SPARQL queries and manually mapped to corresponding SIO classes, object properties and datatype properties using the appropriate subclass relation (i.e. rdfs:subClassOf, owl:SubObjectPropertyOf). Bio2RDF dataset vocabularies and their SIO-mappings are stored in separate OWL ontologies on the bio2rdf-mapping GitHub repository[8].

2.7 SPARQL Endpoints

Each dataset was loaded into a separate instance of OpenLink Virtuoso Community Edition version 6.1.6 with the faceted browser, SPARQL 1.1 query federation and Cross-Origin Resource Sharing (CORS) enabled.

2.8 Bio2RDF Web Application

Bio2RDF Linked Data IRIs are made resolvable through the Bio2RDF Web Application, a servlet based application that uses the QueryAll Linked Data library [16] to dynamically answer requests for Bio2RDF IRIs by aggregating the results of SPARQL queries to Bio2RDF SPARQL endpoints that are automatically selected based on the structure of the query IRI. The Web Application can be configured to resolve queries using multiple SPARQL endpoints, each of which may handle different namespaces and identifier patterns. Such configurations are stored as RDF, and specified using Web Application profiles. Profiles are designed to allow different

[7] http://vocab.sindice.net/analytics#
[8] https://github.com/bio2rdf/bio2rdf-mapping

hosts to reuse the same configuration documents in slightly different ways. For example, the Bio2RDF Web Application R2 profile has been configured to resolve queries that include the new '_resource' and '_vocabulary' namespaces (section 2.1), as well existing query types used by the base Bio2RDF profile, and to resolve these queries using the R2 SPARQL endpoints.

The Bio2RDF Web Application accepts RDF requests in the Accept Request and does not use URL suffixes for Content Negotiation, as most Linked Data providers do, as that would make it difficult to reliably distinguish identifiers across all of the namespaces that are resolved by Bio2RDF. Specifically, there is no guarantee that a namespace will not contain identifiers ending in the same suffix as a file format. For example, if a namespace had the identifier "plants.html", the Bio2RDF Web Application would not be able to resolve the URI consistently to non-HTML formats using Content Negotiation. For this reason, the Bio2RDF Web Application directive to resolve HTML is a prefixed path, which is easy for any scriptable User Agent to generate. In the example above the identifier could be resolved to an RDF/XML document using "/rdfxml/namespace:plants.html'', without any ambiguity as to the meaning of the request, as the file format is stripped from the prefix by the web application, based on the web application configuration.

2.9 Resolving Bio2RDF IRIs Using Multiple SPARQL Endpoints

The Bio2RDF Web Application is designed to be used as an interface to a range of different Linked Data providers. It includes declarative rules that are used to map queries between the Bio2RDF IRI format and the identifiers used by each Linked Data provider. For example, the Bio2RDF R2 Web Application has been configured to resolve queries of the form

```
http://bio2rdf.org/uniprot:P05067
```

using UniProt's new SPARQL endpoint, currently available at http://beta.sparql.uniprot.org/sparql. In this way, as it becomes increasingly commonplace for data providers to publish their data at their own SPARQL endpoints, Bio2RDF will be able to leverage these resources and incorporate them into the Bio2RDF network, while still supporting queries that follow Bio2RDF IRI conventions.

3 Results

3.1 Bio2RDF Release 2

Nineteen datasets, including 5 new datasets, were generated as part of R2 (**Table 1**). R2 also includes 3 datasets that are themselves aggregates of datasets which are now available as one resource. For instance, iRefIndex consists of 13 datasets (BIND, BioGRID, CORUM, DIP, HPRD, InnateDB, IntAct, MatrixDB, MINT, MPact, MPIDB, MPPI and OPHID) while NCBO's Bioportal collection currently consists of 100 OBO ontologies including ChEBI, Protein Ontology and the Gene Ontology.

We also have 10 additional updated scripts that are currently generating updated datasets and SPARQL endpoints to be available with the next release: ChemBL, DBPedia, GenBank, PathwayCommons, the RCSB Protein Databank, PubChem, PubMed, RefSeq, UniProt (including UniRef and UniParc) and UniSTS.

Dataset SPARQL endpoints are available at http://[namespace].bio2rdf.org. For example, the Saccharomyces Genome Database (SGD) SPARQL endpoint is available at http://sgd.bio2rdf.org. All updated Bio2RDF Linked Data and their corresponding Virtuoso DB files are available for download at http://download.bio2rdf.org.

Table 1. Bio2RDF Release 2 datasets with select dataset metrics. The asterisks indicate datasets that are new to Bio2RDF.

Dataset	*Namespace*	*# of triples*	*# of unique subjects*	*# of unique predicates*	*# of unique objects*
Affymetrix	affymetrix	44469611	1370219	79	13097194
Biomodels*	biomodels	589753	87671	38	209005
Bioportal*	bioportal	15384622	4425342	191	7668644
Comparative Toxicogenomics Database	ctd	141845167	12840989	27	13347992
DrugBank	drugbank	1121468	172084	75	526976
NCBI Gene	ncbigene	394026267	12543449	60	121538103
Gene Ontology Annotations	goa	80028873	4710165	28	19924391
HUGO Gene Nomenclature Committee	hgnc	836060	37320	63	519628
Homologene	homologene	1281881	43605	17	1011783
InterPro*	interpro	999031	23794	34	211346
iProClass	iproclass	211365460	11680053	29	97484111
iRefIndex	irefindex	31042135	1933717	32	4276466
Medical Subject Headings	mesh	4172230	232573	60	1405919
National Drug Code Directory*	ndc	17814216	301654	30	650650
Online Mendelian Inheritance in Man	omim	1848729	205821	61	1305149
Pharmacogenomics Knowledge Base	pharmgkb	37949275	5157921	43	10852303
SABIO-RK*	sabiork	2618288	393157	41	797554
Saccharomyces Genome Database	sgd	5551009	725694	62	1175694
NCBI Taxonomy	taxon	17814216	965020	33	2467675
Total	**19**	**1,010,758,291**	**57850248**	**1003**	**298470583**

3.2 Metric Informed Querying

Dataset metrics (section 2.5) provide an overview of the contents of a dataset and can be used to guide the development of SPARQL queries. **Table 2** shows values for the type-relation-type metric in the DrugBank dataset. In the first row we note that 11,512 unique pharmaceuticals are paired with 56 different units using the 'form' predicate, indicating the enormous number of possible formulations. Further in the list, we see that 1,074 unique drugs are involved in 10,891 drug-drug interactions, most of these arising from FDA drug product labels.

Table 2. Selected DrugBank dataset metrics describing the frequencies of type-relation-type occurrences. The namespace for subject types, predicates, and object types is http://bio2rdf.org/drugbank_vocabulary

Subject Type	*Subject Count*	*Predicate*	*Object Type*	*Object Count*
Pharmaceutical	11512	form	Unit	56
Drug-Transporter-Interaction	1440	drug	Drug	534
Drug-Transporter-Interaction	1440	transporter	Target	88
Drug	1266	dosage	Dosage	230
Patent	1255	country	Country	2
Drug	1127	product	Pharmaceutical	11512
Drug	**1074**	**ddi-interactor-in**	**Drug-Drug-Interaction**	**10891**
Drug	532	patent	Patent	1255
Drug	277	mixture	Mixture	3317
Dosage	230	route	Route	42
Drug-Target-Interaction	84	target	Target	43

The type-relation-type metric gives the necessary information to understand how object types are related to one another in the RDF graph. It can also inform the construction of an immediately useful SPARQL query, without losing time generating 'exploratory' queries to become familiar with the dataset model. For instance, the above table suggests that in order to retrieve the targets that are involved in drug-target interactions, one should specify the 'target' predicate, to link to a target from its drug-target interaction(s):

```
PREFIX drugbank_vocabulary:
<http://bio2rdf.org/drugbank_vocabulary:>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?dti ?target ?targetName
WHERE {
  ?dti a drugbank_vocabulary:Drug-Target-Interaction .
  ?dti drugbank_vocabulary:target ?target .
  ?target rdfs:label ?targetName.
}
```

Some of the results of this query are listed in Table 3.

Table 3. Partial results from a query to obtain drug-target interactions from the Bio2RDF DrugBank SPARQL endpoint

Drug Target Interaction IRI	*Target IRI*	*Target label*
drugbank_resource:DB00002_1102	drugbank_target:1102	"Low affinity immunoglobulin gamma Fc region receptor III-B [drugbank_target:1102]"@en
drugbank_resource:DB00002_3814	drugbank_target:3814	"Complement C1r subcomponent [drugbank_target:3814]"@en
drugbank_resource:DB00002_3815	drugbank_target:3815	"Complement C1q subcomponent subunit A [drug-bank_target:3815]"@en
drugbank_resource:DB00002_3820	drugbank_target:3820	"Low affinity immunoglobulin gamma Fc region receptor II-b [drugbank_target:3820]"@en
drugbank_resource:DB00002_3821	drugbank_target:3821	"Low affinity immunoglobulin gamma Fc region receptor II-c [drugbank_target:3821]"@en

Dataset metrics can also facilitate federated queries over multiple Bio2RDF endpoints in a similar manner. For example, the following query retrieves all biochemical reactions from the Bio2RDF Biomodels endpoint that are kinds of "protein catabolic process", as defined by the Gene Ontology in the NCBO Bioportal endpoint:

```
PREFIX biopax_vocab: <http://bio2rdf.org/biopax_vocabulary:>
SELECT ?go ?label count(distinct ?x)
WHERE {
    ?go rdfs:label ?label .
    ?go rdfs:subClassOf ?goparent OPTION (TRANSITIVE) .
    ?goparent rdfs:label ?parentlabel .
     FILTER strstarts(str(?parentlabel), "protein catabolic process")
    SERVICE <http://biomodels.bio2rdf.org/sparql> {
     ?x biopax_vocabulary:identical-to ?go .
     ?x         a         <http://www.biopax.org/release/biopax-
level3.owl#BiochemicalReaction> .
    }
}
```

3.3 Bio2RDF Dataset Vocabulary-SIO Mapping

The mappings between Bio2RDF dataset vocabularies and SIO make it possible to formulate queries that can be applied across all Bio2RDF SPARQL endpoints, and can be used to integrate data from multiple sources, as opposed to *a priori* formulation of dataset specific queries against targeted endpoints. For instance, we can ask for chemicals that effect the 'Diabetes II mellitus' pathway and that are available in tablet form using the Comparative Toxicogenomics Database (CTD) and the National Drug Codes (NDC) Bio2RDF datasets, and the mappings of their vocabularies to SIO:

```
define input:inference "http://bio2rdf.org/sio_mappings"
PREFIX sio: <http://semanticscience.org/resource/>
PREFIX ctd_vocab: <http:bio2rdf.org/ctd_vocabulary:>
PREFIX ndc_vocab: <http://bio2rdf.org/ndc_vocabulary:>
SELECT ?chemical ?chemicalLabel
WHERE {
  #SIO_01126: 'chemical substance'
  ?chemical a sio:SIO_011126.
  ?chemical rdfs:label ?chemicalLabel .
  #affects Diabetes mellitus pathway
  ?chemical ctd_vocab:pathway <http://bio2rdf.org/kegg:04930> .
  #dosage form: tablet, extended release
  ?chemical ndc_vocab:dosage-form
  <http://bio2rdf.org/ndc_vocabulary:00426c812b33febc3f9cd1fee8
  cc83ce> .
}
```

This query is possible because the classes 'ctd_vocab:Chemical' and 'ndc_vocab:human-prescription-drug' have been mapped as subclasses of the SIO class 'chemical substance'[9].

4 Discussion

Bio2RDF Release 2 marks several important milestones for the open source Bio2RDF project. First, the consolidation of scripts into a single GitHub repository will make it easier for the community to report problems, contribute code fixes, or contribute new scripts to add more data into the Bio2RDF network of linked data for the life sciences. Already, we are working with members of the W3C Linking Open Drug Data (LODD) to add their code to this GitHub repository, identify and select an open source license, and improve the linking of Bio2RDF data. With new RDF generation guidelines and example queries that demonstrate use of dataset metrics and provenance, we believe that Bio2RDF has the potential to become a central meeting point for developing the biomedical semantic web. Indeed, we welcome those that think Bio2RDF could be useful to their projects to contact us on the mailing list and participate in improving this community resource.

A major aspect of what makes Bio2RDF successful from a Linked Data perspective is the use of a central registry of datasets in order to normalize generated IRIs. Although we previously created a large aggregated namespace directory, the lack of extensive curation meant that the directory contained significant overlap and omissions. Importantly, no script specifically made use of this registry, and thus adherence to the namespaces was strictly in the hands of developers at the time of writing the code. In consolidating the scripts, we found significant divergence in the use of a preferred namespace for generating Bio2RDF IRIs, either because of the overlap in

[9] http://semanticscience.org/resource/SIO_011126

directory content, or in the community adopting another preferred prefix. With the addition of an API to automatically generate the preferred Bio2RDF IRI from any number of dataset prefixes (community-preferred synonyms can be recorded), all Bio2RDF IRIs can be validated such that unknown dataset prefixes must be defined in the registry. Importantly, our registry has been shared with maintainers of identifiers.org in order for their contents to be incorporated into the MIRIAM registry [17] which powers that URL resolving service. Once we have merged our resource listings, we expect to make direct use of the MIRIAM registry to list new entries, and to have identifiers.org list Bio2RDF as a resolver for most of its entries. Moreover, since the MIRIAM registry describes regular expressions that specify the identifier pattern, Bio2RDF scripts will be able to check whether an identifier is valid for a given namespace, thereby improving the quality of data produced by Bio2RDF scripts.

The dataset metrics that we now compute for each Bio2RDF dataset have significant value for users and providers. First, users can get fast and easy access to basic dataset metrics (number of triples, *etc*.) as well as more sophisticated summaries such as which types are in the dataset and how are they connected to one another. This data graph summary is the basis for SparQLed, an open source tool to assist in query composition through context-sensitive autocomplete functionality. Use of these summaries also reduces the server load for data provider servers, which in turns frees up resources to more quickly respond to interesting domain-specific queries. Second, we anticipate that these metrics may be useful in monitoring dataset flux. Bio2RDF now plans to provide bi-annual release of data, and as such, we will develop infrastructure to monitor change in order to understand which datasets are evolving, and how are they changing. Thus, users will be better able to focus in on content changes and providers will be able to make informed decisions about the hardware and software resources required to provision the data to Bio2RDF users.

Our demonstration of using SIO to map Bio2RDF dataset vocabularies helps facilitate the composition of queries for the basic kinds of data or their relationships. Since SIO contains unified and rich axiomatic descriptions of its classes and properties, in the future we intend to explore how these can be automatically reasoned about to improve query answering with newly entailed facts as well as to check the consistency of Bio2RDF linked data itself.

Acknowledgements. This research was supported by an NSERC CGSD to AC, and NSERC funding to JCT and MD. We also acknowledge technical support from Marc-Alexandre Nolin, constructive but anonymous peer-reviewers, and useful discussions from the Bio2RDF community.

References

1. Howe, D., Costanzo, M., Fey, P., Gojobori, T., Hannick, L., Hide, W., Hill, D.P., Kania, R., Schaeffer, M., St. Pierre, S., et al.: Big data: The future of biocuration. Nature 455(7209), 47–50 (2008)
2. Goble, C., Stevens, R.: State of the nation in data integration for bioinformatics. J. Biomed. Inform. 41(5), 687–693 (2008)
3. Cerami, E.G., Bader, G.D., Gross, B.E., Sander, C.: cPath: open source software for collecting, storing, and querying biological pathways. BMC Bioinformatics 7, 497 (2006)

4. Chen, H., Yu, T., Chen, J.Y.: Semantic Web meets Integrative Biology: a survey. Brief Bioinform. (2012)
5. Ruebenacker, O., Moraru, I.I., Schaff, J.C., Blinov, M.L.: Integrating BioPAX pathway knowledge with SBML models. IET Syst. Biol. 3(5), 317–328 (2009)
6. Sansone, S.A., Rocca-Serra, P., Field, D., Maguire, E., Taylor, C., Hofmann, O., Fang, H., Neumann, S., Tong, W., Amaral-Zettler, L., et al.: Toward interoperable bioscience data. Nat. Genet. 44(2), 121–126 (2012)
7. Berlanga, R., Jimenez-Ruiz, E., Nebot, V.: Exploring and linking biomedical resources through multidimensional semantic spaces. BMC Bioinformatics 13(suppl. 1), S6 (2012)
8. Gennari, J.H., Neal, M.L., Galdzicki, M., Cook, D.L.: Multiple ontologies in action: composite annotations for biosimulation models. J. Biomed. Inform. 44(1), 146–154 (2011)
9. Hoehndorf, R., Dumontier, M., Gennari, J.H., Wimalaratne, S., de Bono, B., Cook, D.L., Gkoutos, G.V.: Integrating systems biology models and biomedical ontologies. BMC Syst. Biol. 5, 124 (2011)
10. Hoehndorf, R., Dumontier, M., Oellrich, A., Rebholz-Schuhmann, D., Schofield, P.N., Gkoutos, G.V.: Interoperability between biomedical ontologies through relation expansion, upper-level ontologies and automatic reasoning. PLoS One 6(7), e22006 (2011)
11. Jonquet, C., Lependu, P., Falconer, S., Coulet, A., Noy, N.F., Musen, M.A., Shah, N.H.: NCBO Resource Index: Ontology-Based Search and Mining of Biomedical Resources. Web Semant. 9(3), 316–324 (2011)
12. Ruttenberg, A., Rees, J.A., Samwald, M., Marshall, M.S.: Life sciences on the Semantic Web: the Neurocommons and beyond. Brief Bioinform. 10(2), 193–204 (2009)
13. Momtchev, V., Peychev, D., Primov, T., Georgiev, G.: Expanding the Pathway and Interaction Knowledge in Linked Life Data. In: Semantic Web Challenge: 2009, Amsterdam (2009)
14. Chen, B., Dong, X., Jiao, D., Wang, H., Zhu, Q., Ding, Y., Wild, D.J.: Chem2Bio2RDF: a semantic framework for linking and data mining chemogenomic and systems chemical biology data. BMC Bioinformatics 11, 255 (2010)
15. Campinas, S., Perry, T.E., Ceccarelli, D., Delbru, R., Tummarello, G.: Introducing RDF Graph Summary with Application to Assisted SPARQL Formulation, pp. 261–266 (2012)
16. Ansell, P.: Model and prototype for querying multiple linked scientific datasets. Future Generation Computer Systems 27(3), 329–333 (2011)
17. Juty, N., Le Novere, N., Laibe, C.: Identifiers.org and MIRIAM Registry: community resources to provide persistent identification. Nucleic Acids Res. 40(Database issue), D580–D586 (2012)

Observing Linked Data Dynamics

Tobias Käfer[1], Ahmed Abdelrahman[2], Jürgen Umbrich[2], Patrick O'Byrne[2], and Aidan Hogan[2]

[1] Institute AIFB, Karlsruhe Institute of Technology, Germany
[2] Digital Enterprise Research Institute, National University of Ireland Galway

Abstract. In this paper, we present the design and first results of the *Dynamic Linked Data Observatory*: a long-term experiment to monitor the two-hop neighbourhood of a core set of eighty thousand diverse Linked Data documents on a weekly basis. We present the methodology used for sampling the URIs to monitor, retrieving the documents, and further crawling part of the two-hop neighbourhood. Having now run this experiment for six months, we analyse the dynamics of the monitored documents over the data collected thus far. We look at the estimated lifespan of the core documents, how often they go on-line or off-line, how often they change; we further investigate domain-level trends. Next we look at changes within the RDF content of the core documents across the weekly snapshots, examining the elements (i.e., triples, subjects, predicates, objects, classes) that are most frequently added or removed. Thereafter, we look at how the links between dereferenceable documents evolves over time in the two-hop neighbourhood.

1 Introduction

The Web of (Linked) Data is unquestionably dynamic: over time, documents come online, documents go offline, and the content of online documents changes. However, the dynamics of Linked Data are not yet well understood in terms of how stable documents are over time, what kinds of changes are most frequently encountered, and so forth. Knowledge about Linked Data dynamics is important for a wide range of applications: effective caching, link maintenance, versioning, etc. The current lack of understanding about Linked Data dynamics can be attributed to a lack of suitable collections to analyse: to track changes over time, and to ultimately derive meaningful results about the dynamics of Linked Data, we need to monitor a fixed set of diverse Linked Data documents at a fixed interval over a long period of time. As of yet, no such collection has been made available to the Linked Data research community.

In this paper, we aim to shed light on the dynamics of Linked Data. We first present some use-cases to motivate why knowledge about the dynamics of Linked Data is important to the community (§ 2). We then introduce the DYnamic Linked Data Observatory (DYLDO), which we have created to monitor a fixed set of Linked Data documents (and their neighbourhood) on a weekly basis for an indefinite period of time: we discuss our methodology for selecting

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 213–227, 2013.

documents and for monitoring these sources (§ 3). After six months of monitoring, we analyse the 29 weekly snapshots collected and analyse the dynamics exhibited by documents in the collection. We look at: (§ 4) the stability and lifespan of documents in the snapshots, and how often their content changes; and (§ 5) the types of changes these documents undergo: are they additions or deletions, what elements of the RDF document changed, and so forth.

This paper is a continuation of a previous workshop paper [5], where we originally motivated and outlined the methodology for our DYnamic Linked Data Observatory. Herein, we summarise discussion on the observatory and focus on our first concrete results for Linked Data dynamics after 29 weeks of monitoring.

2 Motivation and Novelty

We first discuss a brief selection of use-cases to help motivate our work on Linked Data dynamics and its importance to the community.

Focused Synchronisation: Various centralised search & query approaches for Linked Data rely on locally replicated copies of RDF harvested from the Web. As the original sources change, replicated indexes become stale, affecting the up-to-dateness of results. More fine-grained knowledge about Linked Data dynamics would allow centralised engines to, e.g., focus on keeping synchronised with those domains whose contributions change rapidly.

Smart Caching: Conversely, "live querying" approaches for Linked Data dereference and discover sources on the fly. However, remote lookups are expensive to execute. Knowledge about Linked Data dynamics can help to identify which sources can be cached to save time and resources, how long cached data can be expected to remain valid, and whether there are dependencies in the cache (e.g., if a document from a particular domain changes, should all documents from that domain be invalidated?).

Hybrid Architectures: A core engineering trade-off for systems dealing with lots of data is pre-processing overhead vs. runtime-processing overhead. In general, pre-processing (e.g., caching, local indexing, reasoning materialisation, etc.) is better suited to static data, whereas runtime-processing (e.g., live dereferencing, backward-chaining, etc.) is better suited to dynamic data. In a hybrid architecture, knowledge about dynamics can be used to delegate both data and requests into static/dynamic pipelines. Static data can be cached and deeply pre-processed, whereas dynamic requests may invoke a "live querying" component or backward-chaining reasoning, and so forth.

Link Maintenance: When Linked Data publishers embed links to external domains in their data, deadlinks will occur after some time, or links may no longer be appropriate after remote data changes. Furthermore, novel sources may serve as useful targets to link. Knowledge about dynamics can help publishers to decide how frequently their link-sets need to be updated depending on, e.g., the domain they target or the type of link.

Versioning: When changes are made to a dataset, versioning should be applied to ensure that parties relying on the data in question do not suffer

adverse effects (e.g., through use of deprecation instead of simply removing data). Versioning is particularly relevant for vocabularies on the Web, whose semantics may change over time to reflect usage. Knowledge about Linked Data dynamics can show how changes propagate on the Web and inform the design of mature versioning methodologies.

In terms of existing works, various papers on the dynamics of the HTML-centric Web have been published by, e.g., Coffman et al. [3], Brewington and Cybenko [1], Lim et al. [8], Cho and Garcia-Molina [2], Fetterly et al. [4] and Ntoulas et al. [9]. These works analysed the rate of change of documents, patterns in change (e.g., time of day, day of the week), growth rate of the Web, dynamicity of links, the relation between top-level domains and dynamicity, etc. We refer readers to the broad survey by Ke et al. [6] about Web dynamics. As opposed to these related works, we focus specifically on the dynamicity of RDF documents in the context of Linked Data.

Few papers specifically analyse RDF or Linked Data dynamics. Popitsch and Haslhofer [10] propose DSNotify to help maintain links between datasets, but only have knowledge of DBpedia dynamics. In previous work, we showed that two centralised query indexes of Linked Data (OpenLink's LOD Cache[1] and Sindice's SPARQL endpoint[2]) often return stale results [13]. In another previous work, we analysed changes in documents over 24 snapshots of RDF Web data [12]; however, the coverage of each snapshot varied and our analysis was rather "best-effort". Addressing this problem, we later proposed the Dynamic Linked Data Observatory [5] to collect the snapshots upon which this work is based.

3 Dynamic Linked Data Observatory

To study the dynamics of Linked Data in a principled way, we require a principled way of monitoring a sample of Linked Data documents over time. Given a lack of suitable data available elsewhere, earlier this year we proposed and implemented the Dynamic Linked Data Observatory to perform this monitoring [5]. Each week, a fixed set of documents is retrieved and the content stored. From this core set of documents, we perform a brief crawl to find well-linked documents in their close neighbourhood. We began the weekly monitoring experiments on 2012/05/06, and have collected 29 snapshots until the time of writing. Herein, we outline our methodology for sampling and monitoring documents. *Full details of our sampling and crawling configurations are available in [5].*

3.1 Sampling Methodology

We wish to monitor a broad cross-section of Linked Data documents for a sample that would lead to manageable weekly snapshot sizes and that would not overburden publishers with repetitive deep crawls. In February 2012, we extracted

[1] `http://lod.openlinksw.com/sparql`; retr. 2013/03/12.
[2] `http://sparql.sindice.com/`; retr. 2013/03/12.

the list of 220 URIs available on the DataHub site under the "LOD cloud" group, offering entry points for (most) of the datasets listed in the LOD cloud.[3] To this, we added the top-220 documents extracted from the Billion Triple Challenge (BTC) 2011 dataset as determined by PageRank over the graph of dereferenceable documents. These initial 440 URIs offer core entry points into both the LOD cloud and BTC perspectives of the Web of Data (see [5] for details).

From these 440 URIs, we then wished to expand our sample by means of a crawl that would stay in the vicinity of our core URIs (and, e.g., avoid getting trapped in high-volume exporters with low out-degrees such as the hi5.com or livejournal.com sites). We thus performed a 2-hop breadth first crawl using the 440 URIs as our seed-list, considering all URIs mentioned in an RDF document as a potential link, looking for RDF/XML, RDFa, N-Triples or Turtle content, enforcing a two-second politeness delay between lookups to the same site. We repeated this crawl 10 times to account for the possibility of non-determinism and instability of hosted documents. We then took the union of all URIs that dereferenced to RDF content in one of the crawls, resulting in a core monitoring set of 95,737 dereferenceable URIs spanning 652 pay-level domains[4], giving an average of 146.8 dereferenceable URIs per domain (see [5] for full details).

3.2 Monitoring Methodology

The core aim of the weekly monitoring setup is to dereference and download the content for the list of 95,737 URIs sampled in the previous step. Since this set of documents is static, we also extend this "kernel" of monitored data by crawling a further 95,737 URIs starting from the kernel. The content of this extended crawl varies from week to week, and captures new documents in the neighbourhood of the kernel, as well as raw data reflecting changes in the link-structure of the kernel. The extension of the kernel is done by breadth-first crawl, and involves at least 2 hops (sometimes 3 hops) to meet the quota.

We have performed this monitoring on a weekly basis since 2012/05/06, yielding 29 weekly snapshots at the time of writing. Each snapshot consists of the content retrieved for the core kernel URIs (following redirects), the content of the expanded crawl, a set of redirects, and access logs for URIs that were looked up. Table 1 enumerates the average and total amount of data retrieved over the 29 weeks for the kernel documents and for the expanded crawl. The 95,737 kernel URIs yielded an average of 68,997 documents: though all URIs were deemed to dereference to RDF during sampling, some dereference to the same RDF document and some now fail to dereference. The number of unique documents appearing in at least one kernel snapshot was 86,696 and the analogous figure for domains was 620 (vs. 652 for the source URIs). In terms of the diversity of the kernel, the documents in each snapshot came from an average of 573.6 domains. The sum of all kernel snapshots yields around 464 million quadruples.

[3] `http://thedatahub.org/group/lodcloud`; retr. 2013/03/12.

[4] The level of domain which an agent can register and must pay for: e.g., dbpedia.org, bbc.co.uk. We may refer to pay-level-domains as PLDs or as simply "domains".

Table 1. Overall statistics across all 29 snapshots

Statistic	Kernel	Extended
Mean pay-level domains	573.6 ±16.6	1,738.6 ±218
Mean documents	68,996.9 ±5,555.2	152,355.7 ±2,356.3
Mean quadruples	16,001,671 ±988,820	94,725,595 ±10,279,806
Sum quadruples	464,048,460	2,747,042,282

By comparison, the extended snapshots contain 3× the number of domains, 2.2× the number of documents, and 5.9× the amount of raw data (there was thus a higher statement per document ratio in the extended crawl). In this paper, we currently focus on a first analysis of changes within the kernel documents.

4 Document-Level Dynamics

In this section, for the documents retrieved from the fixed set of kernel URIs, we first look at the availability of documents over time, the estimated life-span and death-rate of these documents, and their rates of change.

4.1 Availability/Occurrence

As aforementioned, 86,696 RDF documents appeared in (i.e., returned content for) at least one kernel. Figure 1 shows the distribution of the availability of these documents, counting for how many snapshots they appeared, measuring their stability over the 29 weeks. We see that 26% were available for all 29 weeks of the monitoring period. 55% of documents were available for 27 weeks or more and the mean availability for documents was 23.1 snapshots (79.7% availability).

With respect to this "one-in-five" unavailability of documents, Figure 2 provides a breakdown of the HTTP response codes and errors encountered while accessing URIs in the kernel (after following redirects). Response codes in `2xx` are most common: all of these were `200 Okay` indicating content was returned. The remaining responses indicate errors, where we see increasing instability over the monitoring time-frame. Most errors were `5xx` server error codes, the most common (32%) of which were `500 Internal Server Error`. The "other" category of errors related to unknown hosts and other HTTP-level inconsistencies such as self-redirects. A small but growing number of errors were `4xx` codes, 96% of which were specifically `404 Not Found`, indicating that there is no longer any document at that location. We next investigate these "dead documents".

Discussion: A one-in-five unavailability rate suggests that an agent traversing Linked Data documents can, on a single pass, expect to miss about 20% of potential content. This unavailability is not unique to Linked Data: for example, looking at 151 million HTML pages in 2003, Fetterly et al. [4] managed to download only 49.2% of pages eleven times in eleven weeks; in fact, our results are much more stable by comparison (cf. [4, Figure 4] and Figure 2). One may then ask how often unavailability is temporary, rather than permanent.

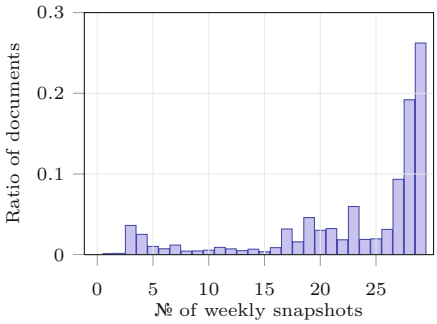


Fig. 1. Appearances of documents

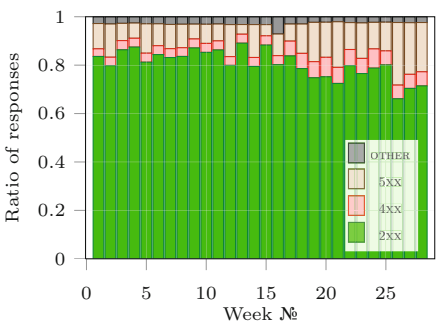


Fig. 2. Response distributions

4.2 Death Rate

Given estimates about their stability, we now estimate the loss of documents in the kernel over time by identifying *dead documents*: documents that (are likely to) have gone permanently offline. First, we look at the *last-heartbeat* of documents: the last weekly snapshot in which the document appeared such that, e.g., if a document was last seen in week 2, this document is unlikely to ever return. Figure 3 shows the evolving last heart-beats of kernel documents where, e.g., we see that 95% of documents have appeared at least once since the 14^{th} snapshot (2012/08/05). The further left the life-span, the longer the document is offline and the less likely that it will return. Thus the sharp downward trend observable for the last three snapshots could be due to temporary issues.

Taking another perspective, we also estimate the death-rate of documents by looking specifically at `404` errors that often indicate a permanent error (vs. `5xx` codes that may indicate, e.g., temporary unavailability or bugs). We found that 98.3% of URIs that return a `404` error never return content again in our monitoring frame, and 99.7% of URIs that return two sequential `404` errors never return. Based on returning a sequence of `404` codes up to the most recent snapshot, Figure 4 shows the rate at which documents die in a manner comparable with the analogous "last heart-beat" measures: the `404` death-rate likely underestimates the amount of dead documents (since it does not cover all possible causes), whereas the last heart-beat measure certainly overestimates the amount of dead documents. Combining both perspectives, 5% of documents have returned a trailing sequence of five or more `404`s or have been offline for more than 14 weeks, strongly indicating death.

Discussion: The one-in-twenty death-rate of Linked Data documents over six-months is pertinent for link-maintenance (detecting and avoiding dead-links) and for cache maintenance. The death-rate of 5% over six months can be compared favourably with death-rates of 20.5% observed by Koehler [7] in 1999 and 48% observed by Ntoulas et al. [9] in 2004 for HTML documents. We conjecture that since (cool) URIs also serve as names in Linked Data, such documents often have more stable URLs than, e.g., HTML URLs that often contain query strings.

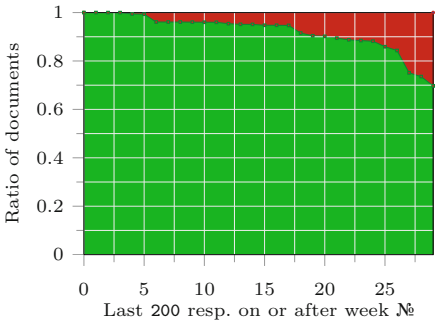


Fig. 3. Last heartbeat of documents

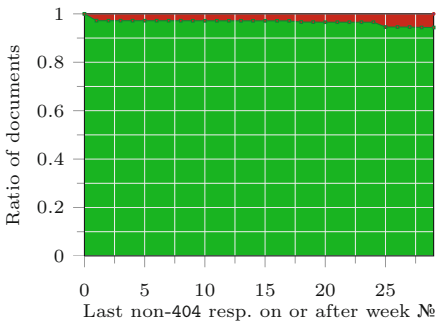


Fig. 4. Documents reported dead

4.3 Change Ratio

Next we compare the RDF content of the documents on a week-to-week basis. For each document, we compare 28 sequential version pairs. If a document was not available for a given week, we make the comparison with the most recent available version of the document. We wish to compare RDF content and not document syntax: thus, our comparison is complicated by the presence of existential blank nodes. In theory, our crawler uses a deterministic mechanism for labelling blank-nodes such that, for a given Web document, the labels of blank nodes will be consistent if blank nodes are consistently labelled in the source document and/or the order of implicit blank nodes remains the same. However, some documents in our collection provide fresh, explicit blank node labels upon every access.[5] Henceforth, when comparing graphs, we apply an approximation whereby we rewrite all blank nodes to a single, global, fresh constant (i.e., we considering all blank nodes as equal). This allows us to detect changes in documents, including additions and deletions of statements, irrespective of blank node labels. We compared this approximation to an isomorphism check for RDF graph equivalence and found that it corresponded in all pair-wise comparisons of document versions for both positive and negative cases.

The distribution of changes for the kernel documents across the 29 snapshots is plotted in Figure 5, where we see the ratio of documents with 0–28 changes across the snapshot pairs. At $x = 0$, we see that 62.2% of documents did not change over the 29 weeks. Thereafter, we see that most other documents changed infrequently or changed very frequently: 23.2% fell into the slightly dynamic $[1, 3]$ interval, 8.4% fell into the highly dynamic $[24, 28]$ interval, and 6.2% fell into the large remaining $[4, 23]$ middle interval.

Next, we are interested to characterise changes of documents within the same pay-level-domain. In Figure 6, we plot domains along two dimensions of change: the x-axis represents the ratio of documents on that domain that exhibited at least one change in the monitoring period, the y-axis represents the mean number of changes for the documents on that domain (including only those that

[5] See, e.g., `http://dbtune.org/artists/last-fm/Baracudas.rdf`; retr. 2013/02/12.

changed at least once), and the size of the tick indicates the number of sampled documents for the domain. We also annotate some examples of notable domains. Many domains sit at the origin indicating no changes in any document. Relatedly, since the majority of domains tend to cluster towards three of the four corners, we can consider the following classification of domains:

STATIC domains contain a low ratio of documents that change, and these documents change infrequently. Per Figure 6, 322 domains fell into the STATIC quadrant (**51.9%**), including linkedmdb.org, bbc.co.uk, w3.org, etc.

BULK domains contain a high ratio of documents that change, but these documents change infrequently. Per Figure 6, 182 domains fell into the BULK quadrant (**29.4%**), including dbpedia.org, freebase.com, bio2rdf.org, etc.

DUAL domains contain a low ratio of documents that change, but these documents change frequently. Per Figure 6, only 6 domains fell into the DUAL quadrant (**1.0%**), including loc.gov and geospecies.org.

ACTIVE domains contains a high ratio of documents that change, and these documents change frequently. Per Figure 6, 110 domains fell into the ACTIVE quadrant (**17.7%**), including dbtropes.org, dbtune.org, linkeddata.es, etc.

We highlight that for many of the BULK domains, although a large number of documents changed in the course of our observations, all changes for these domain tended to happen together: for such domains, the median number of weeks with changes was 4 (with no change on the domain between 24 weeks).

Based on meta-data from the LOD cloud and the DataHub[6], in Table 2, we show the breakdown of domains in the categories outlined above for (i) dataset topic, and (ii) whether the data is exported directly by the producer or by a third party. We could not locate topic or producer information for many (non-LOD) domains with few documents (cf. Table 2). Since domains may host multiple datasets, if we found multiple topics or production types associated to a single domain, we categorised it as cross-domain or *both*, respectively. In general, we see few high-level patterns in dynamicity for different topics or methods of production. Perhaps most notably, third-party exporters tend to be more active than first-party producers (presumably due to "live exporters"). Also, user-generated domains tended to be more active (though the number of such domains was low).

Discussion: We find that 62.2% of documents did not change in the 29 weeks and thus are obvious candidates for long-term caching. This compares with, e.g., 56% of static HTML pages reported by Brewington and Cybenko [1] in 2000, 65.5% reported by Fetterly et al. [4] in 2003 and 50% reported by Ntoulas et al. [9] in 2004. Such works also confirm that past dynamicity can be used to predict future dynamicity. Our work also clusters changes per domain, helping to design synchronisation strategies, where, e.g., a change detected for a BULK site such as dbpedia.org suggests that all documents from that domain should be refreshed. Similarly, Ntoulas et al. [9] showed that change predictions made for individual sites can often (but not always) be accurate.

[6] `http://lod-cloud.net`; `http://datahub.io/`; retr. 2013/03/08.

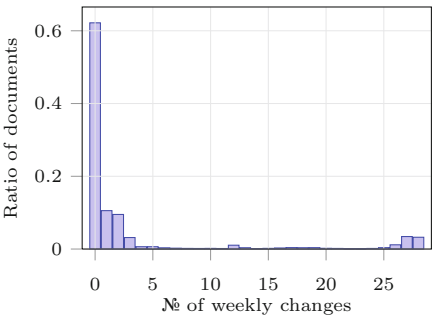


Fig. 5. Document change distribution

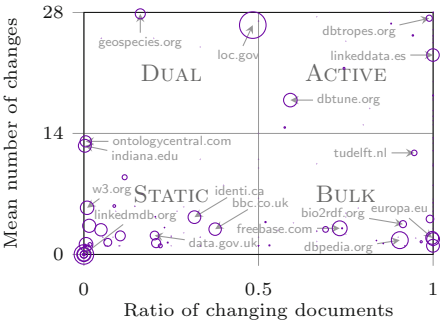


Fig. 6. Clustering of domain changes

Table 2. Dynamicity of Linked Data domains per topic and per party involved

Category	Doc №	Dom №	Static		Bulk		Dual		Active	
			№	%	№	%	№	%	№	%
cross-domain	34,872	33	21	63.64	6	18.18	2	6.06	4	12.12
geographic	4,693	10	6	60.00	2	20.00	1	10.00	1	10.00
government	5,544	14	10	71.43	3	21.43	0	0.00	1	7.14
life-sciences	2,930	4	2	50.00	2	50.00	0	0.00	0	0.00
media	8,104	10	6	60.00	2	20.00	0	0.00	2	20.00
publications	14,666	35	24	68.57	8	22.86	2	5.71	1	2.86
user-generated	7,740	12	7	58.33	0	0.00	0	0.00	5	41.67
unknown	8,147	502	246	49.00	159	31.67	1	0.20	96	19.12
first-party	22,649	50	38	76.00	8	16.00	2	4.00	2	4.00
third-party	29,078	61	37	60.66	12	19.67	1	1.64	11	18.03
both	27,520	23	13	56.52	6	26.09	2	8.70	2	8.70
unknown	7,449	486	234	48.15	156	32.10	1	0.21	95	19.55
total	86,696	620	322	51.94	182	29.35	6	0.97	110	17.74

5 RDF-Level Dynamics

We see that Linked Data documents change with varying degrees of breadth and frequency on different domains, and that documents on some domains, such as dbtropes.org, change widely and often. We now look at what kinds of changes are occurring on an RDF-level within these documents.

5.1 Types of Triple-Level Changes

We first look at the types of changes for documents. We found that 27.6% of documents only ever updated values for terms (one per triple) in the RDF graph they contain across the 29 weeks, keeping the number of triples static: such changes would include, e.g., updating a literal value like as an access-date entry. A further 24.0% of documents only added triples across the 29 weeks, representing monotonic additions. Changes for other documents involved a mix of additions, single-term updates and deletions across the different weeks.

In Figure 7, we plot the ratio of documents for which we found at least one triple addition over the 29 weeks against the ratio of documents for which we

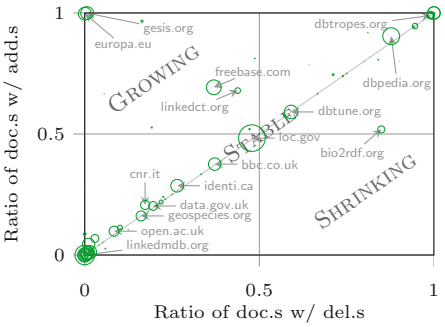


Fig. 7. Ratio of documents with additions vs. deletions per domain

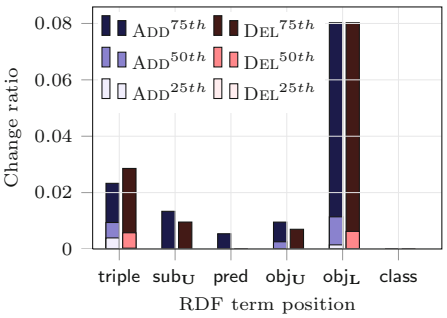


Fig. 8. Additions (left) and deletions (right) for different RDF elements

encountered some deletion over the 29 weeks, looking for high-level patterns. For the purposes of this plot, we consider a term update as an addition and a deletion of a triple. We see that most of the domains fall along a stable line where an equal number of documents involve some additions and some deletions: again, many of these documents correspond to the 27.6% that only update individual values (an add and a delete). Close to the $(0, 1)$ point, we see two large "monotonic" domains (europa.eu and gesis.org) that almost always only ever *add* triples to their documents. The one notable domain in the SHRINKING area was bio2rdf.org, for which 52% of documents had additions and 85% had deletions.

Discussion: For Linked Data warehouses, additions are often cheaper than deletions (esp. if, e.g., inference and truth maintenance are required). Here we see that additions to Linked Data documents are almost always accompanied by deletions, emphasising the importance of efficient revision strategies for warehouses. In relation to the HTML Web, Brewington and Cybenko [1] show that the content of HTML pages tends to grow over time, though their results rather reflect technological trends over a period of many years ('95–'99).

5.2 Types of Term-Level Changes

Next we look at the types of terms changing in the RDF content of the kernel. Figure 8 plots the 25^{th}, 50^{th} and 75^{th} percentiles[7] for the addition/deletion of RDF triples and the terms they contain. We only consider documents that changed at least once in the 29 weeks and omit blank node terms due to possible homomorphisms (relying on our approximation for *triples* involving blank nodes). We compare changes to subject URIs, predicates, object URIs, object literals and classes (values for `rdf:type`). The y-axis reflects the ratio of triples or terms that changed versus the total number of unique such elements observed in the documents considered (the y-range is small: $[0, 0.08]$). A ratio of 0.08 for object literal additions thus indicates that, over 29 weeks, the number of unique

[7] Higher percentiles cause too much compression of the results; hence we omit them.

object literals added to the documents at that percentile was 0.08× the total number of unique object literals appearing in those documents.

We see some clear trends. First, we see that additions and deletions are often of a similar magnitude, reflecting back on previous observations about terms often being directly replaced. Second, the most dynamic position of an RDF triple is the object, with a high churn of object literal values in particular. Conversely, predicates are occasionally added but rarely removed from documents. Analogously, class terms are very rarely added and very rarely removed (barely even seen above the x-axis). These latter two observations suggest that the *schema signature* of documents (set of property/class terms used) is generally static.

Discussion: The types of terms that change offer interesting high-level patterns into the dynamicity of RDF in general. For example, the observation that the set of properties and classes instantiated by a document rarely changes lends empirical strength to proposals for schema-level summaries of data, such as proposed by Stuckenschmidt et al. [11]. On the other hand, we see that literals are the most dynamic element of RDF. The following section sheds light on why this might be the case.

5.3 Dynamic Predicates

Though we have seen that predicates themselves are rarely added or removed, we are interested to see which predicates are indicative of dynamic statements. Table 3 presents the ten most dynamic predicates according to the ratio of added (+) and deleted (−) statements involving that predicate, divided by the total number of statements for that predicate across all snapshots; we only include predicates that appear in all snapshots and appear in $\geq 1,000$ statements overall.[8] Where added and deleted ratios correspond closely, this suggests frequent "value updates". The two `dbtont:` predicates are used on the third-party dptropes.com domain to indicate a time-stamp since the relevant data were parsed or fetched from the original source (tvtropes.org); the `swivt:`, `prv:` and `linkedct:` predicates also provide time-stamps indicating the last time data were refreshed for documents on various domains. The two `sioc:` predicates are used to track dynamic discussions and posts on the social gnoss.com domain. The `media:image` predicate appears for RDFa image meta-data, most of which are embedded in msn.com news pages. The `xhtml:bookmark` predicate represents links embedded as RDFa in various dynamic XHTML pages.

Discussion: Identifying dynamic predicates allows warehouses to know, in a granular fashion, which parts of an input query relate to static knowledge and which parts to dynamic knowledge (e.g., see our previous proposals on this topic [13]). Per our results, when considering cached content, the ratio of additions indicates the potential to miss answers involving triples with a given predicate, and the ratio of deletions indicates the potential to return stale answers. With respect to the most dynamic predicates, we identify that they are

[8] Prefixes can be found at `http://prefix.cc`; retr. 2013/03/12.

Table 3. Top-10 dynamic predicates ('*' indicates `provenance_time_updated` and `provenance_time_added`, respectively)

№	Predicate	Total	+	−
1	dbtont:parsed	35,911	0.94	0.94
2	sioc:has_discussion	3,171	0.87	0.99
3	sioc:content	107,387	0.87	0.98
4	dbtont:fetched	34,894	0.53	0.53
5	swivt:creationDate	35,295	0.53	0.53
6	media:image	1,377	0.49	0.49
7	prv:performedAt	16,706	0.45	0.45
8	xhtml:bookmark	17,852	0.45	0.44
9	linkedct:p.t.u*	2,652	0.42	0.42
10	linkedct:p.t.a*	2,652	0.42	0.42

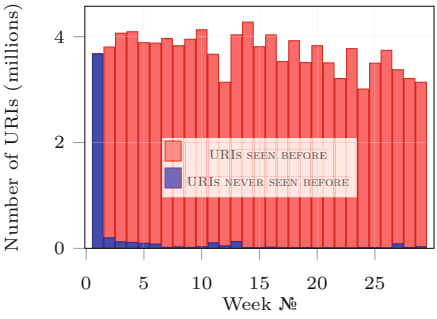


Fig. 9. Links extracted from kernels

often trivial time-stamps. Comparatively, Fetterly et al. [4] and Ntoulas et al. [9] both discuss how the majority of changes in HTML documents are very minor, involving hit counters, time-stamps, etc.

5.4 RDF Link Structure

Finally, we look at the evolving nature of the link structure of documents over time. We first want to see if the overall level of links tends to increase or decrease over time, and are interested to see at what rate fresh links are added to the kernel. We consider any URI in any position of a triple as a potential link from the kernel. Figure 9 plots the evolution of the volume of such links over time. We see that the number of links can fluctuate based on the availability of documents (with parallels to, e.g., response code distributions for each week illustrated in Figure 2). A second key observation is that the ratio of fresh URI links added to the kernel is in general quite small: we consider a URI as fresh if it has not been seen for *any* kernel snapshot before. This clearly indicates that the outward link structure of the kernel remains quite static (aside from instability) over time. In fact, if anything, links are on a decreasing trend as documents die off.

That said, after the initial stabilisation over the first month of observations, we do find that a few domains are consistently contributing some fresh links to the kernel: sec.gov, identi.ca, zitgist.com, dbtropes.org, ontologycentral.com and freebase.com offer a low-volume but constant stream of fresh outward links from week to week. Other domains—including bbc.co.uk, bnf.fr, dbpedia.org, linkedct.org, bio2rdf.org, etc.—tend to introduce new links in batches, corresponding with the update characteristics of domains plotted previously in Figure 6. However, such domains are the exception rather than the rule.

Discussion: Knowledge about how links change over time is important for any agent that traverses Linked Data documents (in terms of reachability, discoverability, etc.) or analyses link structure (e.g., to compute PageRank), etc. Ntoulas et al. [9] found that hyperlinks in HTML documents tend to be more dynamic than other forms of content, estimating that 25% of links are new each week (though considering a growing set of documents). In comparison, our results

seem much more static. This seems counter-intuitive in that Linked Data itself is fundamentally comprised of URIs and thus links; however, we already saw that URI terms in RDF documents change slowly (compared to, e.g., literals).

6 Conclusions

Six years on from the original publication of the Linked Data principles, we present some first concrete results on the dynamics of Linked Data documents.

Our first contribution is the design, implementation and upkeep of the Dynamic Linked Data Observatory. We have been running this observatory since May 2012 and have collected a significant corpus of data that captures the inherent dynamics of Linked Data. We will continue to run this experiment indefinitely, allowing us to draw further conclusions over the data. We make all data available for the community; please see `http://swse.deri.org/dyldo/` for up-to-date weekly snapshots. In the near future, we plan to extend this site with live statistics, meta-data and APIs.

Our second core contribution is the comprehensive analysis of the dynamics of Linked Data presented here. Based on monitoring 86,696 Linked Data documents for 29 weeks, we found that documents were unavailable 20% of the time and further estimated that 5% of documents had gone permanently offline in that period. We then determined that 62.2% of documents had no change in that time, where other documents either changed very frequently (8.4%), or very infrequently (23.2%), with few documents in between. Of the documents that did change, many updated individual RDF terms in the document (27.4%) or only ever added triples (23.1%). We found that domains tended to be either very static (44.5%), have a high ratio of documents that change infrequently (28.2%), or have a high ratio of documents that change frequently (25%); most domains contain a balance of documents with additions and deletions. With respect to the types of changes occurring on an RDF level, we found that object literals were the most liable to change ($0.01\times$ ratio for median/50^{th} percentile; $0.08\times$ for 75^{th} percentile), whereas the schema signature of documents—involving predicates and values for `rdf:type`—changed very infrequently. We identified predicates involved in the highest ratio of new/removed triples and found that they often relate to time-stamps. Finally, we showed that the rate of fresh links being added to the documents is low, varying between 4,960–126,944 depending on bulk domain updates that week.

In terms of connecting these observations back to our original use-cases outlined in Section 2, we make the following observations:

Focused Synchronisation: We identified the general rate of change of documents, and found that dynamicity tended to follow certain predictable patterns for PLDs. For example, static domains infrequently require light synchronisation, bulk domains occasionally require heavy synchronisation, dual domains require frequent light synchronisation, active domains require frequent heavy synchronisation (or live querying techniques), etc.

Smart Caching: Reversing the previous use-case, we found that 62.2% of documents didn't change over the six months and found that 51.9% of domains were considered static (and thus are candidates for long-term caching). Applications that rely on a schema-level index or schema-level cache of documents can rest assured that the schema-signature of documents tends to be very (though not completely) static. Furthermore, we identified particular predicates whose triples should not be cached due to high rates of updates.

Hybrid Architectures: A hybrid architecture could be built along a number of logical data partitions. First, we showed that domains tend to fall into a few clusters, where static and bulk domains could be supported by heavy materialisation approaches, whereas active domains are best supported through decentralised live-querying approaches. Conversely, we also showed, for example, that different schema patterns in the data were indicators of different levels of dynamicity, where partitioning could be done on a per-predicate basis instead, etc.

Link Maintenance: We found instability in documents, though much of this instability was of a temporary nature. However, we found that 5% of documents had died off during our monitoring period, suggesting an initial estimate for the arrival of deadlinks.

Versioning: We have not tackled the issue of versioning in depth. Some conclusions could be applied incidentally to the area of versioning (e.g., about the frequency of change of different types of RDF terms and the balancing of additions vs. deletions), but further more specialised analyses of the data (by us or the community) would be needed to generate concrete guidelines.

As highlighted by the last use-case, there is still further work to do.

7 Future Directions

In this paper, we focused on analysis of the kernel documents since they were retrieved through a consistent (and thus comparable) set of URIs. In future work, we would like to leverage the extended datasets, in particular to look at how often new RDF documents arise in the neighbourhood of the kernel.

A shortcoming of our current work is that we cannot say anything about changes at levels more fine-grained than a week. In our original proposal for the Dynamic Linked Data Observatory, we proposed to implement dynamic monitoring of documents that changed each week in increasingly more fine-grained intervals. We have yet to implement this feature, though this would give us knowledge of intra-week dynamics for (at least) a small number of highly-dynamic sources.

Finally, at some stage, we may need to consider an incremental extension of our kernel to include new Linked Data sources that are coming online. Our idea at the moment would involve infrequently adding 20% of fresh URIs on top of the kernel, possibly on a yearly basis. In general, we are open to extending our monitoring while maintaining the core kernel snapshots.

Acknowledgements. This paper was funded in part by Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Lion-2). We thank our anonymous reviewers (both LDOW and ESWC) for their helpful comments. We also thank Andrew Gallagher and Gerard Conneely at DERI, and the OpenCirrus team at KIT's Steinbuch Centre for Computing, for technical and infrastructural help.

References

1. Brewington, B., Cybenko, G.: Keeping up with the changing web. Computer 33(5), 52–58 (2000)
2. Cho, J., Garcia-Molina, H.: Estimating frequency of change. ACM Transactions on Internet Technology 3(3), 256–290 (2003)
3. Coffman Jr., E.G., Liu, Z., Weber, R.R.: Optimal robot scheduling for web search engines. Journal of Scheduling 1, 0–21 (1997)
4. Fetterly, D., Manasse, M., Najork, M., Wiener, J.L.: A large-scale study of the evolution of Web pages. In: WWW, pp. 669–678. ACM (2003)
5. Käfer, T., Umbrich, J., Hogan, A., Polleres, A.: DyLDO: Towards a Dynamic Linked Data Observatory. In: LDOW at WWW. CEUR-WS, vol. 937 (2012)
6. Ke, Y., Deng, L., Ng, W., Lee, D.L.: Web dynamics and their ramifications for the development of Web search engines. Computer Networks 50(10), 1430–1447 (2006)
7. Koehler, W.: An analysis of Web page and web site constancy and permanence. Journal of the American Society for Information Science 50(2), 162–180 (1999)
8. Lim, L., Wang, M., Padmanabhan, S., Vitter, J.S., Agarwal, R.: Characterizing Web document change. In: Wang, X.S., Yu, G., Lu, H. (eds.) WAIM 2001. LNCS, vol. 2118, pp. 133–144. Springer, Heidelberg (2001)
9. Ntoulas, A., Cho, J., Olston, C.: What's new on the Web? The evolution of the Web from a search engine perspective. In: WWW, pp. 1–12. ACM (2004)
10. Popitsch, N., Haslhofer, B.: DSNotify – a solution for event detection and link maintenance in dynamic datasets. J. Web Sem. 9(3), 266–283 (2011)
11. Stuckenschmidt, H., Vdovjak, R., Houben, G.-J., Broekstra, J.: Index structures and algorithms for querying distributed RDF repositories. In: WWW, pp. 631–639. ACM (2004)
12. Umbrich, J., Hausenblas, M., Hogan, A., Polleres, A., Decker, S.: Towards Dataset Dynamics: Change Frequency of Linked Open Data Sources. In: Proc. of LDOW at WWW. CEUR-WS, vol. 628 (2010)
13. Umbrich, J., Karnstedt, M., Hogan, A., Parreira, J.X.: Hybrid SPARQL Queries: Fresh vs. Fast Results. In: Cudré-Mauroux, P., et al. (eds.) ISWC 2012, Part I. LNCS, vol. 7649, pp. 608–624. Springer, Heidelberg (2012)

A Systematic Investigation of Explicit and Implicit Schema Information on the Linked Open Data Cloud

Thomas Gottron, Malte Knauf, Stefan Scheglmann, and Ansgar Scherp

WeST – Institute for Web Science and Technologies
University of Koblenz-Landau
56070 Koblenz, Germany
{gottron,mknauf,schegi,scherp}@uni-koblenz.de

Abstract. Schema information about resources in the Linked Open Data (LOD) cloud can be provided in a twofold way: it can be explicitly defined by attaching RDF types to the resources. Or it is provided implicitly via the definition of the resources' properties. In this paper, we present a method and metrics to analyse the information theoretic properties and the correlation between the two manifestations of schema information. Furthermore, we actually perform such an analysis on large-scale linked data sets. To this end, we have extracted schema information regarding the types and properties defined in the data set segments provided for the Billion Triples Challenge 2012. We have conducted an in depth analysis and have computed various entropy measures as well as the mutual information encoded in the two types of schema information. Our analysis provides insights into the information encoded in the different schema characteristics. Two major findings are that implicit schema information is far more discriminative and that applications involving schema information based on either types or properties alone will only capture between 63.5% and 88.1% of the schema information contained in the data. Based on these observations, we derive conclusions about the design of future schemas for LOD as well as potential application scenarios.

1 Introduction

Schema information of semantic data on the Linked Open Data (LOD) cloud is given in a twofold way: explicitly by providing the type of a resource and implicitly via the definition of its properties. These two manifestations of schema information are to a certain extent redundant, i.e., certain resource types entail typical properties and certain properties occur mainly in the context of particular types. For instance, we would expect a resource of type foaf:Person to have the properties foaf:name or foaf:age. Likewise, we can assume a resource with the property skos:prefLabel to be of type skos:Concept.

Schema information over LOD is used for various purposes such as indexing distributed data sources [10], searching in large graph databases [13], optimizing the execution of queries [14] or recommending appropriate vocabularies to linked data engineers [16]. Thus, it is an important question to which degree explicit and implicit schema information is correlated, i.e., to which extend the use of RDF types and properties appear together to describe resources. A high correlation of explicit and implicit

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 228–242, 2013.

schema information corresponds to redundant information—a fact which can be exploited, for instance, when indexing the LOD cloud and providing a central lookup table for LOD sources. One application in this context is the opportunity to compress a schema based index for LOD as motivated and requested by Neumann and Weikum [15]. More even, it is of interest, which schema information actually needs to be extracted from the Linked Open Data cloud and which information might be inferred[1]. Finally, a high correlation can be exploited directly for recommending typical combinations of types or properties when modelling Linked Data [16]. This leads us to the overall question to which extent the explicit schema information provided by RDF types coincides with the implicit schema information of the properties used in the LOD cloud and how consistent are the observed patterns and redundancies.

A fundamental prerequisite to answer this question is the availability of a reliable schema extracted from the LOD cloud that takes into account both explicit and implicit schema information. With the SchemEX approach [10,9], we can compute such a schema for huge amounts of RDF triples in an efficient manner. In the context of this paper, we describe a method and metrics for leveraging a schema obtained in this way to investigate the information theoretic properties and global dependencies between RDF types and properties. As the discussion of the related work in the subsequent section shows, such methods are—to the best of our knowledge—not available and an investigation as presented in this paper has not been done before. We will close this gap and consider for our analysis different data sets crawled from the LOD cloud and contained in the Billion Triples Challenge 2012 data set. The data sets cover data of different origin and quality and serve as basis for our experiments.

In Section 3, we will introduce a probabilistic schema distribution model. Based on this model, we identify different information theoretic metrics that are of interest. The metrics comprise different types of entropy as well as mutual information. In Section 4, we describe a method of how to estimate the relevant probabilities from a schema-based index and introduce the data sets we use for our analysis of linked data. The results of our investigation are shown in Section 5 where we also draw some conclusions regarding the design and application of future LOD schema. In summary, we have observed that the redundancy of explicit and implicit schema information on different parts of the LOD varied from 63.5% to 88.1%. Thus, a general schema for LOD should not be build on either explicit or implicit schema information only and should ideally integrate both types of information. Nevertheless, we also observed several highly indicative sets of properties, allowing a prediction of the types of resources.

2 Related Work

One application where schema information can be of value is query optimization. Neumann and Moerkotte [14] employ so-called *characteristic sets*, which basically classify RDF resources by the correlation of their (outgoing) predicate links. Knowledge about these sets allows for quite precise estimates of the result cardinality of join operations. Further insights into the correlation between properties in an RDF graph were not necessary. Neither were explicit schema information provided in form of RDF types

[1] Inference here can be realized in both ways: semantically or statistically.

considered. A similar approach is presented by Maduko et al. [13]. Here the focus was on efficient approaches to estimate subgraph frequencies in a graph database. This subgraph frequency information is then used for conducting efficient queries on the graph database. In their work, Maduko et al. use both implicit schema information and explicit schema information. However, they do not determine the cardinality of intermediate join results of the two schema information sources for executing the queries. Harth et al. [6] propose an approximative approach to optimize queries over multiple distributed LOD sources. They build a QTree index structure over the sources, which is used to determine the contribution of the single sources to the query results.

Several tools aim at providing statistics for the LOD cloud. LODStats [2] is a tool and framework for computing 32 different statistics on Linked Open Data such as those covered by the Vocabulary of Interlinked Data sets (VoID) [1]. The tool provides descriptive statistics such as the frequencies of property usage and datatype usages, the average length of literals, or counting the number of namespaces appearing at the subject URI position [2]. LODStats operates on single triple patterns, i.e., it does not provide statistics of, e.g., star patterns or other (arbitrary) graph patterns. However, it covers more complex schema-level characteristics like the RDFS subclass hierarchy depth [2]. Overall, analysis of the correlating use of different properties, RDF types, or the common appearance of properties and types like we investigate is out of scope. Also make-void[2] computes VoID-statistics for a given RDF file. These statistics usually contain information about the total number of triples, classes, properties, instances for each class, the uses of each property and the number of triples that link a subject on one domain to an object on another domain. Another framework for statistic generation on RDF data is RDFStats[3]. In contrast to make-void, RDFStats can also operate on SPARQL endpoints and uses a different vocabulary for its statistics.

Hogan et al. have conducted an empirical study to investigate the conformance of linked data sources with 14 different linked data principles [8]. As metric, the authors apply the number of unique namespaces used by the respective data providers and provide a ranked list in terms of top-5 and bottom-5 data providers. Among others, the authors analysed how different classes and properties of vocabularies defined at one data source are re-used and mixed by other linked data providers. In contrast, the analysis of the correlation of class terms and property terms of different (or the same) vocabularies done here is agnostic to the actual source the linked data originates from. Bizer et al. have recently analysed the joined occurrence of a single class with a single property on the structured data extracted from a large web crawl[4]. Lorey et al. [11] developed a frequent item set approach over properties for the purpose of detecting appropriate and diverging use of ontologies. None of these works addresses information theory metrics as it is done in the paper at hand. The application of information theoretic measures on RDF data is addressed in [12]. However, the analysis there is focussing on a different level of schema re-use of concepts and does not consider any property information.

[2] `https://github.com/cygri/make-void` (accessed 9 March 2013).
[3] `http://rdfstats.sourceforge.net/` (accessed 9 March 2013).
[4] `http://webdatacommons.org/` (accessed 9 March 2013).

3 Probabilistic Schema Model and Metrics

Schema information on the LOD cloud can be provided explicitly by the use of RDF type properties. There are no (practical) boundaries to the number of types that can be attached to a resource. In practice, we can observe resources which have no type as well as resources with several hundred types. In addition, schema information can be provided implicitly by the properties used to describe a resource. These properties connect one resource to another resource or a literal value. In this way, they implicitly describe the type of a resource by its relations. Again, it is possible to observe resources which have no relation (beyond a type description) as well as resources with hundreds of properties.

The goal of the analysis in this paper is to measure and quantify the information theoretic properties of the explicit schema information given by RDF types and the implicit schema information provided by the used properties. To this end, in Section 3.1 we first introduce a probabilistic model for the occurrence of types and properties of resources. This allows us to measure the schema information contained in types, properties or both together. In order to do so, we present different metrics such as entropy of marginal distributions, conditional entropy and mutual information in Section 3.2.

3.1 A Probabilistic Distribution Model for Types and Properties

We are interested in two observations about the resources on the LOD cloud: their types and their properties. To be more specific, we are interested in combinations of types and combinations of properties. A particular combination of types is a set of types attached to a resource. The space of all possible combinations therefore is the power set $\mathcal{P}(\mathit{Classes})$ of all class types in the data. While the power set itself is a huge set, we can actually restrict ourself to the subset $TS \subset \mathcal{P}(\mathit{Classes})$ of actually observed combinations of RDF types in the LOD cloud. For a given resource, we can now observe $t \in TS$ which corresponds to a set of types (e.g., the set $\{$foaf:Person, dbpedia:Politician$\}$).

Likewise, the properties observed for a resource is a combination of all possible properties. Accordingly here we deal with an element from the power set $\mathcal{P}(\mathit{Properties})$ of all observed properties. Again, we only need to consider the subset PS of actually occurred property sets. For an individual resource, we observe $r \in PS$ which corresponds to the set of its properties[5] (e.g., the set $\{$foaf:familyName, foaf:givenName, dbpedia:spouse$\}$).

To model the joint distribution of type sets and property sets, we introduce two random variables T and R. These take as values the elements in TS and PS, respectively. Both random variables are of discrete nature and their joint distribution can be characterized by:

$$P(T = t, R = r) = p(t, r) \tag{1}$$

where $p(t, r)$ is the probability for a randomly chosen resource to observe the concrete set t of attached types and the set r of properties. Based on this joint distribution, we can also identify the marginal distributions of T and R:

[5] Please note, we use the letter r for sets of properties (inspired by the term relation), as p will be used to denote probabilities.

$$P(T = t) = \sum_{r \in PS} p(t, r) \quad , \quad P(R = r) = \sum_{t \in TS} p(t, r) \tag{2}$$

3.2 Metrics of Interest

For analysing the LOD cloud, we are interested in several characteristics of the joint distribution $P(T, R)$ introduced above. The main questions that we want to answer are:

(a) How much information is encoded in the type set or property set of a resource on a global scale?
(b) How much information is still contained in the properties, once we know the types of a resource?
(c) How much information is still contained in the types, once we know the properties of a resource?
(d) To which degree can one information (either properties or types) explain the respective other?

To answer these questions, we introduce appropriate metrics that can be applied to the joint distribution of type sets and property sets. All our metrics are based on the *entropy* of probabilistic distributions [17], the standard concept to measure information.

Entropy of the Marginal Distributions. To answer the question of (a) how much information is encoded in the type or property set of a resource, we need to look at the marginal distributions. These provide us with the probability of a certain resource to show a particular set of types or properties. The entropy of the marginal distributions of T and R is defined as:

$$H(T) = -\sum_{t \in TS} P(T = t) \cdot \log_2 \left(P(T = t)\right) \tag{3}$$

$$H(R) = -\sum_{r \in PS} P(R = r) \cdot \log_2 \left(P(R = r)\right) \tag{4}$$

The values $H(T)$ and $H(R)$ give us an idea of how much information is encoded in the sets of types or properties of the resources. A higher value corresponds to more information, which in turn means that the sets of types or sets of properties appear more equally distributed. To be more concrete: an entropy value of 0 indicates that there is no information contained. For instance, a value of $H(T) = 0$ would indicate that all resources have exactly the same set of types (likewise for $H(R) = 0$). A maximal value, instead, is reached when the distribution is an equal distribution, i.e., each set of types or properties is equally probable. This fact also allows for normalizing the entropy values by:

$$H_0(T) = \frac{H(T)}{H^T_{\max}} = \frac{H(T)}{\log_2(|T|)} \quad , \quad H_0(R) = \frac{H(R)}{H^R_{\max}} = \frac{H(R)}{\log_2(|R|)} \tag{5}$$

The normalized entropy value ranges between 0 and 1 and indicates whether the distribution is closer to a degenerated or a uniform distribution.

Conditional Entropy. The question (b), how much information is still contained in the properties, once we know the types of a resource implies a conditional probability and, thus, a conditional entropy. We have to take a look at the distribution of the property sets given that we already know the types of a resource. The entropy in this case (i.e., the conditional entropy) conveys how much information is still in the additional observation of the properties. Again, if the set of types perfectly defines the set of properties to expect, there would be no more information to be gained. Thus, the conditional entropy would be zero. If, instead, the types were virtually independent from the properties, we would expect to observe the marginal distribution of the properties and its according entropy. Formally the conditional entropy for a given type set t is defined as:

$$H(R|T=t) = -\sum_{r \in PS} P(R=r|T=t) \log_2 \left(P(R=r|T=t)\right) \tag{6}$$

$$= -\sum_{r \in PS} \frac{p(t,r)}{P(T=t)} \log_2 \left(\frac{p(t,r)}{P(T=t)}\right) \tag{7}$$

Equivalently, to answer question (c), the conditional entropy for a given property set r is:

$$H(T|R=r) = -\sum_{t \in TS} \frac{p(t,r)}{P(R=r)} \log_2 \left(\frac{p(t,r)}{P(R=r)}\right) \tag{8}$$

These conditional entropies are fixed to one particular set of types t or set of properties r. As we are interested in a global insight of a large scale data set like the LOD cloud, it is not feasible to look at all the individual observations. Rather we need an aggregated value.

One value of particular interest is a conditional entropy of 0. For instance, in the case of $H(R|T = t) = 0$ knowing the set of types t is already conveying all the information, i.e. the set of properties can be derived with probability 1. Equivalently in the case of $H(T|R = r) = 0$ we can derive the set of types from the set of properties. Accordingly we are interested in the probability of such a conditional entropy of 0, e.g. $P(H(R|T = t) = 0)$ for the case of given type sets. Treating the conditional entropy itself as a random variable allows for easily estimating this probability by $P(H(R|T = t) = 0) = \sum_{H(R|T=t)=0} P(T = t)$.

Expected Conditional Entropy. A similar approach is taken for the expected conditional entropy $H(R|T)$. This aggregated value also considers the conditional entropy as a random variable and computes the expected values of this variable based on the probability to actually observe a certain set of types t. The definition of this aggregation is:

$$H(R|T) = \sum_{t \in TS} P(T=t) H(R|T=t) \tag{9}$$

$$= -\sum_{t \in TS} \sum_{r \in PS} p(t,r) \log_2 \left(\frac{p(t,r)}{P(T=t)} \right) \tag{10}$$

and equivalently $H(T|R)$ is for a given set of properties r:

$$H(T|R) = -\sum_{r \in PS} \sum_{t \in TS} p(t,r) \log_2 \left(\frac{p(t,r)}{P(R=r)} \right) \tag{11}$$

Joint Entropy. Finally, we will also take a look at the joint entropy of T and R, which is defined as:

$$H(T,R) = -\sum_{t \in TS} \sum_{r \in PS} p(t,r) \log_2 \left(p(t,r) \right) \tag{12}$$

Mutual Information. To finally answer the question of (d) how far one of the schema information (either properties or types) can explain the respective other, we employ mutual information (MI) [3]. MI is a metric to capture the joint information conveyed by two random variables – and thereby their redundancy. The MI of explicit and implicit schema information of the LOD cloud is defined as:

$$I(T,R) = \sum_{r \in PS} \sum_{t \in TS} p(t,r) \log_2 \frac{p(t,r)}{P(T=t) \cdot P(R=r)} \tag{13}$$

The $\log$ expression in this sum, i.e., the expression $\log_2 \frac{p(t,r)}{P(T=t) \cdot P(R=r)}$ is also known as *pointwise mutual information* (PMI). PMI can be explained as the strength of the correlation of two events, in our case how strongly a particular type set and a particular property set are associated with each other.

One characteristics of MI is the open range of its values. A normalization of MI to the interval $[-1, 1]$ is given in [18] and involves the entropy of the marginal distributions of T and R. It is used as a direct measure for redundancy and is defined as:

$$I_0(T,R) = \frac{I(T,R)}{\min\left(H(T), H(R)\right)} \tag{14}$$

4 Empirical Analysis of Linked Open Data

In the previous section, we have elaborated the metrics to obtain the relevant insights into the information and redundancy encoded in a LOD schema. In this section, we provide an approach to estimate the required probabilities from a SchemEX index structure, apply this approach to real world data and compute the metrics for our analyses.

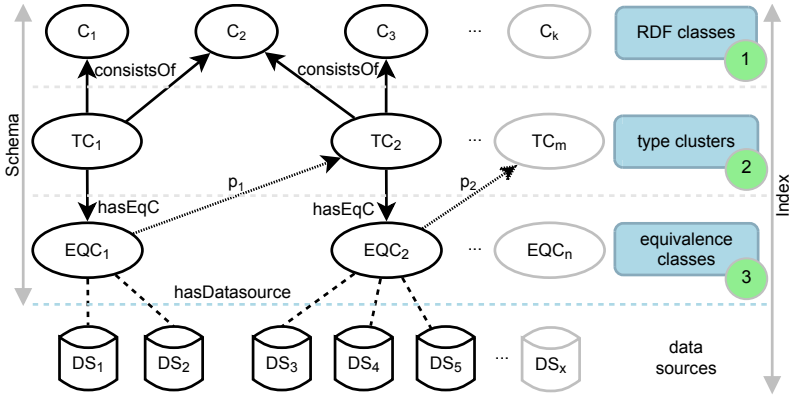


Fig. 1. SchemEX index structure with three layers leveraging RDF typings and property sets

4.1 The SchemEX Index as Basis for the Analysis

The purpose of SchemEX [9,10,5] is to link schema information to data sources which provide resources conforming to this schema element. Data sources are, e.g., static RDF documents and SPARQL endpoints [7]. The central schema elements of SchemEX are Typeclusters (TC) and Equivalence classes (EQC). A TC contains all data sources which provide resources conforming to a well defined set of types/classes. The EQC divide the data sources in each TC into disjoint subsets, defined by the set of properties the instances have and in which TC the object of the triple lies. An overview of the information contained in a SchemEX index is shown in Figure 1.

It is important to note that data sources can occur in several TC or EQC as they typically describe more than one and—in particular—different kinds of resources. However, different occurrences of a data source conform to different (in particular disjoint) sets of resources. Different data volume can be reflected by annotating the data sources attached to schema elements with the *number* of resources which exhibited the according schema information [5].

Noteworthy about SchemEX is, that it can be computed very efficiently and for large data sets using a stream-based approach. In this case, the analytical component is operating in a single pass fashion over a set of RDF triples. By using a windowing technique, it is possible to obtain a very accurate schema of the processed data using commodity hardware. However, the windowing technique entails a certain loss of schema information. The extent of this loss has been analysed in detail in [4]. The type of schema information and the metrics we use in the context of this paper are relatively stable. Deviations typically range up to 5%, in single cases differences of up to 10% have been observed in an empirical evaluation.

4.2 Estimating Probabilities from a SchemEX Index

The TC elements in SchemEX [9] described in Section 4.1 correspond directly to the notion of types sets in TS given in Section 3.1. The equivalence classes in SchemEX

subdivide the typeclusters and are defined by the set of properties the triples have as well as the typecluster the object of triple lies in. Hence, they are more finegrained than the property sets we are interested in. However, aggregating the equivalence classes defined by the same set of properties over all attached typeclusters, we obtain exactly the property sets PS introduced in Section 3.1. In this way we can easily construct the set PS from a SchemEX index.

As stated above, each entry in the SchemEX index refers to a distinct set of resources. Even if some of the resources are actually located in the same data source. This is provided by the pairwise disjoint character of equivalence classes. In conclusion, we can treat each entry in the index as a different set of resources, even if it is actually reflected by the same URL denoting a common data source.

If we denote with $\mathrm{DS}(t, r)$ the set of data source entries in the SchemEX index that correspond to the resources with types t and properties r, we can estimate the above probability of observing a resource to have a particular type and property set by:

$$\hat{p}(t, r) = \frac{\sum_{d \in \mathrm{DS}(t,r)} |d|}{N}$$

Where N is the number of all resources used to build the SchemEX and $|d|$ is the number of resources in data source d with the type set t and the property set r.

The estimates for the probabilities $p(t, r)$ above are central to all relevant metrics and effectively need only to be aggregated and normalized accordingly. However, the number of observed type sets and property sets indicates the high number of possible combinations (i.e., $|TS| \times |PS|$). The pragmatic solution to this quadratic development of combinations is not to compute all of the probabilities, but only those which actually have a non zero value. This does not affect the results of the computed metrics, as zero probabilities do not affect their overall values.

4.3 Data Sets

For our empirical analysis, we use the different segments of the data set provided for the Billion Triple Challenge (BTC) 2012. The BTC data set has been crawled from the web in a typical web spider fashion and contains about 1.44 billion triples. It is divided into five segments according to the set of URLs used as seed for the crawling process: Datahub, DBPedia, Freebase, Rest and Timbl. Details about the different parts and the crawling strategies used for collecting the data are described on the BTC 2012 data set's website[6]. As the efficient stream-based computation of a schema entails a certain loss of accuracy regarding the schema, we have to check that these inaccuracies do not affect the overall results. To this end, we use smaller data sets to compute the schema once with our stream-based approach and once in lossless approach and compare the metrics on these two schemas. As the computation of a gold standard schema has high requirements regarding the hardware resources, we were limited to derive lossless schema for data sets of up to 20 million triples. As small data sets, we used (A) the full *Rest* subset

[6] BTC 2012 data set: `http://km.aifb.kit.edu/projects/btc-2012/` (accessed 9 March 2013).

(22,328,242 triples), (B) an extract of the *Datahub* subset (20,505,209 triples) and (C) an extract of the *Timbl* subset (9,897,795 triples)[7].

The stream-based approach is also applicable to the full data crawls of (D) *Datahub*, (E) *DBPedia*, (F) *Freebase* and (G) *Timbl*. We used the same settings as in [9], using a window size of 50,000 instances for schema extraction. While the small data sets serve the purpose of confirming the stability of the stream-based approach, the larger data sets are used for the actual analysis of explicit and implicit schema information on the LOD cloud. We consider the data sets particularly useful as they span different aspects of the LOD cloud. With *Datahub*, we have got a sample of several publicly available linked RDF data sources registered in a central location. *DBpedia* is interesting as it is one of the central and most connected resources in the LOD cloud extracted from the collaboratively curated Wikipedia. *Freebase*, instead, is also a collaborative knowledge base, but here the users directly operate on the structural data. The *Timbl* data set is a crawl starting at the FOAF profile of Tim Berners-Lee (thus, the name). Hence, it provides a snapshot from yet a different part of the LOD cloud, namely starting at small, manually maintained RDF files.

5 Results of Our Analysis

Table 1 gives an overview of the count statistics and metric values obtained for the smaller data sets (A), (B) and (C). The table compares the values of the lossless gold standard schema computation with the efficient stream based approach. The observed deviations in the number of type sets in the data sets (A), (B) and (C) are very low and confirm the accuracy observed in previous experiments [4]. While for the data sets (B) and (C) also the number of property sets obtained by the stream-based approach does not differ much from the gold standard, we observed a slightly stronger deviation on the Rest (A) data set. The sheer count of type and property sets, however, does not reflect the number of data sources and resources behind the individual elements in the schema. Thus, it is necessary to consider the distributions and the metrics derived from those. Here, we observe a generally quite good behaviour of the efficient schema approximation using the stream-based approach. The differences in the metrics are relatively small and consistent within each data set. In conclusion, we decided that the loss of accuracy due to the efficient stream-based schema computation is counterbalanced by the capabilities to analyse data sets which are an order of magnitude larger: the observation of more data allows for a more sound evaluation of schema information on the LOD cloud.

Table 2 gives an overview of the computed metrics on the large data sets. Already the differences in the number of observed type and property sets underline the heterogeneity of the data sets. We will now go into the details of the single metrics.

Entropy in Type and Property Sets. We can observe the tendency that the property sets convey more information than type sets. This can be observed in the higher values of the normalized entropies. For instance, the normalized marginal entropy of the

[7] The extracts correspond to the data sets that would have been obtained by stopping the crawling process after 2 hops from the Datahub URI seed set and 4 hops from the Timbl URI seed set. We did not produce extracts for DBpedia and Freebase as the hop information is not provided for these BTC subsets.

Table 1. Statistics of the schema information obtained for the smaller data sets when using lossless and efficient (stream-based) schema computation

Data set		**(A) Rest**		**(B) Datahub (extract)**		**(C) Timbl (extract)**	
Number of Triples		22.3M		20.5M		9.9M	
Schema construction		lossless	efficient	lossless	efficient	lossless	efficient
Type sets	$\|T\|$	791	793	3,601	3,656	1,306	1,302
Property sets	$\|R\|$	8,705	7,522	4,100	4,276	3,015	3,085
Entropy of type sets	$H(T)$	2.572	2.428	3.524	3.487	2.839	2.337
Normalized entropy of type sets	$H_0(T)$	0.267	0.252	0.298	0.295	0.274	0.226
Entropy of property sets	$H(R)$	4.106	4.708	6.008	6.048	3.891	3.258
Normalized entropy of property sets	$H_0(R)$	0.314	0.366	0.501	0.501	0.337	0.281
Expected conditional entropy, given properties	$H(T\|R)$	0.295	0.289	1.158	1.131	0.670	0.512
Probability of $H(T\|R=r)=0$	$P(H(T\|R=r)=0)$	29.32%	38.02%	60.77%	57.79%	27.81%	21.52%
Expected conditional entropy, given types	$H(R\|T)$	1.829	2.568	3.643	3.692	1.723	1.433
Probability of $H(R\|T=t)=0$	$P(H(R\|T=t)=0)$	6.22%	5.31%	12.01%	11.08%	6.06%	4.51%
Joint entropy	$H(T,R)$	4.401	4.997	7.166	7.179	4.561	3.770
Mutual Information	$I(T,R)$	2.277	2.140	2.365	2.356	2.169	1.824
Normalized Mutual Information	$I_0(T,R)$	0.885	0.881	0.671	0.676	0.764	0.781

Table 2. Statistics of the schema information obtained for the full data sets when using efficient (stream-based) schema computation

Data set		**(A) Rest**	**(D) Datahub (full)**	**(E) DBpedia**	**(F) Freebase**	**(G) Timbl (full)**
Number of Triples		22.3M	910.1M	198.1M	101.2M	204.8M
Type sets	$\|T\|$	793	28,924	1,026,272	69,732	4,139
Property sets	$\|R\|$	7,522	14,712	391,170	162,023	9,619
Entropy of type sets	$H(T)$	2.428	3.904	1.856	2.037	2.568
Normalized marginal entropy of type sets	$H_0(T)$	0.252	0.263	0.093	0.127	0.214
Entropy of property sets	$H(R)$	4.708	3.460	6.027	2.868	3.646
Normalized entropy of property sets	$H_0(R)$	0.366	0.250	0.324	0.166	0.276
Expected conditional entropy, given properties	$H(T\|R)$	0.289	1.319	0.688	0.286	0.386
Probability of $H(T\|R=r)=0$	$P(H(T\|R=r)=0)$	38.02%	11.59%	54.85%	80.89%	15.15%
Expected conditional entropy, given types	$H(R\|T)$	2.568	0.876	4.856	1.117	1.464
Probability of $H(R\|T=t)=0$	$P(H(R\|T=t)=0)$	5.31%	10.83%	3.73%	2.05%	1.60%
Joint entropy	$H(T,R)$	4.997	4.779	6.723	3.154	4.032
Mutual Information	$I(T,R)$	2.140	2.585	1.178	1.751	2.182
Normalized Mutual Information	$I_0(T,R)$	0.881	0.747	0.635	0.860	0.850

property sets has a value of 0.324 on the *DBpedia* (E) data set, while the normalized marginal entropy of the type sets is 0.093. This observation provides a hint that on *DBpedia* the distribution into type sets is far more skewed than the distribution of property sets. Similar observations can be made for the data set (A), (F) and (G), though to a lower extent. An exception is the *Datahub* data set (D), where the distribution of resources in type sets and property sets seems comparable.

Conditional Entropies. Looking at the expected conditional entropies reveals some interesting insights. Recall that the aggregation we chose for the conditional entropy provides us with the expected entropy, given a certain type set or property set. We can see in Table 2 that the entropy given a property set tends to be far lower than the one when given a type set. In conclusion: knowing the properties of a resource in these cases already tells us a lot about the resource, as the entropy of the conditional distribution can be expected to be quite low. On the contrary, when knowing the type of a resource the entropy of the distribution of the property sets can be expected to be still relatively high (when compared to the entropy of the marginal distribution). We looked at the data more closely to investigate how often a given type set is already a clear indicator for the set of properties (and vice versa). This insight is provided by considering the probabilities $P(H(R|T = t) = 0)$ and $P(H(T|R = r) = 0)$ to observe a conditional entropy of 0. The most extreme case is the *Freebase* (F) data set, where for 80.89% of all resources it is sufficient to know the set of properties in order to conclude the set of types associated with this resource. Knowing, instead, the types of a resource conveys less information: only in 2.05% of the cases this is sufficient to predict the set of properties of a resource. Again, and with the exception of *Datahub* (D), the other data sets exhibit a similar trend. However, at very different levels: the probability of knowing the type set for a given property set ranges between 15.15% and 54.85%. The *Datahub* data set shows a far more balanced behaviour. Both probabilities $P(H(R|T = t) = 0)$ and $P(H(T|R = r) = 0)$ are at around 11%, confirming the particular form of this data set.

Mutual Information. Finally, the value of the normalized MI gives us insights on how much one information (either properties or types) explains the respective other. Also here, we observe a quite wide range from 0.635 on *DBpedia* (E) to 0.881 on *Rest* (A). Accordingly, extracting only type or only property information from LOD can already explain a quite large share of the contained information. However, given our observations a significant part of the schema information is encoded also in the respective other part. The degree of this additional information depends on the part of the LOD cloud considered. As a rule of thumb, we hypothesise that collaborative approaches without a guideline for a schema (such as *DBpedia*) tend to be less redundant than data with a narrow domain (*Timbl*) or some weak schema structure (*Freebase*).

Discussion of the Results. The observations on the large data sets provide us with insights into the form and structure of schema information on the LOD cloud. First of all, the distribution of type sets and property sets tend to have a relatively high normalized entropy. We can conclude that the structure of the data is not dominated by a few combinations of types or properties. Accordingly for the extraction of schema information,

we cannot reduce the schema to a small and fixed structure but need to consider the wide variety of type and property information. Otherwise the schema would loose too much information.

A second observation is the dependency between types and properties. The conditional entropy reveals that the properties of a resource usually tell much more about its type than the other way around. This observation is interesting for various applications. For instance, suggesting a data engineer the types of a resource based on the already modelled properties seems quite promising. We assume that this observation can also be seen as an evidence that property information on the LOD cloud actually considers implicit or explicit agreements about the domain and range of the according property. However, this observation is not valid for the entire LOD cloud. Depending on the concrete setting and use case, a specific analysis might need to be run.

Finally, the observed MI values underline the variance of schema information in the LOD cloud. Ranges from 63.5% to 88.1% redundancy between the type sets and property sets have been observed. Thus, approaches building a schema only over one of these two types of schema information run at the risk of a significant loss of information.

6 Conclusions and Future Work

In this paper, we have proposed a method and metrics for conducting in depth analysis of schema information on Linked Open Data. In particular, we have addressed the question of dependencies between the types of resources and their properties. Based on the five segments of the BTC 2012 data set, we have computed various entropy metrics as well as mutual information. In conclusion, we observe a trend of a reasonably high redundancy between the types and properties attached to resources. As more detailed conclusion, we can derive that the properties of a resource are rather indicative for the type of the resource. In the other direction, the indication is less strong. However, this observation is nor valid for all sources on the LOD cloud. In conclusion, if the application and data domain is not known, it is necessary to capture both: explicit and implicit schema information.

As future work, we plan to deepen these insights and incorporate the obtained deeper understanding into various applications. Therefore, we will look into the details of the conditional distributions for given type sets and property sets. In this way, we might identify which sets of types and properties allow for highly precise predictions of the respective other schema information. On the application side, we plan to use the gained insights for various purposes: index compression for SchemEX as well as the detection of schema patterns that are stable enough—and thereby suitable—for constructing an API for accessing LOD resources.

Acknowledgements. The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 257859, ROBUST.

References

1. Alexander, K., Cyganiak, R., Hausenblas, M., Zhao, J.: Describing linked datasets with the void vocabulary, http://www.w3.org/TR/void/ (accessed March 9, 2013)
2. Auer, S., Demter, J., Martin, M., Lehmann, J.: LODStats – an extensible framework for high-performance dataset analytics. In: ten Teije, A., Völker, J., Handschuh, S., Stuckenschmidt, H., d'Acquin, M., Nikolov, A., Aussenac-Gilles, N., Hernandez, N. (eds.) EKAW 2012. LNCS, vol. 7603, pp. 353–362. Springer, Heidelberg (2012)
3. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley-Interscience (1991)
4. Gottron, T., Pickhardt, R.: A detailed analysis of the quality of stream-based schema construction on linked open data. In: CSWS 2012: Proceedings of the Chinese Semantic Web Symposium (2012) (to appear)
5. Gottron, T., Scherp, A., Krayer, B., Peters, A.: Get the google feeling: Supporting users in finding – relevant sources of linked open data at web-scale. In: Semantic Web Challenge, Submission to the Billion Triple Track (2012)
6. Harth, A., Hose, K., Karnstedt, M., Polleres, A., Sattler, K.U., Umbrich, J.: Data summaries for on-demand queries over linked data. In: WWW, pp. 411–420. ACM (2010)
7. Heath, T., Bizer, C.: Linked Data: Evolving the Web Into a Global Data Space. Synthesis Lectures on the Semantic Web: Theory and Technology. Morgan & Claypool (2011)
8. Hogan, A., Umbrich, J., Harth, A., Cyganiak, R., Polleres, A., Decker, S.: An empirical survey of linked data conformance. Web Semantics: Science, Services and Agents on the World Wide Web 14, 14–44 (2012)
9. Konrath, M., Gottron, T., Scherp, A.: Schemex – web-scale indexed schema extraction of linked open data. In: Semantic Web Challenge, Submission to the Billion Triple Track (2011)
10. Konrath, M., Gottron, T., Staab, S., Scherp, A.: Schemex—efficient construction of a data catalogue by stream-based indexing of linked data. Web Semantics: Science, Services and Agents on the World Wide Web 16, 52–58 (2012); The Semantic Web Challenge 2011
11. Lorey, J., Abedjan, Z., Naumann, F., Böhm, C.: Rdf ontology (re-) engineering through large-scale data mining. In: Semantic Web Challenge (2011)
12. Luo, X., Shinavier, J.: Entropy-based metrics for evaluating schema reuse. In: Gómez-Pérez, A., Yu, Y., Ding, Y. (eds.) ASWC 2009. LNCS, vol. 5926, pp. 321–331. Springer, Heidelberg (2009)
13. Maduko, A., Anyanwu, K., Sheth, A., Schliekelman, P.: Graph summaries for subgraph frequency estimation. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 508–523. Springer, Heidelberg (2008)
14. Neumann, T., Moerkotte, G.: Characteristic sets: Accurate cardinality estimation for rdf queries with multiple joins. In: Proceedings of the 27th International Conference on Data Engineering, ICDE 2011, Hannover, Germany, April 11-16, pp. 984–994 (2011)
15. Neumann, T., Weikum, G.: Scalable join processing on very large rdf graphs. In: SIGMOD Conference, pp. 627–640. ACM (2009)
16. Schaible, J., Gottron, T., Scheglmann, S., Scherp, A.: LOVER: Support for Modeling Data Using Linked Open Vocabularies. In: LWDM 2013: 3rd International Workshop on Linked Web Data Management (to appear, 2013)
17. Shannon, C.: A mathematical theory of communication. Bell System Technical Journal 27, 379–423, 623–656 (1948)
18. Yao, Y.Y.: Information-theoretic measures for knowledge discovery and data mining. In: Karmeshu (ed.) Entropy Measures, Maximum Entropy Principle and Emerging Applications. STUDFUZZ, vol. 119, pp. 115–136. Springer, Heidelberg (2003)

Lightweight Spatial Conjunctive Query Answering Using Keywords*

Thomas Eiter, Thomas Krennwallner, and Patrik Schneider

Institut für Informationssysteme, Technische Universität Wien
Favoritenstraße 9-11, A-1040 Vienna, Austria
{eiter,tkren,patrik}@kr.tuwien.ac.at

Abstract. With the advent of publicly available geospatial data, ontology-based data access (OBDA) over spatial data has gained increasing interest. Spatio-relational DBMSs are used to implement geographic information systems (GIS) and are fit to manage large amounts of data and geographic objects such as points, lines, polygons, etc. In this paper, we extend the Description Logic DL-Lite with spatial objects and show how to answer spatial conjunctive queries (SCQs) over ontologies—that is, conjunctive queries with point-set topological relations such as *next* and *within*—expressed in this language. The goal of this extension is to enable an off-the-shelf use of spatio-relational DBMSs to answer SCQs using rewriting techniques, where data sources and geographic objects are stored in a database and spatial conjunctive queries are rewritten to SQL statements with spatial functions. Furthermore, we consider keyword-based querying over spatial OBDA data sources, and show how to map queries expressed as simple keyword lists describing objects of interest to SCQs, using a meta-model for completing the SCQs with spatial aspects. We have implemented our lightweight approach to spatial OBDA in a prototype and show initial experimental results using data sources such as Open Street Maps and Open Government Data Vienna from an associated project. We show that for real-world scenarios, practical queries are expressible under meta-model completion, and that query answering is computationally feasible.

1 Introduction

By the ever increasing availability of mobile devices, *location-aware* search providers are becoming increasingly commonplace. Search providers (e.g., Google Maps `http://maps.google.com/` or Nokia Maps `http://here.net`) offer the possibility to explore their surroundings for desired locations, also called *points-of-interest (POIs)*, but usually miss the possibility to express spatial relations (e.g., *next* and *within*). For an expressive location-aware search, the combination of Semantic Web techniques and spatial data processing (with spatial relations) is appropriate, given they provide a data backbone for spatial and taxonomic information to query semantically-enriched POIs.

To realize location-aware semantic search support, one needs to capture *categories* of POIs (e.g., Italian restaurant), their relations to additional *qualitative attributes* (e.g.,

* Supported by the Austrian Research Promotion Agency (FFG) project P828897, and the Austrian Science Fund (FWF) project P20840.

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 243–258, 2013.

$$
\begin{array}{rcl}
Shop \sqsubseteq SpatialFeat & hasOp \sqsubseteq hasQVal & Op \sqsubseteq QVal \\
\exists hasQVal^- \sqsubseteq SpatialFeat & Shop \sqsubseteq \exists hasOp & Wlan \sqsubseteq QVal \\
Park \sqsubseteq SpatialFeat & \exists hasOp^- \sqsubseteq Op & GuestGarden \sqsubseteq QVal \\
Supermarket \sqsubseteq Shop & QVal \sqsubseteq \exists hasQVal & SpatialFeat \sqsubseteq \neg Geometry \\
Walmart \sqsubseteq Op & &
\end{array}
$$

Fig. 1. Ontology with integrated meta-model (TBox excerpt; role names start lowercase)

having a guest garden). Further, one needs to capture the *spatial relations between POIs* (e.g., located inside a park). For modeling and interpreting a user's intention, it seems suggestive to use ontology languages and associated reasoning services. However, for spatial aspects we need to extend or combine them with spatial data reasoning. Furthermore, we must respect that ordinary users of location-aware search need a plain query interface; they are not experts in query languages, and an interface to express search intentions by lists of *keywords* in a Google-like manner would be desirable.

However, we face several obstacles for a seamless keyword-based querying and integration of geospatial data sources and ontologies. First, for a meaningful search result, we need to consider data obtained by integrating multiple data sources, which may be provided by autonomous vendors in heterogeneous formats (e.g., OpenStreetMap or Open Government Data data, a restaurant guide, etc). Using various data sources of substantial size gives the opportunity to find intended POIs, which may fall into multiple concepts ranging from rather generic to more detailed ones such as "restaurant" vs. "pizzeria." Moreover, we can exploit the structure of the taxonomic information that is implicit in the data sources by making it concrete in an ontology. Such *ontology-based data access* can be used to answer broad queries like "restaurants with Italian Cuisine," that should return pizzerias, trattorias, and osterias.

Second, from keyword-based input, we must generate meaningful formal queries to an ontology. In that, we must respect that the users may have no prior knowledge of the domain. Thus, we must be able to recognize and generate *relevant* combinations of possible keywords according to the ontology that represents the domain.

Third, as we query mainly spatial data, we need to capture spatial relations between different spatial objects and give users the possibility to use a fixed set of keywords to express them. For spatial querying answering, we must define an appropriate semantics and provide techniques that combine spatial with ontological query answering.

Fourth, a lot of research has been put into efficient query answering techniques over *lightweight ontology languages*, such as the DL-Lite family [7]. Conjunctive query (CQ) evaluation over DL-Lite ontologies can be delegated, by *first-order query rewriting*, to a Relational Database Management System (RDBMS), which facilitates scalable query processing. To secure this for an extension with spatial reasoning, the first-order rewritability of the latter is desirable. Furthermore, as first-order rewritings of queries might get prohibitively large in general (a known feature), also issues of manageable query generation from keywords must be respected.

We address the above issues with the following approach outlined in a nutshell.

- Various data sources are integrated via a global schema represented by an DL-Lite$_R$ ontology that is enriched with spatial information. The ontology-based knowledge base

(KB) is separated into a TBox, an ABox with *normal* individuals and a spatio-relational database with *spatial objects*. We apply a mild extension to DL-Lite$_R$ by associating individuals to spatial objects by a predefined binding. A preprocessor creates this binding using a domain-specific heuristic (which is not considered here).

- The enriched ontology can be accessed, at the system level, by *spatial conjunctive queries (SCQ)*, which extend conjunctive queries with spatial predicates (e.g. intersects). In such queries, individuals can be located with spatial objects whose relationships are determined. By rewriting techniques, and in exploiting the $PerfectRef$ algorithm [7], SCQs can be rewritten to a union of conjunctive queries (UCQ). Under certain syntactic conditions, a 2-stage evaluation—evaluation of the ontology part of the query (over the ABox, which is stored in an RDBMS) followed by filtering via spatial conditions—is possible, which makes this approach attractive for practical realization.
- For keyword-based query answering, concepts of the ontology are labeled with keywords. On query evaluation, the keywords which the user enters are mapped to concepts and roles from the ontology; an *auto-completion* service aids the user to compensate lack of domain knowledge. Based on the keyword structure, a feasible CQ is generated and extended with spatial predicates to SCQs; in that, we use a specific *meta-model* that is stored in the ontology. Fig. 1 shows an excerpt of the ontology; the concept $SpatialFeat$ intuitively says that the individual has spatial features, which is extended by the subroles of $hasQVal$ with qualitative values, which are asserted to subconcepts of $QVal$. Furthermore, the individual is represented by a geometry, asserted to subconcepts of $Geometry$. However, also normal role assertions for qualitative attributes are considered (e.g., a restaurant with a guest garden).

We have implemented this approach in an experimental prototype, which is part of a more extensive system for smart, semantically enriched route planning system (MyITS, `http://myits.at/`) over real world data sources such as OpenStreetMap (OSM), Open Government Data (OGD) of Vienna, and the *Falter* restaurant guide for Vienna. The data sources are integrated by a global schema represented by an ontology expressed in DL-Lite$_R$. It turns out that naively generated UCQs may be too large for execution on conventional RDBMS. We thus improved our approach by exploiting the structure of the TBox in an *optimized* generation of queries from keyword, to eventually obtain smaller UCQs. First experiments show that this approach is feasible in a real-world scenario. Furthermore, we show that the optimizations described are important for feasibility. An extended version of this paper provides more details that are omitted for space reasons.[1]

2 Preliminaries

We adopt DL-Lite$_R$ [7] as the underlying ontological language and introduce an approach in which the FO-rewriting of $PerfectRef$ (see [7] and [6] for details) is strictly separated from spatial querying. As a result of this separation, we only allow spatial predicates (e.g., $Contains$) on the top level of the query structure. Regarding the semantics, we following partly the ideas of [15], but focus primarily on query answering (not solely satisfiability). Furthermore, we use a different notion for spatial relations.

[1] `http://www.kr.tuwien.ac.at/staff/patrik/ESWC2013Ext.pdf`

Point-Set Topological Relations. We follow the point-set topological relation model in [13], where spatial relations are defined in terms of pure set theoretic operations. The realization of spatial objects is based on a set $P_E \subseteq \mathbb{R}^2$ of points in the plane; the (names of) spatial objects themselves are in a set Γ_S. While the set of points for a spatial object s is infinite (unless it is a point), it can be finitely defined by an associated *admissible geometry* $g(s)$. The geometries are defined by sequences $p = (p_1, \ldots, p_n)$ of points that induce a point ($n = 1$), a line segment ($n = 2$), a line ($n > 2$), or a polygon. All points used for admissible geometries are from a finite set $P_F \subseteq P_E$ of points.

Spatio-relational Database. Thus, we define a *spatio-relational database* over Γ_S as a pair $\mathcal{S} = (P_F, g)$ of a point set $P_F \subseteq \mathbb{R}^2$ and a mapping $g : \Gamma_S \to \bigcup_{i \geq 1} {P_F}^i$.

The extent of a geometry p (full point set) is given by the function $points(p)$ and is a (possibly infinite) subset of P_E. For a spatial object s, we let $points(s) = points(g(s))$. We need $points$ to evaluate the spatial relations of two spatial objects via their respective geometries. For our spatio-thematic KBs, we consider the following types of admissible geometries p over P_F (with their representation), and let $P_E = \bigcup_{s \in \Gamma_S} points(s)$: a

- *point* is a sequence $p = (p_1)$, where $points(p_1) = \{p_1\}$;
- *line segment* is a sequence $p = (p_1, p_2)$, and $points(p) = \{\alpha p_1 + (1 - \alpha)p_2 \mid \alpha \in \mathbb{R}, 0 \leq \alpha \leq 1\}$;
- *line* is a sequence $p = (p_1, \ldots, p_n)$ of line segments (p_i, p_{i+1}), $1 \leq i < n$, the first (p_1, p_2) and last (p_{n-1}, p_n) segments do not share an end-point, and $points(p) = \bigcup_{i=1}^{n-1} points(p_i)$;
- *polygon* is like a line but (p_1, p_2) and (p_{n-1}, p_n) share an end point; we have $points(a) = \bigcup_{i=1}^{n-1} points(p_i) \cup int(l_c)$, where $int(l_c)$ is the interior built from the separation of P_E by p into two disjoint regions.

Some $s \in \Gamma_S$ may serve to define via g a bounding box. We omit more complex geometries like areas or polygons with holes. Based on $points(x)$, we can define the spatial relation of point-sets in terms of *pure set operations*:

- $Equals(x, y)$: $points(x) = points(y)$ and $NotEquals(x, y)$: $points(x) \neq points(y)$;
- $Inside(x, y)$: $points(x) \subseteq points(y)$ and $Outside(x, y)$: $points(x) \cap points(y) = \emptyset$;
- $Intersect(x, y)$: $points(x) \cap points(y) \neq \emptyset$ and $NextTo(x, y)$: $b(x) \cap b(y) \neq \emptyset$, where $b(z) = \{a \in P_E \mid dist(a, points(z)) \leq d_B\}$ for a distance function $dist$: $\mathbb{R}^2 \to \mathbb{R}_0^+$ and a distance value $d_B \in \mathbb{R}$.

Now for any spatial relation $S(s, s')$ and $s, s' \in \Gamma_S$, holds on a spatio-relational DB $\mathcal{S}$, written $\mathcal{S} \models S(s, s')$, if $S(g(s), g(s')$ evaluates to true. Relative to $points$ and $dist$ (and d_B), this is easily captured by a first-order formula over $(\mathbb{R}^2, \leq)$, and with regard to geo-spatial RDBMS trivially first-order expressible.

Note that the space model of [13] differs from the more detailed *9-Intersection model* (DE-9IM) of [10], which considers strict separation of the interior and object boundary; this leads to 9 instead of 5 spatial relations. We also omit spatial predicates in the signature, assuming a "standard" point-set interpretation of the spatial-relations [13]. Our approach is modular and flexible enough to allow further relations (e.g., $connects$) or use other interpretations such as DE-9IM.

Syntax and Semantics of DL-Lite$_R$. We recall the definitions from [7]. Consider a vocabulary of individual names Γ_I, *atomic concepts* Γ_C, and *atomic roles* Γ_R. Given atomic concepts A and atomic roles P, we define *basic* concepts B and *basic* roles R, *complex* concepts C and *complex* role expressions E, and P^- be the *inverse* of P as

$$B ::= A \mid \exists R \qquad C ::= B \mid \neg B \qquad R ::= P \mid P^- \qquad E ::= R \mid \neg R \ .$$

A DL-Lite$_R$ *knowledge base* is a pair $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ where the TBox $\mathcal{T}$ consists of a finite set of *inclusion assertions* of the form $B \sqsubseteq C$ and $R \sqsubseteq E$, and the ABox $\mathcal{A}$ is a finite set of *membership assertions* on atomic concepts and on atomic roles of the form $A(a)$ and $P(a, b)$, where a and b are individual names.

The semantics of DL-Lite$_R$ is given in terms of FO interpretations $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, where $\Delta^\mathcal{I}$ is a nonempty domain and $\cdot^\mathcal{I}$ an *interpretation function* such that $a^\mathcal{I} \in \Delta^\mathcal{I}$ for all $a \in \Gamma_I$, $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$ for all $A \in \Gamma_C$, $P^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$ for all $P \in \Gamma_R$, and $(P^-)^\mathcal{I} = \{(a_2, a_1) \mid (a_1, a_2) \in P^\mathcal{I}\}$; $(\exists R)^\mathcal{I} = \{a_1 \mid \exists a_2 \in \Delta^\mathcal{I} \text{ s.t. } (a_1, a_2) \in R^\mathcal{I}\}$; $(\neg B)^\mathcal{I} = \Delta^\mathcal{I} \setminus B^\mathcal{I}$; and $(\neg R)^\mathcal{I} = \Delta^\mathcal{I} \times \Delta^\mathcal{I} \setminus R^\mathcal{I}$.
The notions of satisfaction of inclusion axioms and assertions, TBox and ABox resp. knowledge base is as usual, as well as logical implication; both are denoted with $\models$. We assume the *unique name assumption* holds for different individuals and values.

Checking satisfiability of DL-Lite$_R$ ontologies is *first-order (FO) rewritable* [7], i.e., for all $\mathcal{T}$, there is a Boolean FO query $Q_\mathcal{T}$ (constructible from $\mathcal{T}$) s.t. for every $\mathcal{A}$, $\mathcal{T} \cup \mathcal{A}$ is satisfiable iff $DB(\mathcal{A}) \not\models Q_\mathcal{T}$, where $DB(\mathcal{A})$ is the *least Herbrand model* of $\mathcal{A}$.

3 DL-Lite$_R(S)$

In this section, we extend DL-Lite$_R$ with spatial objects to DL-Lite$_R(S)$. We present its syntax and semantics, a transformation of to DL-Lite, and show that satisfiability and conjunctive query answering over DL-Lite$_R(S)$ KBs are FO-rewritable.

Syntax. Let Γ_S and Γ_I be pairwise disjoint sets as defined above. A *spatio-thematic knowledge base* (KB) is defined as $\mathcal{L}_\mathcal{S} = \langle \mathcal{T}, \mathcal{A}, \mathcal{S}, \mathcal{B} \rangle$, where $\mathcal{T}$ (resp. $\mathcal{A}$) is defined as a DL-Lite$_R$ TBox (resp. ABox), $\mathcal{S}$ is a spatio-relational database, and $\mathcal{B} \subseteq \Gamma_I \times \Gamma_S$ is a partial function called the *binding from $\mathcal{A}$ to $\mathcal{S}$*, similar to [15]; we apply a mild extension to DL-Lite$_R$ by associating individuals from $\mathcal{A}$ to spatial objects from $\mathcal{S}$.

We assume $\mathcal{B}$ to be already given. Furthermore, we extend DL-Lite$_R$ with the ability to specify the *localization* of a concept. For this purpose, we extend the syntax with

$$C ::= B \mid \neg B \mid (loc\ A) \mid (loc_s\ A), \quad s \in \Gamma_S,$$

where A is an atomic concept in $\mathcal{T}$; intuitively, $(loc\ A)$ is the set of individuals in A that can have a spatial extension, and $(loc_s\ A)$ is the subset which have extension s.

Semantics. Our aim is to give a semantics to the localization concepts $(loc\ A)$ and $(loc_s\ A)$ such that a KB $\mathcal{L}_\mathcal{S} = \langle \mathcal{T}, \mathcal{A}, \mathcal{S}, \mathcal{B} \rangle$ can be readily transformed into an ordinary DL-Lite$_R$ KB $\mathcal{K}_\mathcal{S} = \langle \mathcal{T}', \mathcal{A}' \rangle$, using concepts $C_{\mathcal{S}_\mathcal{T}}$ and C_s for individuals with some spatial extension resp. located at s. Note that $C_{\mathcal{S}_\mathcal{T}}$ cannot be forced to be the union of all C_s, as this would introduce disjunction (this hinders the passing from the open to the closed world assumption, which is important for the FO-rewriting of DL-Lite).

An (DL-Lite$_R$) *interpretation of* $\mathcal{L}_\mathcal{S}$ is a structure $\mathcal{I}=\langle \Delta^\mathcal{I}, \cdot^\mathcal{I}, b^\mathcal{I} \rangle$, where $\langle \Delta^\mathcal{I}, \cdot^\mathcal{I} \rangle$ is an interpretation of $\langle \mathcal{T}, \mathcal{A} \rangle$, and $b^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Gamma_S$ is a partial function that assigns some individuals a location, such that for every $a \in \Gamma_I$, $(a, s) \in \mathcal{B}$ implies $b^\mathcal{I}(a^\mathcal{I}) = s$.

We extend the semantics of the previous section with $(loc\ A)$, $(loc_s\ A)$, where A is an atomic concept in $\mathcal{T}$:

$$(loc\ A)^\mathcal{I} \supseteq \{e \in \Delta^\mathcal{I} \mid e \in A^\mathcal{I} \wedge \exists s \in \Gamma_S : (e, s) \in b^\mathcal{I}\},$$
$$(loc_s\ A)^\mathcal{I} = \{e \in \Delta^\mathcal{I} \mid e \in A^\mathcal{I} \wedge (e, s) \in b^\mathcal{I}\}.$$

The interpretation of complex concepts, satisfaction, etc. is then as usual. For example, $A \sqsubseteq (loc_s\ A)$ expresses that all individuals in A are located at s; $B \sqsubseteq (loc\ A)$ states that individuals in B can have a location if they are in A.

Transformation to DL-Lite$_R$. Let $C_{\mathcal{S}_\mathcal{T}}$ and C_s, for every $s \in \Gamma_S$, be fresh concepts. We transform $\mathcal{L}_\mathcal{S}$ to $\mathcal{K}_\mathcal{S} = \langle \mathcal{T}', \mathcal{A}' \rangle$, where $\mathcal{T}' = \tau(\mathcal{T}) \cup \mathcal{T}_\mathcal{S}$ and $\mathcal{A}' = \tau(\mathcal{A}) \cup \mathcal{A}_\mathcal{B}$, and

- $\tau(X)$ replaces each occurrence of $(loc\ A)$ and $(loc_s\ A)$ in X with $C_{\mathcal{S}_\mathcal{T}} \sqcap A$ and $C_s \sqcap A$, respectively, and splits $\sqcap$ up;
- $\mathcal{T}_\mathcal{S}$ represents generic localization information via concepts, and contains the axiom $C_s \sqsubseteq C_{\mathcal{S}_\mathcal{T}}$, and the constraints $C_s \sqsubseteq \neg C_{s'}$ for all $s \neq s' \in \Gamma_S$;
- $\mathcal{A}_\mathcal{B}$ represents the concrete bindings between $\mathcal{A}$ and Γ_S, and for every $(a, s) \in \mathcal{B}$, we add $C_s(a)$ in $\mathcal{A}_\mathcal{B}$. Note that we do not assert $\neg C_s(a)$ for $(a, s) \notin \mathcal{B}$, keeping the open world assumption for bindings.

For example, let A (resp. $C_{\mathcal{S}_\mathcal{T}}$) be the concept $Park$ (resp. $SpatialFeat$), cp be the spatial object of "City Park," and the polygon $poly\_cp$ representing cp's spatial extend. The KB has the assertions $Park \sqsubseteq (loc\ Park)$, $CityParkCafe \sqsubseteq (loc_{cp}\ Park)$, and $CityParkCafe(c)$. Then, the transformation produces $Park \sqsubseteq (SpatialFeat \sqcap Park)$, $CityParkCafe \sqsubseteq (C_{poly\_cp} \sqcap Park)$, $C_{poly\_cp} \sqsubseteq SpatialFeat$, and $C_{poly\_cp}(cp)$.

Note that $\mathcal{K}_\mathcal{S}$ is indeed a DL-Lite$_R$ ontology, by the syntactic restrictions on localization concepts. It is not hard to verify that the models of $\mathcal{L}_\mathcal{S}$ and $\mathcal{K}_\mathcal{S}$ with the same domain ($\Delta^\mathcal{I} = \Delta^{\mathcal{I}'}$) coincide on common concepts and roles as follows: (i) if $\mathcal{I} \models \mathcal{L}_\mathcal{S}$, then $\mathcal{I}' \models \mathcal{K}_\mathcal{S}$ where $C_s^{\mathcal{I}'} = \{e \in \Delta^\mathcal{I} \mid (e, s) \in b^\mathcal{I}\}$, $C_{\mathcal{S}_\mathcal{T}}^{\mathcal{I}'} = \bigcup_{s \in \Gamma_S} C^{\mathcal{I}'}$ $(= dom(b^\mathcal{I}))$; conversely, (ii) if $\mathcal{I}' \models \mathcal{K}_\mathcal{S}$, then $\mathcal{I} \models \mathcal{L}_\mathcal{S}$ where $b^\mathcal{I} = \{(e, s) \mid e \in C_s^{\mathcal{I}'}\}$ and $(loc\ A)^\mathcal{I} = C_{\mathcal{S}_\mathcal{T}}^{\mathcal{I}'} \cap A^{\mathcal{I}'}$. As an easy consequence of this correspondence, we obtain:

Proposition 1. *Satisfiability checking and CQ answering for ontologies in DL-Lite$_R(S)$ is FO-rewritable.*

As the models of $\mathcal{L}_\mathcal{S}$ and $\mathcal{K}_\mathcal{S}$ correspond, we can check satisfiability on $\mathcal{K}_\mathcal{S}$, i.e., a standard DL-Lite$_R$ KB. An ontology CQ q over $\mathcal{L}_\mathcal{S}$ is easily rewritten to a CQ over $\mathcal{K}_\mathcal{S}$.

4 Query Answering in DL-Lite$_R(S)$

We next define spatial conjunctive queries (SCQ) over $\mathcal{L}_\mathcal{S} = \langle \mathcal{T}, \mathcal{A}, \mathcal{S}, \mathcal{B} \rangle$. Such queries may contain ontology and spatial predicates. Formally, an SCQ $q(\mathbf{x})$ is a formula

$$Q_{O_1}(\mathbf{x}, \mathbf{y}) \wedge \cdots \wedge Q_{O_n}(\mathbf{x}, \mathbf{y}) \wedge Q_{S_1}(\mathbf{x}, \mathbf{y}) \wedge \cdots \wedge Q_{S_m}(\mathbf{x}, \mathbf{y}), \tag{1}$$

where $\mathbf{x}$ are the *distinguished* variables and $\mathbf{y}$ are either *non-distinguished* (*bound*) variables or individuals from Γ_I. Each $Q_{O_i}(\mathbf{x}, \mathbf{y})$ is an atom for $\mathcal{T}$ and of the form

$A(z)$ or $P(z, z')$, with z, z' from $\mathbf{x} \cup \mathbf{y}$; the atoms $Q_{S_j}(\mathbf{x}, \mathbf{y})$ are over the vocabulary for the spatial relations in Sec. 2 and of the form $S(z, z')$, with z, z' from $\mathbf{x} \cup \mathbf{y}$.

For example, $q(x) = Playground(x) \wedge Within(x, y) \wedge Park(y)$ is a SCQ which intuitively returns the playgrounds located in parks.

Semantics. Let $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}, b^{\mathcal{I}} \rangle$ be an interpretation of $\mathcal{L}_{\mathcal{S}}$. A *match* for $q(\mathbf{x})$ in $\mathcal{I}$ is a function $\pi : \mathbf{x} \cup \mathbf{y} \to \Delta^{\mathcal{I}}$ such that $\pi(c) = c^{\mathcal{I}}$, for each constant c in $\mathbf{x} \cup \mathbf{y}$, and for each $i = 1, \ldots n$ and $j = 1, \ldots, m$, (i) $\pi(z) \in A^{\mathcal{I}}$, if $Q_{O_i}(\mathbf{x}, \mathbf{y}) = A(z)$; (ii) $(\pi(z), \pi(z')) \in P^{\mathcal{I}}$, if $Q_{O_i}(\mathbf{x}, \mathbf{y}) = P(z, z')$; and (iii) $\exists s, s' \in \Gamma_S : (\pi(z), s) \in b^{\mathcal{I}} \wedge (\pi(z'), s') \in b^{\mathcal{I}} \wedge \mathcal{S} \models S(s, s')$, if $Q_{S_j}(\mathbf{x}, \mathbf{y}) = S(z, z')$. That is, for spatial predicates individuals must have (unique) spatial extensions and the relationship between them must hold.

Then, a tuple $\mathbf{c} = c_1, \ldots, c_k$ over Γ_I is an *answer* for $q(\mathbf{x})$ in $\mathcal{I}$, $\mathbf{x} = x_1, \ldots$, if $q(\mathbf{x})$ has some match π in $\mathcal{I}$ such that $\pi(x_i) = c_i$, $i = 1, \ldots, k$; furthermore, $\mathbf{c}$ is an answer for $q(\mathbf{x})$ over $\mathcal{L}_{\mathcal{S}}$, if it is an answer in every model $\mathcal{I}$ of $\mathcal{L}_{\mathcal{S}}$. The *result* of $q(\mathbf{x})$ over $\mathcal{L}_{\mathcal{S}}$, denoted $res(q(\mathbf{x}), \mathcal{L}_{\mathcal{S}})$, is the set of all its answers.

The semantic correspondence between $\mathcal{L}_{\mathcal{S}}$ and $\mathcal{K}_{\mathcal{S}} = \langle \mathcal{T}', \mathcal{A}' \rangle$ guarantees that we can transform $q(\mathbf{x})$ into an equivalent query over $\mathcal{L}_{\mathcal{S}}{}' = \langle \mathcal{T}', \mathcal{A}', \mathcal{S}, \mathcal{B} \rangle$ by replacing each spatial atom $S(z, z')$ in $q(\mathbf{x})$ with

$$\bigvee_{s,s' \in \Gamma_S} (C_s(z) \wedge C_s(z') \wedge S(s, s')). \tag{2}$$

The resulting formula is easily cast into form $uq(\mathbf{x}) = q_1(\mathbf{x}) \vee \cdots \vee q_l(\mathbf{x})$, i.e., a union of CQs $q_i(\mathbf{x})$. The answers of $uq(\mathbf{x})$ in an interpretation $\mathcal{I}'$ of $\mathcal{L}_{\mathcal{S}}{}'$ are the answers of all $q_i(\mathbf{x})$ in $\mathcal{I}'$, and $res(uq(\mathbf{x}), \mathcal{L}_{\mathcal{S}}{}')$ is defined in the obvious way. We then can show:

Proposition 2. *For every SQC $q(\mathbf{x})$ over $\mathcal{L}_{\mathcal{S}}$, $res(q(\mathbf{x}), \mathcal{L}_{\mathcal{S}}) = res(uq(\mathbf{x}), \mathcal{L}_{\mathcal{S}}{}')$.*

Hence, answering SCQs in DL-Lite$_R(S)$ ontologies is FOL-rewritable. In particular, for fixed $\mathcal{S}$, we can eliminate $S(s, s')$ from (2), which yields a pure ontology query. Alternatively, we can replace it with $S_{s,s'}(z)$, where $S_{s,s'}$ is a fresh concept, and add $C_s \sqsubseteq S_{s,s'}$ to the TBox $\mathcal{T}'$ iff $\mathcal{S} \models S(s, s')$, thus changing $\mathcal{S}$ more flexibly.

Spatial Conjunctive Query Evaluation. The above SCQ rewriting is exponential in the number of spatial atoms, which incurs limitations. However, *if no bounded variables occur in spatial atoms*, we can separate query answering into an ontology part and a spatial query part, which can be efficiently evaluated in two stages:

(1) evaluate the ontology part of the query $q(\mathbf{x})$ (i.e., drop all spatial atoms) over $\mathcal{L}_{\mathcal{S}}{}'$. For that we can apply the *standard* DL-Lite$_R$ query rewriting of $PerfectRef$ and evaluate the result over the ABox, stored in an RDBMS.
(2) filter the result of Step (1), by evaluating the formulas (2) on the bindings π for the distinguished variables $\mathbf{x}$ (which are mapped to individuals). For that, retrieve in Step (1) also all instances of C_s, for all $s \in \Gamma_S$.

Step (2) amounts to computing a *spatial join* $\bowtie_S$, for which (at least) different evaluation strategies exist. One strategy, denoted as O_D, relies entirely on the functions of a spatial-extended RDBMS. The other, denoted as O_I, relies on an internal evaluation of $\bowtie_S$, i.e., spatial relations, where the intermediate results are kept in-memory.

We have considered both strategies, restricting to *acyclic queries* (i.e., the query hypergraph is tree-shaped; see e.g. [12] for a definition). For such queries, join trees can be built, which can be processed in a bottom up manner. In doing so, we distinguish

between ontology and spatial nodes, and actually interleave the DL-Lite$_R$ query rewriting (Step (1)) and spatial join checking (Step (2)). For space reasons, we omit details.

Note that for strategy O_D, we rewrite the spatial atoms ($Contains$, $Within$, etc.) directly to corresponding functions (cf. [8] for details) of the spatial-extension of the RDBMS. The different strategies noticeably affect the performance (see Sec. 8).

5 From Keywords to Spatial Conjunctive Queries

In this section, we provide the details for the generation of SCQ from a *valid* sequence of keywords; We shall consider in the next section how such sequences are obtained in a controlled way, by *automatic completion* and checking *keyword combinations*.

We assume an ontology O_U, which has an associated *meta-model* for structuring the query generation (described below). The generation is realized in three steps. First, the keywords are mapped to concepts from O_U and to spatial predicates. Then, a set of completion rules (which regard the meta-model) is applied to the resulting sequence of atomic formulas. Finally the completed sequence is converted into a SCQ.

We assume that spatio-thematic KBs are labeled, i.e., they are of the form $\mathcal{L}_\mathcal{S} = \langle \mathcal{T}, \mathcal{A}, \mathcal{S}, \mathcal{B}, \mathcal{N} \rangle$, where $\mathcal{N}$ is a set of textual labels representing keywords. The labels of $\mathcal{N}$ are assigned by `rdfs:label` to the concepts of $\mathcal{T}$. Multiple labels can be assigned to a single element, which allows to have synonyms. Further, translations for keywords in different languages can be enabled by the assignments.

Meta-model for Structured Query Generation. We require on the top level of the ontology in use a strict separation of the concepts for spatial features $SpatialFeat$ (e.g., $Park$, $Restaurant$, etc.), qualitative values $QVal$ (e.g., operator Op, $Cuisine$, etc.), and $Geometry$ (e.g., $Point$, $Polygon$, etc.). Since our approach is designed to query spatial objects, every query has to be related to some $SpatialFeat$, which is extended by the subroles of $hasQVal$ with qualitative values (asserted to $QVal$) and is represented by the role $hasGeometry$ as a geometry (asserted to $Geometry$). By this separation on the top level (which also exists in GeoOWL `http://www.w3.org/2005/Incubator/geo/XGR-geo/`), we have a *meta-model*, which is then used for the generation of "meaningful" queries. Any ontology used with our approach has to be structured according to the meta-model. Fig. 1 shows some axioms of the meta-model for a specific ontology.

Generation of SCQs from Keywords. The automatic completion and combination step produces a set of *valid* keyword sequences, from which one sequence $K = (k_1, k_2, \ldots, k_n)$ is chosen (unless the user determines one). Each keyword k_i represents either a concept or a spatial predicate. We must connect all k_i according to the meta-model to obtain SCQs, which then evaluate to spatial objects.

The rewriting of K to a SCQ Q is based on three steps that resemble a transducer with a context-free (left-recursive) grammar and a set of completion rules. The latter are important, because even if the transducer generates syntactically correct queries, their results might not consist of spatial objects. E.g., we could have a query $ItalianCuisine(x)$, but the results R = (*pizza, pasta, ...*) could not be located on a map. Therefore, we have to extend the query as follows: $Restaurant(x) \wedge hasCuisine(x, y) \wedge ItalianCuisine(y)$.

Table 1. Completion rules; the result of rules (R1)–(R4) is denoted as subquery (SQ)

(R1) If $C_1 \sqsubseteq \mathit{SpatialFeat}$ and $C_2 \sqsubseteq \mathit{QualAttribute}$ rewrite to $(C_1 \ \ \mathit{hasQVal} \ \ C_2)$;
(R2) If $C_1 \sqsubseteq \mathit{SpatialFeat}$, $C_2 \sqsubseteq \mathit{QualAttribute}$, $C_3 \sqsubseteq \mathit{QualAttribute}$ rewrite to $((C_1 \ \ \mathit{hasQVal} \ \ C_2) \ \ \mathit{hasQVal} \ \ C_3)$;
(R3) If $C_1 \sqsubseteq \mathit{QualAttribute}$ rewrite to $(\mathit{SpatialFeat} \ \ \mathit{hasQVal} \ \ C_1)$;
(R4) If $C_1 \sqsubseteq \mathit{QualAttribute}$ and $C_2 \sqsubseteq \mathit{QualAttribute}$ rewrite to $((\mathit{SpatialFeat} \ \ \mathit{hasQVal} \ \ C_1) \ \ \mathit{hasQVal} \ \ C_2)$;
(R5) If $E_1 \sqsubseteq \mathit{SpatialFeat}$ or E_1 is a SQ, $E_2 \sqsubseteq \mathit{SpatialFeat}$ or E_2 is SQ, and S is a spatial predicate rewrite to $((E_1) \ \ S \ \ E_2)$;
(R6) If $E_1 \sqsubseteq \mathit{SpatialFeat}$ or E_1 is a SQ, and $E_2 \sqsubseteq \mathit{SpatialFeat}$ or E_2 is SQ rewrite to $((E_1) \ \ \mathit{NextTo} \ \ E_2)$;

In the following, we describe the three steps in the rewriting of K in detail:

(1) We obtain a new sequence K' from the sequence K by replacing every keyword with either a concept from $\mathcal{T}$ or a predefined spatial predicate. We check whether the keywords are associated to a concept in $\mathcal{N}$, otherwise we ignore it.
(2) We apply the completion rules in Table 1 on K' in a left-to-right order until no rules are applicable, resulting in a sequence K''.
(3) We generate the query $q(\mathbf{x})$ from K'' according to the function

$$f(K'') = (\cdots((C_1(x_1) \wedge E_{1,1}(x_1, y_1) \wedge E_{1,2}(y_1)) \wedge \chi_2) \wedge \cdots) \wedge \chi_n$$

where $\chi_i = E_{i,1}(\vartheta(E_{i-1,1}), y_i) \wedge E_{i,2}(y_i)$ and C_1 is a concept atom; each $E_{i,1}$ is either empty, a role atom, or a spatial atom, and each $E_{i,2}$ is either empty or a concept atom; $\vartheta(E_{i,1})$ is x_i if $E_{i,1}$ is a spatial atom, and x_{i-1} if $E_{i,1}$ is a role atom. These assignments ensure that the spatial atoms are always related to the top concept, while role atoms are related to the next level in the query tree.

After these steps, we obtain a valid SPQ $q(\mathbf{x})$ for query evaluation (Sec. 4). For rules (R2)–(R4), Table 1 shows in fact a simplified version, as they could be extended to arbitrary sequences of $\mathit{QualAttributes}$. Furthermore, rule (R6) defines a *default* relationship, if two spatial features are queried. Rewriting them to a simple conjunction between $C_1(x)$ and $C_2(x)$ would often lead to empty results, as two identical objects assigned to different concepts do not often exist within geospatial data sources.

Example 1. Given the keywords $K = (\textit{italian cuisine, non-smoking, in, park})$, we apply the first step, where we replace every k_i with an associated concept C_i from $\mathcal{N}$: $K' = (\mathit{ItalianCuisine}, \mathit{NonSmoking}, \mathit{Within}, \mathit{Park})$. In the second step we apply the completion rules to obtain $K'' = (((\mathit{SpatialFeat} \ \ \mathit{hasQVal} \ \ \mathit{ItalianCuisine}) \ \ \mathit{hasQVal} \ \ \mathit{NonSmoking}) \ \ \mathit{Within} \ \ \mathit{Park})$. Finally we get a SCQ $q(x_1, x_2) = f(K'')$ with

$$\mathit{SpatialFeat}(x_1) \wedge \mathit{hasQVal}(x_1, y_1) \wedge \mathit{ItalianCuisine}(y_1) \wedge \mathit{hasQVal}(x_1, y_2) \wedge \mathit{NonSmoking}(y_2) \wedge \mathit{Within}(x_1, x_2) \wedge \mathit{Park}(x_2) \ .$$

6 Generating Keyword Sequences

Since our approach is designed to have a single text-field for the keyword entries, we aim to provide fast automatic completion, keywords detection, and keyword combination functions. If a user enters keywords on a user interface (UI), we guide her by

automatic completion and by showing possible combinations compliant with the ontology. For that, we must take the structure of the KB into account. Furthermore, as many combinations may be compliant, a selection of "relevant" combinations must be provided.

As the need for very low response time (e.g., below 100ms) makes on-demand calculation from the KB infeasible, a *prefix index* is created offline that stores all possible prefixes for a label of $\mathcal{N}$. It amounts to a function $f_P(e)$ which maps a string e to all possible labels of $\mathcal{N}$, such that $\bigcup_{n\in\mathcal{N}}(Pref(e) \subseteq Pref(n))$.

For example, the labels $\mathcal{N} = \{pub, public, park\}$, f_P map *p*, *pub*, and *park* as follows: $\{p\} \to \{park, pub, public\}$, $\{pu\} \to \{pub, public\}$, $\{park\} \to \{park\}$.

Syntactic Connectivity of Concepts. As multiple keywords are entered, we need to determine which concepts are connectable. We use a notion of syntactic connectivity **C** based on the syntactic structure of the KB, which captures the connection between two concepts through subconcepts and roles, but also through a common subsumer. For two concepts, we follow the inclusion assertion and check whether they share a common subsumer denoted as C_S, excluding the top concept. As the KB is based on DL-Lite$_R$, we can capture the following inclusion assertions: (i) concept inclusion $M_C : C_1 \sqsubseteq C_2$, role hierarchies $M_H : R_1 \sqsubseteq R_2$; (ii) role membership which covers the range (resp. domain) of a role as $M_R : \exists R^- \sqsubseteq C$ (resp. $M_D : \exists R \sqsubseteq C$); and (iii) mandatory participation $M_P : C \sqsubseteq \exists R$. We deliberately do not consider disjoint concepts as $C_1 \sqsubseteq \neg C_2$ in the inclusions, and distinguish *direct* and *indirect* connections between two concepts.

A *direct connection* between concepts C_A and C_B exists, denoted $\phi_D(C_A, C_B)$, if a sequence $C_A \to_M \exists R_1 \to_M C_1 \to_M \exists R_1 \dots C_n \to_M \exists R_n \to_M C_B$ exists, where $M = M_D \cup M_H \cup M_C \cup M_R \cup M_P$. Furthermore, an *indirect connection* between C_A and C_B exists, denoted $\phi_I(C_A, C_B)$, if $\phi_D(C_A, C_S) \wedge \phi_D(C_B, C_S)$ holds for some C_S.

Example 2. In the example Fig. 1, the concepts $Supermarket$ and Op are directly connected: $Supermarket \to_{M_C} Shop \to_{M_P} \exists hasOp \to_{M_R} Op$. On the other hand, *GuestGarden* and $Wlan$ are indirectly connected: $GuestGarden \to_{M_C} QVal \to_{M_P} \exists hasQVal \to_{M_R} SpatialFeat \leftarrow_{M_R} \exists hasQVal \leftarrow_{M_P} QVal \leftarrow_{M_C} Wlan$.

In general, several sequences that witness $\phi_D(C_A, C_B)$ resp. $\phi_I(C_A, C_B)$ exist.

Automatic Completion, Detection, and Combination of Keywords. As we get a sequence of entered strings $E = (e_1, e_2, \dots, e_n)$, we need several steps to create the completed keywords, as the strings could be prefixes or misspelled.

First, we obtain the set of labels $L \subseteq \mathcal{N}$ by applying the prefix function $f_P(e_i)$ for every $e_i \in E$. Second, we build several levels of labels $L_1, \cdots, L_m$ based on the size of the subsets of L. As every L_i has the subsets $L_{i,1}, \cdots, L_{i,o}$ of the same size, we check for every $L_{i,j}$, if every pair of concepts (assigned to the labels of $L_{i,j}$) is syntactically connected at least in one direction. If we have found a $L_{i,j}$ with connected concepts, we add all sets of L_i (which are connectable) to the results. This is done by concatenating the labels of every set of L_i and add them to the result strings.

By introducing an iterative algorithm, we return the largest possible combinations of keywords, thus excluding misspelled strings. However, we have in the worst-case exponentially many connectivity checks in the lengths of E.

Example 3. Given E = (*rest, in, non-smok*), we obtain the labels L = {*restaurant, indian food, intl food, non-smoking*}. The first level of L contains the sets $L_{1,1}$ = {*restaurant, indian food, non-smoking*} and $L_{1,2}$ = {*restaurant, intl food, non-smoking*}. The concepts assigned to them are $C_{1,1}$ = $\{Restaurant, IndianCuisine, NonSmoking\}$ and $C_{1,2} = \{Restaurant, IntlCuisine, NonSmoking\}$. Then, we check for $C_{1,1}$, if every pair $(C, C'), C \neq C' \in C_{1,1}$, is syntactically connected, and likewise for $C_{1,2}$. The first two pairs are directly connected and the last pair is indirectly connected by the common subsumer $SpatialFeat$. Hence, the concepts in $C_{1,1}$ (and in $C_{1,2}$) are connectable. Then, we concatenate $L_{1,1}$ (resp. $L_{1,2}$) and add the strings to the results.

7 Refinement of Conjunctive Query Generation

While FO-rewritability of CQ answering over DL-Lite$_R$ KBs implies tractable data complexity, the size of the rewriting can increase exponentially with the number of atoms in the input CQ. Empirical findings [20] are that queries with more than 5-7 atoms can lead to large UCQs (e.g., unions of thousands of CQs) which cannot be handled by current RDBMS. Similar problems emerge with our generated SCQ (Sec. 8). One reason is the completion step in the SCQ generation. The generated SCQ can be too *general*, as we complete the intermediate sequence K' (Sec. 5) with the concept $SpatialFeat$ and role $hasQVal$, which are at the top-level (by our meta-model) of an ontology.

The *refinement* O_Q of the completion step is applied on every ontological subquery of a SCQ of the form $S(x_1) \wedge R_1(x_1, y_1) \wedge C_1(y_1) \wedge \ldots \wedge R_n(x_{n-1}, y_n) \wedge C_n(y_n)$, where $S \sqsubseteq SpatialFeat$, $\{R_1 \ldots, R_n\} \sqsubseteq hasQVal$, and $\{C_1, \ldots, C_n\} \sqsubseteq QualAttribute$ holds. It is based on the following ideas:

- Reduce the concept and role hierarchies: every edge in a path of ϕ_D or ϕ_I is an inclusion assertion, which increases the size of the rewritten UCQ; in particular, role inclusions can cause an exponential blow up [7];
- keep connectivity: by choosing paths according to ϕ_I, we keep the domain, range, mandatory participation, regarding the roles connecting S to $\{C_1, \ldots, C_n\}$.

Before applying O_Q, note that so far, S is a *most common subsumer* different from the top concept with respect to ϕ_I; i.e., for every pairs $(S, C_1), \ldots, (S, C_i)$, $\phi_I(S, C_j)$ holds for all j and the sum of path lengths for $\phi_I(S, C_j)$ is maximal. Thus, we try to minimize the path lengths under the constraint that ϕ_I is fulfilled for all pairs $\phi_I(S, C_j)$.

Briefly, it works as follows. We start the refinement O_Q by taking every subconcept S_i of S. We choose a shortest path, say p_j, according to ϕ_I for every pair (S_i, C_j), $1 \leq j \leq n$, and we add up all path lengths $|p_j|$ to len_{S_i}. Finally, we choose the S_i with the lowest len_{S_i} as a replacement of S and $R_1 \ldots, R_n$, where the latter are replaced with the roles appearing on the shortest paths p_j for S_i.

Example 4. Let $q(x_1)$ be $SpatialFeat(x_1) \wedge\ hasQVal(x_1, y_1) \wedge\ ItalianCuisine(y_1) \wedge\ hasQVal(x_1, y_2) \wedge NonSmoking(y_2)$. For the pairs $(Rest, ItalianCuisine)$ and $(Rest,$

$NonSmoking$), we have a path p_1 of length 2 ($Rest \rightarrow \exists hasCuisine \rightarrow ItalianCuisine$) and another path p_2 of length 2 ($Rest \rightarrow \exists provides \rightarrow NonSmoking$). Hence, the refinement O_Q produces the optimized query $q'(x_1)$, as the original paths are both of length 3 and $Rest$ is a subconcept of $SpatialFeat$: $Rest(x_1) \wedge hasCusine(x_1, y_1) \wedge ItalianCuisine(y_1) \wedge provides(x_1, y_2) \wedge NonSmoking(y_2)$.

We point out that after applying O_Q, we may lose *completeness* with respect to the original SCQ, as shown by the following example. Given a spatio-thematic KB containing ABox assertions $Rest(i_1)$, $hasCuisine(i_1, i_2)$, $ItalianCuisine(i_2)$, $SpatialFeat(i_3)$, $hasQVal(i_3, i_2)$, and $ItalianCuisine(i_2)$, such that $hasCuisine$ has defined domain $Rest$ and range $Cuisine$. The query $q(x_1) = SpatialFeat(x_1) \wedge hasQVal(x_1, y_1) \wedge ItalianCuisine(y_1)$ evaluates to $\{i_1, i_3\}$. If we refine $q(x_1)$ to the SCQ $q'(x_1) = Rest(x_1) \wedge hasCuisine(x_1, y_1) \wedge ItalianCuisine(y_1)$, we just get $\{i_1\}$ as a result. Informally, completeness can be lost if the ABox assertions are very general. One way to keep completeness is thus to impose conditions on the ABox, which ensure that ABox assertions have to fulfill certain conditions.

8 Implementation and Experimental Results

We have implemented a prototype of our keyword-based query answering approach. It is developed in Java 1.6 and uses PostGIS 1.5.1 (for PostgreSQL 9.0) as spatial-extended RDBMS. For the FO-rewriting of DL-Lite$_R$, we adapted OWLGRES 0.1 [22] to obtain the *perfect rewriting* (with *PerfectRef*) of a CQ and the TBox. We evaluate spatial atoms in two different ways (Sec. 4), namely as O_D by using the query evaluation of PostGIS or as O_I as a built-in component of our query evaluation algorithm. For O_D, we use the PostGIS functions for evaluation, e.g., `ST_Contains(x, y)`, and for O_I, we apply the functions of the JTS Topology Suite (`http://tsusiatsoftware.net/jts`).

As part of a consortium with AIT Mobility Department (routing), Fluidtime (UI), ITS Vienna Region (data and routing), we have integrated our prototype for the keyword-based query answering in the MyITS system for intention-oriented route planning (`http://myits.at/`). Currently, the following services are available:

1. *Neighborhood routing*, where a user desires to explore the neighborhood for a keyword-based query; and
2. *Via routing*, where a route is calculated between a given origin-destination pair via some POI, which is dynamically determined by a keyword-based query.

Scenario. Our benchmarks are based on the usage scenarios of MyITS, which has a DL-Lite$_R$ geo-ontology with the following metrics: 324 concepts (with 327 inclusion assertions); 30 roles (with 19 inclusion assertions); 2 inverse roles; 23 (resp. 25) domains (resp. ranges) of roles; 124 *normal* individuals; a maximal depth of 7 (4) in the concept (role) hierarchy (`http://www.kr.tuwien.ac.at/staff/patrik/GeoConceptsMyITS-v0.9-Lite.owl`). For the *spatial* objects, we added and mapped the POIs of greater Vienna contained in OSM ($\approx$ 70k instances), in the Falter database ($\approx$ 3700 instances), and parts of the OGD Vienna data ($\approx$ 7200 instances). The annotation step created $\approx$18700 individuals, which lead to $\approx$18700 concepts and $\approx$26000 role assertions. The low annotation rate of 23% is related to the exclusion of some OSM POIs (e.g., benches, etc.) and the ongoing extensions of the mapping framework.

Table 2. Benchmark Results (Evaluation time in secs), unrefined results in parentheses

(a) Benchmark B_1

	Instances	Query Size	Time
Q_1	106 (109)	438 (2256)	1.66 (4.96)
Q_2	1623 (1623)	51 (2256)	1.23 (5.59)
Q_3	204 ($—^s$)	28 (71712)	1.14 ($—^s$)
Q_4	32 ($—^m$)	56 ($—^m$)	1.48 ($—^m$)
Q_5	3 ($—^m$)	112 ($—^m$)	4.11 ($—^m$)

(b) Benchmark B_2, time only with O_Q

	Instances	Query Size	Time	
			O_I	O_D
Q_6	93 (93)	2 (2)	1.54	19.3
Q_7	378 (378)	4 (4)	2.22	$—^t$
Q_8	26 ($—^s$)	30 (71714)	3.37	$—^t$
Q_9	151 (151)	2 (2)	2.02	$—^t$

Experiments. We conducted our experiments on a Mac OS X 10.6.8 system with an Intel Core i7 2.66GHz and 4 GB of RAM. We increased `shared_buffers` and `work_mem` of PostgreSQL 9.0 to utilize available RAM. For each benchmark, the average of five runs for the query rewriting and evaluation time was calculated, having a timeout of 10 minutes, and a memout of 750 MB for each run. The results shown in Table 2 present runtime in seconds and query size (number of atoms in the CQ), and use $—^s$ to denote DB errors (e.g., the stack depth limit of Postgres 9.0 is reached), $—^m$ for Java heap space limit has been reached (750 MB), and $—^t$ for timeout.

Benchmarks. We designed the first benchmark B_1 based on keywords to measure the refinement O_Q on CQ without spatial predicates. The queries used in B_1 are

Q_1: $(spar)$ matches individuals run by "Spar";
Q_2: $(guest\ garden)$ returns the individuals with a guest garden;
Q_3: $(italian\ cuisine, guest\ garden)$ retrieves individuals that serve italian cuisine (including Pizzerias, etc.) and have a guest garden;
Q_4: $(italian\ cuisine, guest\ garden, wlan)$ gives individuals of Q_3 that in addition provide WLAN; and
Q_5: $(italian\ cuisine, guest\ garden, wlan, child\ friendly)$ returns individuals of Q_4 that in addition are child-friendly.

As described above, the keywords are completed to SCQ prior to evaluation as described.

The benchmark B_2 aims at comparing the database (denoted O_D) and internal evaluation of spatial predicates (denoted O_I) under the refinement O_Q. Its queries are

Q_6: $(playground, within, park)$ returns playgrounds in a park;
Q_7: $(supermarket, next\ to, pharmacy)$ matches supermarkets next to a pharmacy;
Q_8: $(italian\ cuisine, guest\ garden, next\ to, atm, next\ to, metro\ station)$ gives individuals with Italian food and a guest garden, whereby these individuals are next to an ATM and a metro station. The nesting of the query is as previously defined $(((italian\ cuisine, guest\ garden), next\ to), \ldots, metro\ station)$; and
Q_9: $(playground, disjoint, park)$ retrieves playgrounds outside a park.

As the results in Table 2 show, the refinement O_Q is essential for feasibility. Without it, Java exceeds heap space limitation during *perfect rewriting* in most cases, and SQL queries become too large for the RDBMS. The ontology of our scenario is big, yet captures only a domain for cities using OSM, OGD Vienna, and Falter.

As ground truth we assume the unrefined query. We lost completeness only in Q_1; this is due to three objects, which were tagged in OSM as shops but not supermarkets. With respect to the benchmark queries, the OSM tagging and our (heuristic) mapping has a minor effect on the completeness. Further, the results for Q_2 to Q_5 reflect the fact that adding keywords extends the selectivity of the query (smaller results), but enlarges the UCQ considerably.

We were surprised by the large difference between internal and external evaluation of the spatial relations. We would have expected the external evaluation by the RDBMS is more efficient. Rewritten SQL queries have a three-leveled nesting, which consists of spatial joins ($\bowtie_S$) on the first, unions ($\cup$) on the second, and normal joins ($\bowtie$) on third level. It seems that standard query evaluation and optimization (in Postgres 9.0) are overwhelmed by such complex structures.

9 Related Work and Conclusion

Regarding SCQ, the closest to our work is [18], where crisp results for the combination of FO-rewritability of DL-Lite combined with the RCC-family (which offers qualitative models of abstract regions in topological space) are provided. For more expressive DLs, Lutz et al. [17] introduced the notion of ω-admissibility, which allows the combination of $\mathcal{ALC}$ and $RCC8$ [19], for subsumption testing. In PelletSpatial [21], the authors implemented a hybrid combination of $\mathcal{SHOIN}$ and $RCC8$. We follow a different approach in which the spatial regions are considered as point sets as in [14,15]. However, we focus on scalable query answering (without distance primitives) and the related implementation issues. In this way, we face similar challenges as recent Geo-SPARQL engines did (e.g., Strabon [16] and Parliament [3]). However, we have a stronger focus on *ontology-based data access* than on linked open-data (with an RDF data model).

Keyword-based search on the Semantic Web is a well-covered field of research. A necessarily incomplete list of relevant approaches is SemSearch [24], XXploreKnow [23], and QUICK [25] which are general purpose search engines. The KOIOS [4], DOROAM [9], and the system of [2] support (text-based) spatial queries using ontologies. Our approach differs from these systems regarding the expressivity of DL-Lite, with the addition of spatial querying; the use of a meta-model for suitable query generation; and a focus on gradual extendibility with new data sources.

In this paper, we presented an extension of DL-Lite$_R$ with spatial objects using point-set topological relations for query answering. The extension preserves FO-rewritability, which allows us to evaluate a restricted class of conjunctive queries with spatial atoms over existing spatio-relational RDBMS. Second, we provided a technique for the generation of spatial conjunctive queries from a set of keywords. For this, we introduced a combination of a meta-model and completion rules to generate "meaningful" queries. Third, we implemented a prototype and performed experiments to evaluate the applicability in a real-world scenario. From our point of view, the first results are encouraging, as the evaluation time appeared to be moderate (always below 5 secs). Furthermore, our keyword-based approach is easy to extend, the text-based input is lightweight, and it has a reasonable precision through auto-completion and keyword combinations. However, precision could be improved by more advanced query expansion techniques (cf. [11]).

Future research is naturally directed to variants and extensions of the presented ontology and query language. E.g., one could investigate how spatial conjunctive queries work over $\mathcal{EL}$ [1] or Datalog$^{\pm}$ [5]. For our motivating application, the point set model was sufficient, but extending our approach with the DE-9IM model [10] would be appealing and introduce further spatial relations. Then, one could work on query expansion techniques and on refinement of query generation, in a way such that completeness is ensured. Finally, regarding the implementation, one could investigate the reason for the unexpected performance on very large queries with spatial functions and conduct further experiments on larger geospatial DBs, possibly comparing our approach to the mentioned Geo-SPARQL engines.

References

1. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In: IJCAI 2005, pp. 364–369. Morgan-Kaufmann Publishers (2005)
2. Baglioni, M., Masserotti, M.V., Renso, C., Spinsanti, L.: Improving geodatabase semantic querying exploiting ontologies. In: Claramunt, C., Levashkin, S., Bertolotto, M. (eds.) GeoS 2011. LNCS, vol. 6631, pp. 16–33. Springer, Heidelberg (2011)
3. Battle, R., Kolas, D.: Enabling the geospatial Semantic Web with Parliament and GeoSPARQL. Semantic Web Journal 3(4), 355–370 (2012)
4. Bicer, V., Tran, T., Abecker, A., Nedkov, R.: KOIOS: Utilizing semantic search for easy-access and visualization of structured environmental data. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part II. LNCS, vol. 7032, pp. 1–16. Springer, Heidelberg (2011)
5. Calì, A., Gottlob, G., Pieris, A.: Towards more expressive ontology languages: The query answering problem. Artificial Intelligence 193, 87–128 (2012)
6. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., Rodriguez-Muro, M., Rosati, R.: Ontologies and databases: The *DL-Lite* approach. In: Tessaris, S., Franconi, E., Eiter, T., Gutierrez, C., Handschuh, S., Rousset, M.-C., Schmidt, R.A. (eds.) Reasoning Web 2009. LNCS, vol. 5689, pp. 255–356. Springer, Heidelberg (2009)
7. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The DL-Lite family. Journal of Automated Reasoning 39(3), 385–429 (2007)
8. Clementini, E., Sharma, J., Egenhofer, M.J.: Modelling topological spatial relations: Strategies for query processing. Computers & Graphics 18(6), 815–822 (1994)
9. Codescu, M., Horsinka, G., Kutz, O., Mossakowski, T., Rau, R.: DO-ROAM: Activity-oriented search and navigation with OpenStreetMaps. In: Claramunt, C., Levashkin, S., Bertolotto, M. (eds.) GeoS 2011. LNCS, vol. 6631, pp. 88–107. Springer, Heidelberg (2011)
10. Egenhofer, M.J., Franzosa, R.D.: Point set topological relations. International Journal of Geographical Information Systems 5(2), 161–174 (1991)
11. Fu, G., Jones, C.B., Abdelmoty, A.I.: Ontology-based spatial query expansion in information retrieval. In: Meersman, R., Tari, Z. (eds.) CoopIS/DOA/ODBASE 2005, Part II. LNCS, vol. 3761, pp. 1466–1482. Springer, Heidelberg (2005)
12. Gottlob, G., Leone, N., Scarcello, F.: The complexity of acyclic conjunctive queries. Journal of the ACM 48(3), 431–498 (2001)
13. Güting, R.H.: Geo-relational algebra: A model and query language for geometric database systems. In: Schmidt, J.W., Missikoff, M., Ceri, S. (eds.) EDBT 1988. LNCS, vol. 303, pp. 506–527. Springer, Heidelberg (1988)

14. Haarslev, V., Lutz, C., Möller, R.: A description logic with concrete domains and a role-forming predicate operator. Journal of Logic and Computation 9(3), 351–384 (1999)
15. Kutz, O., Wolter, F., Zakharyaschev, M.: A note on concepts and distances. In: DL 2001. CEUR-WS, vol. 49 (2001)
16. Kyzirakos, K., Karpathiotakis, M., Koubarakis, M.: Strabon: A semantic geospatial DBMS. In: Cudré-Mauroux, P., et al. (eds.) ISWC 2012, Part I. LNCS, vol. 7649, pp. 295–311. Springer, Heidelberg (2012)
17. Lutz, C., Milicic, M.: A tableau algorithm for description logics with concrete domains and general TBoxes. Journal of Automated Reasoning 38(1-3), 227–259 (2007)
18. Özçep, Ö.L., Möller, R.: Scalable geo-thematic query answering. In: Cudré-Mauroux, P., et al. (eds.) ISWC 2012, Part I. LNCS, vol. 7649, pp. 658–673. Springer, Heidelberg (2012)
19. Randell, D.A., Cui, Z., Cohn, A.G.: A spatial logic based on regions and connection. In: KR 1992, pp. 165–176. Morgan Kaufmann (1992)
20. Rosati, R., Almatelli, A.: Improving query answering over DL-Lite ontologies. In: KR 2010, pp. 290–300. AAAI Press (2010)
21. Stocker, M., Sirin, E.: Pelletspatial: A hybrid RCC-8 and RDF/OWL reasoning and query engine. In: OWLED 2009. Springer, Heidelberg (2009)
22. Stocker, M., Smith, M.: Owlgres: A scalable OWL reasoner. In: OWLED 2008. Springer, Heidelberg (2008)
23. Tran, T., Cimiano, P., Rudolph, S., Studer, R.: Ontology-based interpretation of keywords for semantic search. In: Aberer, K., et al. (eds.) ISWC/ASWC 2007. LNCS, vol. 4825, pp. 523–536. Springer, Heidelberg (2007)
24. Uren, V.S., Lei, Y., Motta, E.: SemSearch: Refining semantic search. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 874–878. Springer, Heidelberg (2008)
25. Zenz, G., Zhou, X., Minack, E., Siberski, W., Nejdl, W.: From keywords to semantic queries - incremental query construction on the semantic web. J. Web Semant. 7(3), 166–176 (2009)

Representation and Querying of Valid Time of Triples in Linked Geospatial Data*

Konstantina Bereta, Panayiotis Smeros, and Manolis Koubarakis

National and Kapodistrian University of Athens, Greece
{Konstantina.Bereta,psmeros,koubarak}@di.uoa.gr

Abstract. We introduce the temporal component of the stRDF data model and the stSPARQL query language, which have been recently proposed for the representation and querying of linked geospatial data that changes over time. With this temporal component in place, stSPARQL becomes a very expressive query language for linked geospatial data, going beyond the recent OGC standard GeoSPARQL, which has no support for valid time of triples. We present the implementation of the stSPARQL temporal component in the system Strabon, and study its performance experimentally. Strabon is shown to outperform all the systems it has been compared with.

1 Introduction

The introduction of time in data models and query languages has been the subject of extensive research in the field of relational databases [6,20]. Three distinct kinds of time were introduced and studied: *user-defined* time which has no special semantics (e.g., January 1st, 1963 when John has his birthday), *valid* time which is the time an event takes place or a fact is true in the application domain (e.g., the time 2000-2012 when John is a professor) and *transaction* time which is the time when a fact is current in the database (e.g., the system time that gives the exact period when the tuple representing that John is a professor from 2000 to 2012 is current in the database). In these research efforts, many temporal extensions to SQL92 were proposed, leading to the query language TSQL2, the most influential query language for temporal relational databases proposed at that time [20].

However, although the research output of the area of temporal relational databases has been impressive, TSQL2 did not make it into the SQL standard and the commercial adoption of temporal database research was very slow. It is only recently that commercial relational database systems started offering SQL extensions for temporal data, such as IBM DB2, Oracle Workspace manager, and Teradata [2]. Also, in the latest standard of SQL (SQL:2011), an important new feature is the support for valid time (called *application time*) and transaction time. Each SQL:2011 table is allowed to have at most two *periods* (one

* This work was supported in part by the European Commission project TELEIOS (http://www.earthobservatory.eu/)

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 259–274, 2013.

for application time and one for transaction time). A *period* for a table T is defined by associating a user-defined name e.g., EMPLOYMENT_TIME (in the case of application time) or the built-in name SYSTEM_TIME (in the case of transaction time) with two columns of T that are the start and end times of the period (a closed-open convention for periods is followed). These columns must have the same datatype, which must be either DATE or a timestamp type (i.e., no new period datatype is introduced by the standard). Finally, the various SQL statements are enhanced in minimal ways to capture the new temporal features.

Compared to the relational database case, little research has been done to extend the RDF data model and the query language SPARQL with temporal features. Gutierrez et al. [8,9] were the first to propose a formal extension of the RDF data model with valid time support. They also introduce the concept of *anonymous timestamps* in general temporal RDF graphs, i.e., graphs containing quads of the form $(s, p, o)[t]$ where t is a timestamp or an anonymous timestamp x stating that the triple (s, p, o) is valid in some unknown time point x. The work described in [11] subsequently extends the concept of general temporal RDF graphs of [9] to express temporal constraints involving anonymous timestamps. In the same direction, Lopes et al. integrated valid time support in the general framework that they have proposed in [15] for annotating RDF triples. Similarly, Tappolet and Bernstein [22] have proposed the language τ-SPARQL for querying the valid time of triples, showed how to transform τ-SPARQL into standard SPARQL (using named graphs), and briefly discussed an index that can be used for query evaluation. Finally, Perry [19] proposed an extension of SPARQL, called SPARQL-ST, for representing and querying spatiotemporal data. The main idea of [19] is to incorporate geospatial information to the temporal RDF graph model of [9]. The query language SPARQL-ST adds two new types of variables, namely spatial and temporal ones, to the standard SPARQL variables. Temporal variables (denoted by a # prefix) are mapped to time intervals and can appear in the fourth position of a quad as described in [9]. In SPARQL-ST two special filters are introduced: `SPATIAL FILTER` and `TEMPORAL FILTER`. They are used to filter the query results with spatial and temporal constraints (OGC Simple Feature Access topological relations and distance for the spatial part, and Allen's interval relations [3] for the temporal part).

Following the ideas of Perry [19], our group proposed a formal extension of RDF, called stRDF, and the corresponding query language stSPARQL for the representation and querying of temporal and spatial data using linear constraints [13]. stRDF and stSPARQL were later redefined in [14] so that geometries are represented using the Open Geospatial Consortium standards Well-Known-Text (WKT) and Geography Markup Language (GML). Both papers [13] and [14] mention very briefly the temporal dimension of stRDF and do not go into details. Similarly, the version of the system Strabon presented in [14], which implements stRDF and stSPARQL, does not implement the temporal dimension of this data model and query language. In this paper we remedy this situation by introducing all the details of the temporal dimension of stRDF and stSPARQL and implementing it in Strabon.

The original contributions of this paper are the following. We present in detail, for the first time, the valid time dimension of the data model stRDF and the query language stSPARQL. Although the valid time dimension of stRDF and stSPARQL is in the spirit of [19], it is introduced in a language with a much more mature geospatial component based on OGC standards [14]. In addition, the valid time component of stSPARQL offers a richer set of functions for querying valid times than the ones in [19]. With the temporal dimension presented in this paper, stSPARQL also becomes more expressive than the recent OGC standard GeoSPARQL [1]. While stSPARQL can represent and query geospatial data that changes over time, GeoSPARQL only supports static geospatial data.

We discuss our implementation of the valid time component of stRDF and stSPARQL in Strabon. We evaluate the performance of our implementation on two large real-world datasets and compare it to three other implementations: (i) a naive implementation based on the native store of Sesame which we extended with valid time support, (ii) AllegroGraph, which, although it does not offer support for valid time of triples explicitly, it allows the definition of time instants and intervals and their location on a time line together with a rich set of functions for writing user queries, and (iii) the Prolog-based implementation of the query language AnQL[1], which is the only available implementation with explicit support for valid time of triples. Our results show that Strabon outperforms all other implementations.

This paper is structured as follows. In Section 2 we introduce the temporal dimension of the data model stRDF and in Section 3 we present the temporal features of the query language stSPARQL. In Section 4 we describe how we extended the system Strabon with valid time support. In Section 5 we evaluate our implementation experimentally and compare it with other related implementations. In Section 6 we present related work in this field. Section 7 concludes this paper.

2 Valid Time Representation in the Data Model stRDF

In this section we describe the valid time dimension of the data model stRDF presented in [14]. The *time line* assumed is the (discrete) value space of the datatype `xsd:dateTime` of XML-Schema. Two kinds of time primitives are supported: time instants and time periods. A *time instant* is an element of the time line. A *time period* (or simply period) is an expression of the form $[B,E)$, $(B,E]$, (B,E), or $[B,E]$ where B and E are time instants called the *beginning* and the *ending* of the period respectively. Since the time line is discrete, we often assume only periods of the form $[B,E)$ with no loss of generality. Syntactically, time periods are represented by literals of the new datatype `strdf:period` that we introduce in stRDF. The value space of `strdf:period` is the set of all time periods covered by the above definition. The lexical space of `strdf:period` is trivially defined from the lexical space of `xsd:dateTime` and the closed/open

[1] `http://anql.deri.org/`

period notation introduced above. Time instants can also be represented as closed periods with the same beginning and ending time.

Values of the datatype `strdf:period` can be used as objects of a triple to represent *user-defined time*. In addition, they can be used to represent *valid times* of temporal triples which are defined as follows. A *temporal triple (quad)* is an expression of the form `s p o t.` where `s p o.` is an RDF triple and `t` is a time instant or a time period called the *valid time* of a triple. An *stRDF graph* is a set of triples and temporal triples. In other words, some triples in an stRDF graph might not be associated with a valid time.

We also assume the existence of temporal constants `NOW` and `UC` inspired from the literature of temporal databases [5]. `NOW` represents the current time and can appear in the beginning or the ending point of a period. It will be used in stSPARQL queries to be introduced in Section 3. `UC` means "Until Changed" and is used for introducing valid times of a triple that persist until they are explicitly terminated by an update. For example, when John becomes an associate professor in 1/1/2013 this is assumed to hold in the future until an update terminates this fact (e.g., when John is promoted to professor).

Example 1. The following stRDF graph consists of temporal triples that represent the land cover of an area in Spain for the time periods [2000, 2006) and [2006, UC) and triples which encode other information about this area, such as its code and the WKT serialization of its geometry extent. In this and following examples, namespaces are omitted for brevity. The prefix `strdf` stands for `http://strdf.di.uoa.gr/ontology` where one can find all the relevant datatype definitions underlying the model stRDF.

```
corine:Area_4 rdf:type corine:Area .
corine:Area_4 corine:hasID "EU-101324" .
corine:Area_4 corine:hasLandCover corine:coniferousForest
              "[2000-01-01T00:00:00,2006-01-01T00:00:00)"^^strdf:period .
corine:Area_4 corine:hasLandCover corine:naturalGrassland
              "[2006-01-01T00:00:00,UC)"^^strdf:period .
corine:Area_4 corine:hasGeometry "POLYGON((-0.66 42.34, ...))"^^strdf:WKT .
```

The stRDF graph provided above is written using the N-Quads format[2] which has been proposed for the general case of adding context to a triple. The graph has been extracted from a publicly available dataset provided by the European Environmental Agency (EEA) that contains the changes in the CORINE Land Cover dataset for the time period [2000, UC) for various European areas. According to this dataset, the area `corine:Area_4` has been a coniferous forest area until 2006, when the newer version of CORINE showed it to be natural grassland. Until the CORINE Land cover dataset is updated, `UC` is used to denote the persistence of land cover values of 2006 into the future. The last triple of the stRDF graph gives the WKT serialization of the geometry of the area (not all vertices of the polygon are shown due to space considerations). This dataset will be used in our examples but also in the experimental evaluation of Section 5.

[2] `http://sw.deri.org/2008/07/n-quads/`

3 Querying Valid Times Using stSPARQL

The query language stSPARQL is an extension of SPARQL 1.1. Its geospatial features have been presented in [12] and [14]. In this section we introduce for the first time the valid time dimension of stSPARQL. The new features of the language are:

Temporal Triple Patterns. Temporal triple patterns are introduced as the most basic way of querying temporal triples. A *temporal triple pattern* is an expression of the form `s p o t.`, where `s p o.` is a triple pattern and `t` is a time period or a variable.

Temporal Extension Functions. Temporal extension functions are defined in order to express temporal relations between expressions that evaluate values of the datatypes `xsd:dateTime` and `strdf:period`. The first set of such temporal functions are 13 Boolean functions that correspond to the 13 binary relations of Allen's Interval Algebra. stSPARQL offers nine functions that are "syntactic sugar" i.e., they encode frequently-used disjunctions of these relations.

There are also three functions that allow relating an instant with a period:

- `xsd:Boolean strdf:during(xsd:dateTime i2, strdf:period p1)`: returns true if instant `i2` is during the period `p1`.
- `xsd:Boolean strdf:before(xsd:dateTime i2, strdf:period p1)`: returns true if instant `i2` is before the period `p1`.
- `xsd:Boolean strdf:after(xsd:dateTime i2, strdf:period p1)`: returns true if instant `i2` is after the period `p1`.

The above point-to-period relations appear in [16]. The work described in [16] also defines two other functions allowing an instant to be equal to the starting or ending point of a period. In our case these can be expressed using the SPARQL 1.1. operator = (for values of `xsd:dateTime`) and functions `period_start` and `period_end` defined below.

Furthermore, stSPARQL offers a set of functions that construct new (closed-open) periods from existing ones. These functions are the following:

- `strdf:period strdf:period_intersect(period p1, period p2)`: This function is defined if `p1` intersects with `p2` and it returns the intersection of period `p1` with period `p2`.
- `strdf:period strdf:period_union(period p1, period p2)`: This function is defined if period `p1` intersects `p2` and it returns a period that starts with `p1` and finishes with `p2`.
- `strdf:period strdf:minus(period p1, period p2)`: This function is defined if periods `p1` and `p2` are related by one of the Allen's relations `overlaps, overlappedBy, starts, startedBy, finishes, finishedBy` and it returns the a period that is constructed from period `p1` with its common part with `p2` removed.
- `strdf:period strdf:period(xsd:dateTime i1, xsd:dateTime i2)`: This function constructs a (closed-open) period having instant `i1` as beginning and instant `i2` as ending time.

There are also the functions `strdf:period_start` and `strdf:period_end` that take as input a period p and return an output of type `xsd:dateTime` which is the beginning and ending time of the period `p` respectively.

Finally, stSPARQL defines the following functions that compute temporal aggregates:

- `strdf:period strdf:intersectAll(set of period p)`: Returns a period that is the intersection of the elements of the input set that have a common intersection.
- `strdf:period strdf:maximalPeriod(set of period p)`: Constructs a period that begins with the smallest beginning point and ends with the maximum endpoint of the set of periods given as input.

The query language stSPARQL, being an extension of SPARQL 1.1, allows the temporal extension functions defined above in the SELECT, FILTER and HAVING clause of a query. A complete reference of the temporal extension functions of stSPARQL is available on the Web[3].

Temporal Constants. The temporal constants `NOW` and `UC` can be used in queries to retrieve triples whose valid time has not ended at the time of posing the query or we do not know when it ends, respectively.

The new expressive power that the valid time dimension of stSPARQL adds to the version of the language presented in [14], where only the geospatial features were presented, is as follows. First, a rich set of temporal functions are offered to express queries that refer to temporal characteristics of some non-spatial information in a dataset (e.g., see Examples 2, 3 and 6 below). In terms of expressive power, the temporal functions of stSPARQL offer the expressivity of the qualitative relations involving points and intervals studied by Meiri [16]. However, we do not have support (yet) for quantitative temporal constraints in queries (e.g., $T_1 - T_2 \leq 5$). Secondly, these new constructs can be used together with the geospatial features of stSPARQL (geometries, spatial functions, etc.) to express queries on geometries that change over time (see Examples 4 and 5 below). The temporal and spatial functions offered by stSPARQL are orthogonal and can be combined with the functions offered by SPARQL 1.1 in arbitrary ways to query geospatial data that changes over time (e.g., the land cover of an area) but also moving objects [10] (we have chosen not to cover this interesting application in this paper).

In the rest of this section, we give some representative examples that demonstrate the expressive power of stSPARQL.

Example 2. Temporal selection and temporal constants. Return the current land cover of each area mentioned in the dataset.

```
SELECT ?clcArea ?clc
WHERE {?clcArea rdf:type corine:Area;
               corine:hasLandCover ?clc ?t . FILTER(strdf:during(NOW, ?t))}
```

This query is a temporal selection query that uses an extended Turtle syntax that we have devised to encode temporal triple patterns. In this extended syntax, the

[3] `http://www.strabon.di.uoa.gr/stSPARQL`

fourth element is optional and it represents the valid time of the triple pattern. The temporal constant `NOW` is also used.

Example 3. Temporal selection and temporal join. Give all the areas that were forests in 1990 and were burned some time after that time.

```
SELECT ?clcArea
WHERE{?clcArea rdf:type corine:Area ;
      corine:hasLandCover corine:ConiferousForest ?t1 ;
      corine:hasLandCover corine:BurnedArea ?t2 ;
      FILTER(strdf:during(?t1, "1990-01-01T00:00:00"^^xsd:dateTime) && strdf:after(?t2,?t1))}
```

This query shows the use of variables and temporal functions to join information from different triples.

Example 4. Temporal join and spatial metric function. Compute the area occupied by coniferous forests that were burnt at a later time.

```
SELECT ?clcArea (SUM(strdf:area(?geo)) AS ?totalArea)
WHERE {?clcArea rdf:type corine:Area;
                corine:hasLandCover corine:coniferousForest ?t1 ;
                corine:hasLandCover corine:burntArea ?t2 ;
                corine:hasGeometry ?geo .
       FILTER(strdf:before(?t1,?t2))} GROUP BY ?clcArea
```

In this query, a temporal join is performed by using the temporal extension function `strdf:before` to ensure that areas included in the result set were covered by coniferous forests *before* they were burnt. The query also uses the spatial metric function `strdf:area` in the SELECT clause of the query that computes the area of a geometry. The aggregate function `SUM` of SPARQL 1.1 is used to compute the total area occupied by burnt coniferous forests.

Example 5. Temporal join and spatial selection. Return the evolution of the land cover use of all areas contained in a given polygon.

```
SELECT ?clc1 ?t1 ?clc2 ?t2
WHERE {?clcArea rdf:type corine:Area ;
                corine:hasLandCover ?clc1 ?t1 ; corine:hasLandCover ?clc2 ?t2 ;
                clc:hasGeometry ?geo .
       FILTER(strdf:contains(?geo, "POLYGON((-0.66 42.34, ...))"^^strdf:WKT)
       FILTER(strdf:before(?t1,?t2))}
```

The query described above performs a temporal join and a spatial selection. The spatial selection checks whether the geometry of an area is contained in the given polygon. The temporal join is used to capture the temporal evolution of the land cover in pairs of periods that preceed one another .

Example 6. Update statement with temporal joins and period constructor.

```
UPDATE {?area corine:hasLandCover ?clcArea ?coalesced}
WHERE {SELECT (?clcArea AS ?area) ?clcArea (strdf:period_union(?t1,?t2) AS ?coalesced)
       WHERE {?clcArea rdf:type corine:Area ;
                       corine:hasLandCover ?clcArea ?t1; corine:hasLandCover ?clcArea ?t2 .
              FILTER(strdf:meets(?t1,?t2) || strdf:overlaps(?t1,?t2))}}
```

In this update, we perform an operation called *coalescing* in the literature of temporal relational databases: two temporal triples with exactly the same subject, predicate and object, and periods that overlap or meet each other can be "joined" into a single triple with valid time the union of the periods of the original triples [4].

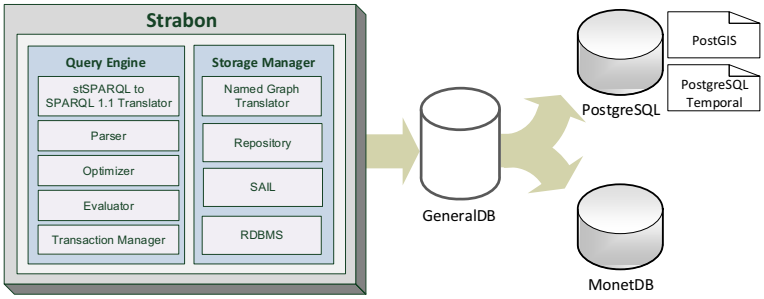


Fig. 1. Architecture of the system Strabon enhanced with valid time support

4 Implementation of Valid Time Support in Strabon

Figure 1 shows the architecture of the system Strabon presented in [14], as it has been extended for valid time support. We have added new components and extended existing ones as we explain below.

As described in [14], Strabon has been implemented by extending Sesame[4] 2.6.3 and using an RDBMS as a backend. Currently, PostgreSQL and MonetDB can be used as backends. To support the geospatial functionality of stSPARQL efficiently as we have shown in [14], Strabon uses PostGIS, an extension of PostgreSQL for storing and querying spatial objects and evaluating spatial operations. To offer support for the valid time dimension of stSPARQL discussed in this paper, the following new components have been added to Strabon.

Named Graph Translator. This component is added to the storage manager and translates the temporal triples of stRDF to standard RDF triples following the named graphs approach of [22] as we discuss below.

stSPARQL to SPARQL 1.1 Translator. This component is added to the query engine so that temporal triple patterns are translated to triple patterns as we discuss below.

PostgreSQL Temporal. This is a temporal extension of PostgreSQL which defines a `PERIOD` datatype and implements a set of temporal functions. This datatype and its associated functions come very handy for the implementation of the valid time suport in Strabon as we will see below. PostgreSQL Temporal also allows the use of a GiST index on `PERIOD` columns. Using this add-on, PostgreSQL becomes "temporally enabled" as it adds support for storing and querying `PERIOD` objects and for evaluating temporal functions.

Storing Temporal Triples. When a user wants to store stRDF data in Strabon, she makes them available in the form of an N-Quads document. This document is decomposed into temporal triples and each temporal triple is processed separately by the storage manager as follows. First, the temporal triple is translated into the named graph representation. To achieve this, a URI is created and it is assigned to a named graph that corrresponds to the validity period of the

[4] `http://www.openrdf.org/`

triple. To ensure that every distinct valid time of a temporal triple corresponds to exactly one named graph, the URI of the graph is constructed using the literal representation of the valid time annotation. Then, the stored triple in the named graph identified by this URI and the URI of the named graph is associated to its corresponding valid time by storing the following triple in the default graph: `(g, strdf:hasValidTime, t)` where `g` is the URI of the graph and `t` is the corresponding valid time. For example, temporal triple

```
corine:Area_4 corine:hasLandCover corine:naturalGrassland
              "[2000-01-01T00:00:00,2006-01-01T00:00:00)"^^strdf:period
```

will be translated into the following standard RDF triples:

```
corine:Area_4 corine:hasLandCover corine:naturalGrassland
corine:2000-01-01T00:00:00_2006-01-01T00:00:00 strdf:hasValidTime
                                  "[2000-01-01T00:00:00,2006-01-01T00:00:00)"^^strdf:period
```

The first triple will be stored in the named graph with URI `corine:2000-01-01T00:00:00_2006-01-01T00:00:00` and the second in the default graph. If later on another temporal triple with the same valid time is stored, its corresponding triple will end-up in the same named graph.

For the temporal literals found during data loading, we deviate from the default behaviour of Sesame by storing the instances of the `strdf:period` datatype in a table with schema *period_values(id int, value period)*. The attribute *id* is used to assign a unique identifier to each period and associate it to its RDF representation as a typed literal. It corresponds to the respective *id* value that is assigned to each URI after the dictionary encoding is performed. The attribute *value* is a temporal column of the `PERIOD` datatype defined in PostgreSQL Temporal. In addition, we construct a GiST index on the *value* column.

Querying Temporal Triples. Let us now explain how the query engine of Strabon presented in [14] has been extended to evaluate temporal triple patterns. When a temporal triple pattern is encountered, the query engine of Strabon executes the following steps. First, the stSPARQL to SPARQL 1.1 Translator converts each temporal triple pattern of the form `s p o t` into the graph pattern `GRAPH ?g s p o . ?g strdf:hasValidTime t.` where `s`, `p`, `o` are RDF terms or variables and `t` is either a variable or an instance of the datatypes `strdf:period` or `xsd:dateTime`. Then the query gets parsed and optimized by the respective components of Strabon and passes to the evaluator which has been modified as follows: If a temporal extension function is present, the evaluator incorporates the table *period_values* to the query tree and it is declared that the arguments of the temporal function will be retrieved from the *period_values* table. In this way, all temporal extension functions are evaluated in the database level using PostgresSQL Temporal. Finally, the RDBMS evaluation module has been extended so that the execution plan produced by the logical level of Strabon is translated into suitable SQL statements. The temporal extension functions are respectively mapped into SQL statements using the functions and operators provided by PostgreSQL Temporal.

5 Evaluation

For the experimental evaluation of our system, we used two different datasets. The first dataset is the GovTrack dataset[5], which consists of RDF data about US Congress. This dataset was created by Civic Impulse, LLC[6] and contains information about US Congress members, bills and voting records. The second dataset is the CORINE Land Cover changes dataset that represents changes for the period [2000, UC), which we have already introduced in Section 2.

The GovTrack dataset contains temporal information in the form of instants and periods, but in standard RDF format using reification. So, in the preprocessing step we transformed the dataset into N-Quads format. For example the 5 triples

```
congress_people:A000069 politico:hasRole _:node17d3oolkdx1 .
_:node17d3oolkdx1 time:from _:node17d3oolkdx2 .
_:node17d3oolkdx1 time:to _:node17d3oolkdx3 .
_:node17d3oolkdx2 time:at "2001-01-03"^^xs:date .
_:node17d3oolkdx3 time:at "2006-12-08"^^xs:date .
```

were transformed into a single quad:

```
congress_people:A000069 politico:hasRole _:node17d3oolkdx1
                        "[2001-01-03T00:00:00, 2006-12-08T00:00:00]"^^strdf:period .
```

The transformed dataset has a total number of 7,900,905 triples, 42,049 of which have periods as valid time and 294,636 have instants.

The CORINE Land Cover changes dataset for the time period [2000, UC) is publicly available in the form of shapefiles and it contains the areas that have changed their land cover between the years 2000 and 2006. Using this dataset, we created a new dataset in N-Quads form which has information about geographic regions such as: unique identifiers, geometries and periods when regions have a landcover. The dataset contains 717,934 temporal triples whose valid time is represented using the `strdf:period` datatype. It also contains 1,076,901 triples without valid times. Using this dataset, we performed temporal and spatial stSPARQL queries, similar to the ones provided in Section 3 as examples.

Our experiments were conducted on an Intel Xeon E5620 with 12MB L3 caches running at 2.4 GHz. The system has 24GB of RAM, 4 disks of striped RAID (level 5) and the operating system installed is Ubuntu 12.04. We ran our queries three times on cold and warm caches, for which we ran each query once before measuring the response time. We compare our system with the following implementations.

The Prolog-based implementation of AnQL. We disabled the inferencer and we followed the data model and the query language that is used in [15], e.g., the above quad is transformed into the following AnQL statement:

```
congress_people:A000069 politico:hasRole _:node1 :[2001-01-03, 2006-12-08] .
```

[5] `http://www.govtrack.us/data/rdf/`
[6] `http://www.civicimpulse.com/`

AllegroGraph. AllegroGraph offers a set of temporal primitives and temporal functions, extending their Prolog query engine, to represent and query temporal information in RDF. AllegroGraph does not provide any high level syntax to annotate triples with their valid time, so, for example, the GovTrack triple that we presented earlier was converted into the following graph:

```
congress_people:A000069 politico:hasRole _:node1 graph:2001-01-03T... .
graph:2001-01-03T...  allegro:starttime "2001-01-03T00:00:00"^^xsd:dateTime .
graph:2001-01-03T...  allegro:endtime "2001-01-03T00:00:00"^^xsd:dateTime .
```

As AllegroGraph supports the N-Quads format, we stored each triple of the dataset in a named graph, by assigning a unique URI to each valid time. Then, we described the beginning and ending times of the period that the named graph corresponds to, using RDF statements with the specific temporal predicates that are defined in AllegroGraph[7]. We used the AllegroGraph Free server edition[8] that allows us to store up to five million statements, so we could not store the full version of the dataset.

Naive implementation. We developed a baseline implementation by extending the Sesame native store with the named graph translators we use in Strabon so that it can store stRDF graphs and query them using stSPARQL queries. We also developed in Java the temporal extension functions that are used in the benchmarks. A similar implementation has been used as a baseline in [14] where we evaluated the geospatial features of Strabon.

We evaluate the performance of the systems in terms of query response time. We compute the response time for each query posed by measuring the elapsed time from query submission till a complete iteration over the results had been completed. We also investigate the scalability with respect to database size and complexity of queries.

We have conducted four experiments that are explained below. Twenty queries were used in the evaluation. Only two queries are shown here; the rest are omitted due to space considerations. However, all datasets and the queries that we used in our experimental evaluation are publicly available[9].

Experiment 1. In this experiment we ran the same query against a number of subsets of the GovTrack dataset of various size, as we wanted to test the scalability of all systems with respect to the dataset size. To achieve this, we created five instances of the GovTrack dataset, each one with exponentially increasing number of triples and quads. The query that is evaluated against these datasets is shown in Figure 2.

Figure 3(a) shows the results of this experiment. As the dataset size increases, more periods need to be processed and as expected, the query response time grows for all systems. This is expected, as posing queries against a large dataset is challenging for memory-based implementations. Interestingly, the AnQL response time in the query Q2 is decreased, when a temporal filter is added to the

[7] http://www.franz.com/agraph/support/documentation/current/temporal-tutorial.html

[8] http://www.franz.com/agraph/downloads/

[9] http://www.strabon.di.uoa.gr/temporal-evaluation/experiments.html

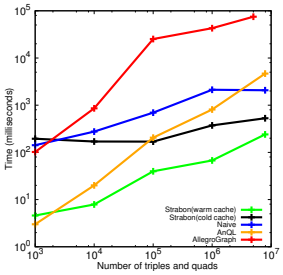
stSPARQL	AnQL	AllegroGraph
SELECT DISTINCT ?x ?name WHERE {?x gov:hasRole ?term ?t . OPTIONAL {?x foaf:name ?name .} FILTER(strdf:after(?t,"[...]"^^strdf:period))}	SELECT DISTINCT ?x ?name WHERE {?x gov:hasRole ?term ?t . OPTIONAL {?x foaf:name ?name .} FILTER(beforeany([[...]],?t))}	(select0-distinct (?x ?name) (q ?x !gov:hasRole ?term ?t) (optional (q ?x !foaf:name ?name)) (interval-after-datetimes ?t ".."))

Fig. 2. Query of Experiment 1

temporal graph pattern of the query Q1. The use of a very selective temporal filter reduces the number of the intermediate results. Also, it the implementation of AnQL performs better in workloads of up to 100,000 triples and quads, as it is a memory-based implementation. The poor performance of the baseline implementation compared to Strabon is reasonable, as Strabon evaluates the temporal extension functions in the RDBMS level using the respective functions of PostgreSQL Temporal and a GiST index on period values, while in the case of the baseline implementation a full scan over all literals is required. AllegroGraph is not charged with the cost of processing the high level syntax for querying the valid time of triples, like the other implementations, therefore it stores two triples to describe each interval of the dataset. This is one of the reasons that it gets outperformed by all other implementations. One can observe that Strabon achieves better scalability in large datasets than the other systems due to the reasons explained earlier. The results when the caches are warm are far better, as the intermediate results fit in main memory, so we have less I/O requests.

Experiment 2. We carried out this experiment to measure the scalability of all systems with respect to queries of varying complexity. The complexity of a query depends on the number and the type of the graph patterns it contains and their selectivity. We posed a set of queries against the GovTrack dataset and we increased the number of triple patterns in each query. As explained earlier, the AllegroGraph repository contains five million statements.

First, in Q2, we have a temporal triple pattern and a temporal selection on its valid time. Then, Q3 is formed by adding a temporal join to Q2. Then Q4 and Q5 are formed by adding some more graph patterns of low selectivity to Q3. Queries with low selectivity match with large graphs of the dataset and as a result the response time increases. This happens basically because in most cases the intermediate results do not fit in the main memory blocks that are available, requiring more I/O requests In the queries Q6 and Q7 we added graph patterns with high selectivity to the previous ones and the response time was decreased. This happened because of the highly selective graph patterns used. The respective response times in warm caches are far better, as expected. What is interesting in this case, is that while in cold caches the reponse time slightly increases from the query Q6 to the query Q7, in warm caches it decreases. This happens because with warm caches, the computational effort is widely reduced and the response time is more dependent of the number of the intermediate results which are produced. The query Q7 produces less intermediate results because it is more selective than Q6. AllegroGraph has the best performance in Q2, which contains only a temporal triple pattern, but when temporal functions are introduced (queries Q3-Q7), it performs worse than any other implementation. Obviously, the evaluation of a temporal join is very costly, as it internally



(a)

System	Q1	Q2	Q3	Q4	Q5	Q6	Q7
Strabon (warm caches)	98	67	92	109	131	35	31
Strabon (cold caches)	326	437	571	647	664	412	430
Naive	3056	5860	5916	6260	6462	2594	2604
AnQL	2028	1715	4275	4379	5802	6913	7472
AllegroGraph	1016	58155	91736	121835	154561	33824	156408

(b)

System	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16
Strabon (warm caches)	217	212	209	207	82	84	81	208	200
Strabon (cold caches)	485	381	377	370	250	248	248	285	376
Naive	6206	6148	6162	6196	6097	6248	6388	6332	6258

(c)

System	Q17	Q18	Q19	Q20
Strabon (warm caches)	3	13	12	208
Strabon (cold caches)	268	368	532	792
Naive	404	2868	2388	200705

(d)

Fig. 3. (a) Experiment 1: Query response time with respect to dataset size. (b), (c), (d) Experiments 2, 3, 4: Query response time in milliseconds.

maps the variables that take part in the temporal join to the respective intervals of the dataset, retrieves their beginning and ending timestamps and then evaluates the temporal operators. The AnQL implementation performs very well in queries of low selectivity but in queries of high selectivity it is outperformed by the baseline implementation. Strabon, even with cold caches, performs significantly better than the other implementations due to the efficient evaluation of the queries in the database level and the use of a temporal index.

Experiment 3. In this experiment we posed temporal queries against the GovTrack dataset in order to test the performance of different temporal operators in the FILTER clause of the query that are typically used to express a temporal join. The triple patterns in the queries posed (Q8-Q16) are identical so the queries differ only in the temporal function used in the FILTER clause of the query. For example query Q8 is the following:

```
SELECT DISTINCT ?x1 ?x2 WHERE {?x1 gov:hasRole ?term ?t1 .
                               ?x2 gov:hasRole ?term ?t2 . FILTER(strdf:during(?t1,?t2))}
```

The results of the experiment are shown in the table of Figure 3(c). For each system, the differences in performance with respect to the different temporal operators used in queries are minor, especially in the case of the naive implementation. As expected, Strabon continues to perform much better than the naive implementation as the implementation of each operator is more efficient.

Experiment 4. In this experiment we evaluate the spatiotemporal capabilities of Strabon and the baseline implementation . We used the CORINE Land Cover changes 2000-2006 dataset. This is a spatiotemporal dataset that contains more temporal triples, but there are only two distinct valid time values. Query Q17 retrieves the valid times of the temporal triples, while query Q18 is more selective

and performs a temporal join. Query Q19 is similar to Q20 but it also retrieves geospatial information so the response time is increased. Query 20 performs a temporal join and a spatial selection, so the reponse time is increased for both systems. Strabon peforms better because the temporal and the spatial operations are evaluated in the database level and the respective indices are used, while in the naive implementation these functions are implemented in Java.

6 Related Work

To the best of our knowledge, the only commercial RDF store that has good support for time is AllegroGraph[10]. AllegroGraph allows the introduction of points and intervals as resources in an RDF graph and their situation on a time line (by connecting them to dates). It also offers a rich set of predicates that can be used to query temporal RDF graphs in Prolog. As in stSPARQL, these predicates include all qualitative relations of [16] involving points and intervals. Therefore, all the temporal queries expressed using Prolog in AllegroGraph can also be expressed by stSPARQL in Strabon.

In [7] another approach is presented for extending RDF with temporal features, using a temporal element that captures more than one time dimensions. A temporal extension of SPARQL, named T-SPARQL, is also proposed which is based on TSQL2. Also, [17] presents a logic-based approach for extending RDF and OWL with valid time and the query language SPARQL for querying and reasoning with RDF, RDFS and OWL2 temporal graphs. To the best of our knowledge, no public implementation of [7] and [17] exists that we could use to compare with Strabon. Similarly, the implementations of [19] and [22] are not publicly available, so they could not be included in our comparison.

In stRDF we have not considered transaction time since the applications that motivated our work required only user-defined time and valid time of triples. The introduction of transaction time to stRDF would result in a much richer data model. We would be able to model not just the history of an application domain, but also the system's knowledge of this history. In the past the relevant rich semantic notions were studied in TSQL2 [20], Telos (which is very close to RDF) [18] and temporal deductive databases [21].

7 Conclusions

In future work, we plan to evaluate the valid time functionalities of Strabon on larger datasets, and continue the experimental comparison with AllegroGraph as soon as we obtain a license of its Enterprise edition. We will also study optimization techniques that can increase the scalability of Strabon. Finally, it would be interesting to define and implement an extension of stSPARQL that offers the ability to represent and reason with qualitative temporal relations in the same way that the Topology vocabulary extension of GeoSPARQL represents topological relations.

[10] `http://www.franz.com/agraph/allegrograph/`

References

1. Open Geospatial Consortium. OGC GeoSPARQL - A geographic query language for RDF data. OGC Candidate Implementation Standard (2012)
2. Al-Kateb, M., Ghazal, A., Crolotte, A., Bhashyam, R., Chimanchode, J., Pakala, S.P.: Temporal Query Processing in Teradata. In: ICDT (2013)
3. Allen, J.F.: Maintaining knowledge about temporal intervals. CACM 26(11) (1983)
4. Boelen, M.H., Snodgrass, R.T., Soo, M.D.: Coalescing in Temporal Databases. IEEE CS 19, 35–42 (1996)
5. Clifford, J., Dyreson, C., Isakowitz, T., Jensen, C.S., Snodgrass, R.T.: On the semantics of now in databases. ACM TODS 22(2), 171–214 (1997)
6. Date, C.J., Darwen, H., Lorentzos, N.A.: Temporal data and the relational model. Elsevier (2002)
7. Grandi, F.: T-SPARQL: a TSQL2-like temporal query language for RDF. In: International Workshop on Querying Graph Structured Data, pp. 21–30 (2010)
8. Gutierrez, C., Hurtado, C., Vaisman, A.: Temporal RDF. In: Gómez-Pérez, A., Euzenat, J. (eds.) ESWC 2005. LNCS, vol. 3532, pp. 93–107. Springer, Heidelberg (2005)
9. Gutierrez, C., Hurtado, C.A., Vaisman, A.: Introducing Time into RDF. IEEE TKDE 19(2), 207–218 (2007)
10. Güting, R.H., Böhlen, M.H., Erwig, M., Jensen, C.S., Lorentzos, N.A., Schneider, M., Vazirgiannis, M.: A foundation for representing and querying moving objects. ACM TODS 25(1), 1–42 (2000)
11. Hurtado, C.A., Vaisman, A.A.: Reasoning with Temporal Constraints in RDF. In: Alferes, J.J., Bailey, J., May, W., Schwertel, U. (eds.) PPSWR 2006. LNCS, vol. 4187, pp. 164–178. Springer, Heidelberg (2006)
12. Koubarakis, M., Karpathiotakis, M., Kyzirakos, K., Nikolaou, C., Sioutis, M.: Data Models and Query Languages for Linked Geospatial Data. In: Eiter, T., Krennwallner, T. (eds.) Reasoning Web 2012. LNCS, vol. 7487, pp. 290–328. Springer, Heidelberg (2012)
13. Koubarakis, M., Kyzirakos, K.: Modeling and Querying Metadata in the Semantic Sensor Web: The Model stRDF and the Query Language stSPARQL. In: Aroyo, L., et al. (eds.) ESWC 2010, Part I. LNCS, vol. 6088, pp. 425–439. Springer, Heidelberg (2010)
14. Kyzirakos, K., Karpathiotakis, M., Koubarakis, M.: Strabon: A Semantic Geospatial DBMS. In: Cudré-Mauroux, P., et al. (eds.) ISWC 2012, Part I. LNCS, vol. 7649, pp. 295–311. Springer, Heidelberg (2012)
15. Lopes, N., Polleres, A., Straccia, U., Zimmermann, A.: AnQL: SPARQLing Up Annotated RDFS. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010, Part I. LNCS, vol. 6496, pp. 518–533. Springer, Heidelberg (2010)
16. Meiri, I.: Combining qualitative and quantitative constraints in temporal reasoning. Artificial Intelligence 87(1-2), 343–385 (1996)
17. Motik, B.: Representing and Querying Validity Time in RDF and OWL: A Logic-Based Approach. Journal of Web Semantics 12-13, 3–21 (2012)
18. Mylopoulos, J., Borgida, A., Jarke, M., Koubarakis, M.: Telos: representing knowledge about information systems. ACM TIS (1990)
19. Perry, M.: A Framework to Support Spatial, Temporal and Thematic Analytics over Semantic Web Data. Ph.D. thesis, Wright State University (2008)

20. Snodgrass, R.T. (ed.): The TSQL2 Temporal Query Language. Springer (1995)
21. Sripada, S.M.: A logical framework for temporal deductive databases. In: Bancilhon, F., DeWitt, D. (eds.) VLDB, pp. 171–182. M. Kaufmann Publ. Inc. (1988)
22. Tappolet, J., Bernstein, A.: Applied Temporal RDF: Efficient Temporal Querying of RDF Data with SPARQL. In: Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 308–322. Springer, Heidelberg (2009)

When to Reach for the Cloud: Using Parallel Hardware for Link Discovery

Axel-Cyrille Ngonga Ngomo, Lars Kolb, Norman Heino, Michael Hartung, Sören Auer, and Erhard Rahm

Department of Computer Science, University of Leipzig
04109 Leipzig, Germany
{ngonga,kolb,heino,hartung,auer,rahm}@informatik.uni-leipzig.de

Abstract. With the ever-growing amount of RDF data available across the Web, the discovery of links between datasets and deduplication of resources within knowledge bases have become tasks of crucial importance. Over the last years, several link discovery approaches have been developed to tackle the runtime and complexity problems that are intrinsic to link discovery. Yet, so far, little attention has been paid to the management of hardware resources for the execution of link discovery tasks. This paper addresses this research gap by investigating the efficient use of hardware resources for link discovery. We implement the $\mathcal{HR}^3$ approach for three different parallel processing paradigms including the use of GPUs and MapReduce platforms. We also perform a thorough performance comparison for these implementations. Our results show that certain tasks that appear to require cloud computing techniques can actually be accomplished using standard parallel hardware. Moreover, our evaluation provides break-even points that can serve as guidelines for deciding on when to use which hardware for link discovery.

Keywords: Link discovery, MapReduce, GPU.

1 Introduction

Link Discovery (LD) is of central importance for realizing the fourth Linked Data principle [1]. With the growth of the Web of Data, the complexity of LD problems has grown considerably. For example, linking places from *LinkedGeoData* and *DBpedia* requires the comparison of hundreds of thousands of instances. Over the last years, several time-efficient algorithms such as *LIMES* [19], *MultiBlock* [9] and $\mathcal{HR}^3$ [18] have been developed to address the problem of the a-priori quadratic runtime of LD approaches. In general, these algorithms aim at minimizing the number of unnecessary similarity computations to carry out. While these approaches have been shown to outperform naïve LD implementations by several orders of magnitude, the sheer size of the number of links can still lead to unpractical runtimes. Thus, cloud implementations of some of these algorithms (e.g., *LIMESMR* [7] and Silk MapReduce[1]) have been recently developed. The speed-up of these implementations is, however, limited

[1] https://www.assembla.com/spaces/silk/wiki/Silk_MapReduce

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 275–289, 2013.

by a considerable input-output overhead that can lead to worse runtimes than on single machines. Interestingly, the use of standard parallel hardware has recently been shown to have the potential to outperform cloud computing techniques [6].

The multiplicity of available hardware solutions for carrying out LD led us to ask the following fundamental question: *When should which type of hardware be used to optimize the runtime of LD processes?* Providing an answer to this question promises to enable the development of highly flexible and scalable LD frameworks that can adapt to the available hardware environment. It will allow to decide intelligently upon when to reach for remote computing services such as cloud computing services in contrast to using local resources such as graphics processing units (GPUs) or multi-processor and multi-core technology. To answer our research question, we compare the runtimes of several implementations of $\mathcal{HR}^3$ for several datasets and find break-even points for different hardware. We chose the $\mathcal{HR}^3$ algorithm because it is the first algorithm with a guaranteed reduction ratio [18]. Thus, it promises to generate less overhead than other LD algorithms for comparable problems. Moreover, this algorithm can be used in manifold scenarios including LD, finding geographically related data (radial search) as well as search space reduction for other LD algorithms. The main contributions of this work are:

- We present the first implementation of a LD approach for GPUs. It relies on the GPU for fast parallel indexing and on the CPU for the computation of distances.
- We show how load-balancing for Map-Reduce can be carried out for LD approaches in affine spaces.
- We obtain guidelines for the use of different parallel hardware for LD by the means of a comparative evaluation of different implementations on real-world datasets from the Linked Open Data Cloud.

The remainder of the paper is organized as follows: We begin by giving a brief overview of $\mathcal{HR}^3$ and other paradigms used in this work. In Section 3, we then show how $\mathcal{HR}^3$ must be altered to run on GPUs. Section 4 focuses on the Map-Reduce implementation of $\mathcal{HR}^3$ as well as the corresponding load balancing approach. Section 5 presents a comparison of the runtimes of the different implementations of $\mathcal{HR}^3$ and derives break-even points for the different types of hardware[2]. The subsequent section gives an overview of related work. Finally, Section 7 summarizes our findings and presents future work.

2 Preliminaries

The specification of *link discovery* adopted herein is tantamount to the definition proposed in [18]. Given a formal relation[3] R and two (not necessarily disjoint) sets of instances S and T, the goal of link discovery is to find the set $M = \{(s,t) \in S \times T : R(s,t)\}$. Given that the explicit computation of R is usually a very complex endeavor, most frameworks reduce the computation of M to that of the computation of an approximation $\tilde{M} = \{(s,t) : \delta(s,t) \leq \theta\}$, where δ is a (complex) distance function and θ is a

[2] Details to the experiments and code are available at `http://limes.sf.net`.

[3] For example, `http://dbpedia.org/property/near`

distance threshold. Note that when $S = T$ and $R =$ `owl:sameAs`, the link discovery task becomes a *deduplication* task. Naïve approaches to computing $\tilde{M}$ have a quadratic time complexity, which is impracticable on large datasets. Consequently, a large number of approaches has been developed to reduce this time complexity (see [18] for an overview). Most of these approaches achieve this goal by optimizing their reduction ratio. In newer literature, the $\mathcal{HR}^3$ algorithm [17] has been shown to be the first algorithm which guarantees that it can achieve any possible reduction ratio.

$\mathcal{HR}^3$ builds upon the *HYPPO* algorithm presented in [16]. The rationale of $\mathcal{HR}^3$ is to maximize the reduction ratio of the computation of $\tilde{M}$ in affine spaces with Minkowski measures. To achieve this goal, $\mathcal{HR}^3$ computes an approximation of $\tilde{M}$ within a discretization of the space $\Omega = S \cup T$. Each point $\omega = (\omega_1, \ldots, \omega_n) \in \Omega$ is mapped to discrete coordinates $(\lfloor \omega_1/\Delta \rfloor, \ldots, \lfloor \omega_n/\Delta \rfloor)$, where $\Delta = \theta/\alpha$ and $\alpha \in \mathbb{N}\backslash\{0\}$ is called the granularity parameter. An example of such a discretization is shown in Figure 1: The point B with coordinates $(12.3436, 51.3339)$ is mapped to the discrete coordinates $(2468, 10226)$. The set of all points with the same discrete coordinates forms a hypercube (short: cube) of width α in the space Ω. The cube that contains ω is called $C(\omega)$. We call the vector $(c_1, \ldots, c_n) = (\lfloor \omega_1/\Delta \rfloor, \ldots, \lfloor \omega_n/\Delta \rfloor) \in \mathbb{N}^n$ the coordinates of $C(\omega)$.

Given the distance threshold θ and the granularity parameter α, $\mathcal{HR}^3$ computes the set of candidates $t \in T$ for each $s \in S$ by using the index function given in Eq. 1.

$$index(C, C') = \begin{cases} 0, \text{ if } \exists i : |c_i - c'_i| \leq 1 \text{ with } i \in \{1, ..., n\}, \\ \sum_{i=1}^{n} (|c_i - c'_i| - 1)^p \text{ else.} \end{cases} \tag{1}$$

where $C = C(s)$ and $C' = C(t)$ are hypercubes and p is the order of the Minkowski measure used in the space Ω.

Now, all source instances s are only compared with the target instances t such that $index(C(s), C(t)) \leq \alpha^p$. In our example, this is equivalent to computing the distance between B and all points contained in the gray-shadowed area on the map. Overall, $\mathcal{HR}^3$ achieve a reduction ratio of ≈ 0.82 on the data in Figure 1 as it only performs 10 comparisons instead of 55.

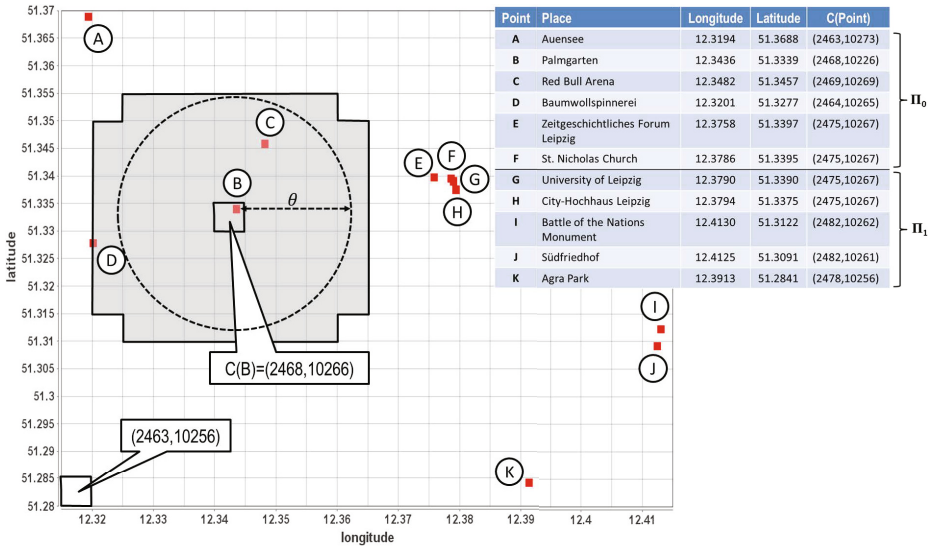
3 Link Discovery on GPUs

3.1 General-Purpose Computing on GPUs

GPUs were originally developed for processing image data. Yet, they have been employed for general-purpose computing tasks in recent years. Compared to CPUs the architecture of GPU hardware exhibits a large number of simpler compute cores and is thus referred to as *massively parallel*. A single compute core typically contains several arithmetic and logic units (ALU) that execute the same instruction on multiple data streams (SIMD).

Parallel code on GPUs is written as *compute kernels*, the submission of which is orchestrated by a host program executed on the CPU. Several frameworks exist for performing general purpose computing on GPUs. In this work we use *OpenCL*[4],

[4] `http://www.khronos.org/opencl/`



Point	Place	Longitude	Latitude	C(Point)	
A	Auensee	12.3194	51.3688	(2463,10273)	Π_0
B	Palmgarten	12.3436	51.3339	(2468,10226)	
C	Red Bull Arena	12.3482	51.3457	(2469,10269)	
D	Baumwollspinnerei	12.3201	51.3277	(2464,10265)	
E	Zeitgeschichtliches Forum Leipzig	12.3758	51.3397	(2475,10267)	
F	St. Nicholas Church	12.3786	51.3395	(2475,10267)	
G	University of Leipzig	12.3790	51.3390	(2475,10267)	Π_1
H	City-Hochhaus Leipzig	12.3794	51.3375	(2475,10267)	
I	Battle of the Nations Monument	12.4130	51.3122	(2482,10262)	
J	Südfriedhof	12.4125	51.3091	(2482,10261)	
K	Agra Park	12.3913	51.2841	(2478,10256)	

Fig. 1. Example dataset containing 11 places from Leipzig. To identify all points with a maximum Euclidean distance $\theta = 0.02$, the space is virtually tiled into hypercubes with an edge length of $\Delta = \theta/4$. A cube is identified by its coordinates $(c_1, ..., c_n)$. The gray-shadowed cells indicate the cubes whose points are compared with B, i.e., $\{C' \mid index(C(B), C') \leq \alpha^p\}$.

a vendor-agnostic industry standard. The memory model as exposed to OpenCL kernels is depicted in Figure 2: An instance of a compute kernel running on a device is called a *work item* or simply thread[5]. Work items are combined into *work groups*. All items within the same group have access to low-latency local memory and the ability to synchronize load/store operations using barriers. Thus, the actual number of kernel instances running in parallel is often limited by register and local memory usage. Each work item is assigned a globally (among all work items) and locally (within a work group) unique identifier, which also imposes a scheduling order. Typically those identifiers are used to compute local and global memory offsets for loading and storing data items that a given thread works on. Data transfer between host program and compute device is done via global device memory to which all work items have access, albeit with higher latency.

Threads on modern GPUs do not run in isolation. They are scheduled in groups of 64 or 32 work items depending on the hardware vendor. All threads within such a group execute the same instruction in lock-step. Any code path deviations due to control flow statements need to be executed by all items, throwing away unnecessary results (predication). It is therefore essential that each work item in such a group performs the same amount of work. The OpenCL framework does not expose the size of such groups to the API user. An upper bound is given by the work group size, which is always an integer multiple of the schedule group size.

[5] We use the therms work item and thread interchangeably in this work.

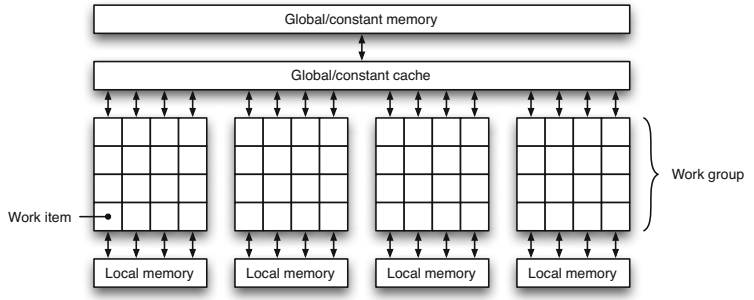


Fig. 2. OpenCL memory model

3.2 GPU-Based $\mathcal{HR}^3$ Implementation

For GPU-based computation all data must be copied to the device via the PCIe bus. We therefore only perform expensive computations on the device that benefit from the massive parallelism. In the case of $\mathcal{HR}^3$ this is the computation of the index function that determines which hypercubes a given cubes needs to be compared with. Since GPUs work best with regular memory access patterns a few preparation steps are needed. These are performed serially on the host. First, we discretize the input space $\Omega = S \cup T$, resulting in a set of hypercubes. All hypercubes are then sorted component-wise. The number of hypercubes determines the global work size. That is, each thread is assigned a hypercube (called *pivot cube*) determined by its global id. The work to be done by each thread is then to compute all those hypercubes that abide by the bound on indexes set by $\mathcal{HR}^3$.

A naïve implementation would have each thread compare its pivot cube to all other cubes, resulting in an amount of work quadratic in the number of hypercubes. A better approach is to minimize the amount of cube comparisons while maintaining an even work distribution among threads within the same group. Since hypercubes are globally sorted and fetched by work items in increasing schedule order, the ordering is maintained also locally. That is, let $g = k + 1$ be the local work group size. The work item with the least local id per group is assigned the smallest pivot cube C^0 while the last work item having the highest local id operates on the largest cube C^k as its pivot. Both work items therefore can determine a lower and upper bound for the whole group as follows. The first item computes the cube $C^{0-\alpha} = (c_1^0 - \alpha, \ldots, c_n^0 - \alpha)$ and the last item computes the cube $C^{k+\alpha} = (c_1^k + \alpha, \ldots, c_n^k + \alpha)$, where c_i^0 and c_i^k are the coordinates of the respective pivot cubes. Thread 0 then determines i_l, the index of the largest cube not greater then $C^{0-\alpha}$ while thread k computes i_u, the index of the smallest cube that is greater than $C^{k+\alpha}$. After a barrier synchronization that ensures all work items in a group can read the values stored by threads 0 and k, all work items compare their pivot cube to cubes at indices $i_l, \ldots (i_u - 1)$ in global device memory. Since all work items access the same memory locations fetches can be efficiently served from global memory cache.

In OpenCL kernels dynamic memory management is not available. That is, all buffers used during a computation must be allocated in advance by the host program. In particular, the size of the result buffer must be known before submitting a kernel to a device.

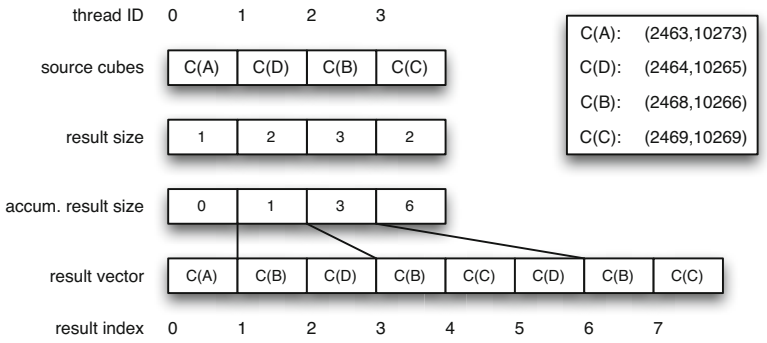


Fig. 3. Result index computation for $\mathcal{HR}^3$ on GPU hardware

We therefore cannot simply write the resulting cubes to an output vector. Instead, we compute results in two passes. During the first pass each thread writes the number of results it needs to produce to an output vector. A prefix sum over this vector yields at each index the accumulated number of results of threads with a lower id. This value can be used as an index into the final output vector at which each thread can start writing its results.

As an example consider Figure 3. It shows four threads ($0 \ldots 3$), each of which loads a single cube from the sorted *source cubes vector*. The index from which each threads loads its cube is given by its id[6]. In this example we assume a granularity factor of $\alpha = 4$. For thread 1 the smallest cube its pivot cube needs to be compared with is $C(D) = (2464, 10265)$ while the largest is $C(B) = (2468, 10266)$. It therefore writes 2 into an output vector, again using its thread id as an index. Thread 0 as well as 2 and 3 do the same, which results in the *result size vector* as depicted in Figure 3. In order to determine the final indexes each thread can use for storing its results in the result vector, an exclusive prefix sum is computed over the result size vector. This operation computes at each index i the sum of the elements at indexes $0 \ldots (i-1)$, resulting in the *accumulated result size vector*. A result vector of the appropriate size is allocated and in a second kernel run each thread can now write the cube coordinates starting at the index computed in the previous step. Indexing results are then copied back to the host where comparison of the actual input points is carried out. Since this operation is dominated by the construction of the result it cannot be significantly improved on parallel hardware.

4 MapReduce-Based Link Discovery

In this section we present an implementation of $\mathcal{HR}^3$ with MapReduce (MR), a programming model designed for parallelizing data-intensive computing in cluster environments [2]. MR implementations like *Apache Hadoop* rely on a distributed file system (DFS) that can be accessed by all nodes. Data is represented by key-value pairs

[6] For means of readability we show only one id per thread that serves as both its local and global id.

Algorithm 1. Basic $\mathcal{HR}^3$ - Map

```
1  map(k_in=unused, v_in = ω)
2      Δ ← θ/α;
3      cid_1 ← getCubeId(C(ω));
4      RC ← getRelatedCubes(C(ω), Δ);
5      foreach C' ∈ RC do
6          cid_2 ← getCubeId(C');
7          if cid_1 ≤ cid_2 then
8              output(cid_1.cid_2.0,
9                  (ω, 0));
10         else
11             output(cid_2.cid_1.1,
12                 (ω, 1));
   // part = hash(cid_1,cid_2) mod r
   // sort component-wise by entire key
   // group by cid_1, cid_2
```

Algorithm 2. Basic $\mathcal{HR}^3$ - Reduce

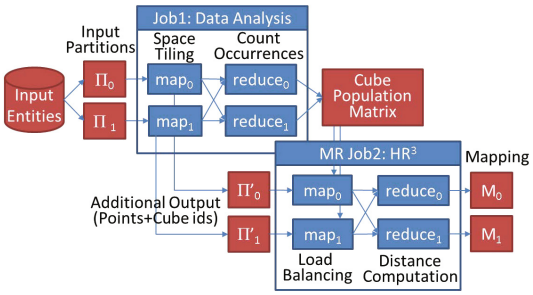
```
1  reduce(k_tmp=cid_1, cid_2,
2    v_tmp=list< ω, flag) >)
3      buf ← {};
4      if cid_1 = cid_2 then
5          foreach (ω, flag) ∈ v_tmp do
6              foreach ω' ∈ buf do
7                  compare(ω, ω');
8              buf ← buf ∪ {ω};
9      else
10         foreach (ω, flag) ∈ v_tmp do
11             if flag=0 then
12                 buf ← buf ∪ {ω};
13             else
14                 foreach ω' ∈ buf do
15                     compare(ω,ω');
```

and a computation is expressed employing two user-defined functions, `map` and `reduce`, which are processed by a fixed number of map (m) and reduce tasks (r). For each intermediate key-value pair produced in the map phase, a target reduce task is determined by applying a partitioning function that operates on the pair's key. The reduce tasks first sort incoming pairs by their intermediate keys. The sorted pairs are then grouped and the reduce function is invoked on all adjacent pairs of the same group.

We describe a straightforward realization of $\mathcal{HR}^3$ as well as an advanced approach that considers skew handling to guarantee load balancing and to avoid unnecessary data replication. In favor of readability, we consider a single dataset only.

4.1 $\mathcal{HR}^3$ with MapReduce

$\mathcal{HR}^3$ can be implemented with a single MR job. The main idea is to compare the points of two related cubes within a single reduce call. We call two cubes C, C' *related* iff $index(C, C') \leq \alpha^p$. For each input point ω, the map function determines the surrounding cube $C(\omega)$ and the set of related cubes RC, which might contain points within the maximum distance. For each cube $C' \in RC$, map outputs a $(\mathsf{cid}_1 \odot \mathsf{cid}_2 \odot \mathsf{flag}, (\mathsf{p}, \mathsf{flag}))$ pair with a composite key and the point itself as value. The first two components of the key identify the two involved cubes using textual cube ids: $\mathsf{cid}_1 = \min\{C(\omega).id, C'.id\}$ and $\mathsf{cid}_2 = \max\{C(\omega).id, C'.id\}$. The flag indicates whether ω belongs to the first or to the second cube. The repartitioning of the output key-value pairs is done by applying a hash function on the first two key components. This assigns all points of $C(\omega) \cup C'$ to the same reduce task. All key-value pairs are sorted by their complete keys. Finally, the reduce function is invoked on all values whose first two key components are equal. In reduce, the actual distance computation takes place. Due to the sorting, it is ensured that all points of the cube with the smaller cube id are processed first allowing for an efficient comparison of points of different cubes. The pseudo-code of the $\mathcal{HR}^3$ implementation is shown in Algorithms 1 and 2.



(a) $\mathcal{HR}^3$ with Load Balancing for MR

	Cube index	Partition Π_0	Partition Π_1
C_0	(2463,10273)	1	0
C_1	(2468,10226)	1	0
C_2	(2469,10269)	1	0
C_3	(2464,10265)	1	0
C_4	(2475,10267)	2	2
C_5	(2482,10262)	0	1
C_6	(2482,10261)	0	1
C_7	(2478,10256)	0	1

Cubes

(b) CPM

Fig. 4. Overview of the MR-based $\mathcal{HR}^3$ implementation with load balancing (left) and the cube population matrix for the example dataset with $m = 2$ (right)

The described approach has two major drawbacks. First, a map task operates only on a fraction of the input data without global knowledge about the overall data distribution. Thus, each point is replicated and repartitioned $|RC|$ times, independently of whether there are points in the related cubes or not. Second, this approach is vulnerable to data skew, i.e., due to the inherent quadratic time complexity varying cube sizes can lead to severe load imbalances of the reduce tasks. Depending on the problem size and the granularity of the space tiling, the scalability of the described approach might be limited to a few nodes only. We provide an advanced approach that addresses these drawbacks in the next section.

4.2 $\mathcal{HR}^3$ with Load Balancing

The advanced approach borrows ideas from the load balancing approaches for Entity Resolution presented in [12]. An overview is shown in 4(a). The overall idea is to schedule a light-weight analysis MR job that linearly scans the input data in parallel and collects global data statistics. The second MR job utilizes these statistics for a data-driven redistribution of points ensuring evenly loaded reduce tasks.

Data Analysis Job. The first job calculates the cube index of each point in the map phase and sums up the number of points per (non-empty) cube in reduce. The output is a cube population matrix (CPM) of size $c \times m$ that specifies the number of points of c cubes across m input partitions. For our running example, an analysis job with $m = 2$ map tasks would read data from two input partitions Π_0 and Π_1 (cf. table in Figure 1) and produce the CPM shown in 4(b).

Distance Computation Job. The second MR job is based on the same number of map tasks and the same partitioning of the input data. At initialization, each map task reads the CPM. Similar to the basic approach, the reduce function processes pairs of related cubes. Because the CPM allows for an easy identification of empty cubes, the number of intermediate key-value pairs can be reduced significantly. As an example, for point B of the running example, the map function of the basic approach would output 77 key-value pairs. With the knowledge encoded in the CPM, this can be reduced to two pairs only, i.e., for computing B's distances to the points C and D, respectively.

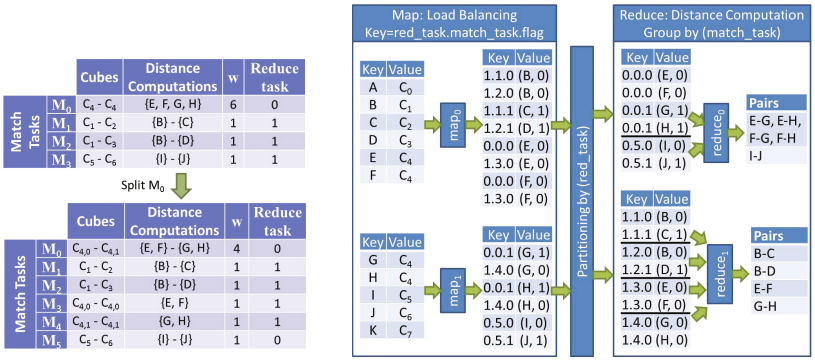


Fig. 5. Match task creation and reduce task assignment with/without splitting of large tasks (left). Example data flow for second MR job (right)

Before processing the first input point, each map tasks constructs a list of so-called match tasks. A match task is a triple (C_i, C_j, w), where C_i, C_j are two related cubes and $w = |C_i| \cdot |C_j|$ ($w = |C_i| \cdot (|C_i| - 1)/2$ for $i = j$) is the corresponding workload. The overall workload W is the sum of the workload of all match tasks. To determine each match task's target reduce task, the list is sorted in descending order of the workload. In this order, match tasks are assigned to the r reduce tasks following a greedy heuristic, i.e., the current match task is assigned to the reduce task with the currently lowest overall workload. The resulting match tasks are shown on the top of 5(a). Obviously, the reduce tasks are still unevenly loaded, because a major part of the overall workload is made up by the match task $C_4 - C_4$. To address this, for each large match task $M = (C_i, C_j, w)$ with $w > W/r$, both cubes are split according to their input partitioning into m subcubes. Consequently, M is split into a set of smaller subtasks, each comprising a pair of split subcubes before the sorting and reduce task assignment takes place. The bottom of 5(a) illustrates the splitting of the large match task $(C_4 - C_4)$. Because its workload $w = 6$ exceeds the average reduce task workload of $9/2 = 4.5$, C_4 is split into two subcubes $C_{4,0}$ (containing E, F) and $C_{4,1}$ (containing G, H). This results in three subtasks $(C_{4,0}, C_{4,0}, 1)$, $(C_{4,1}, C_{4,1}, 1)$, and $(C_{4,0}, C_{4,1}, 4)$ that recompose the original match task. Thus, both reduce tasks compute approximately the same number of distances indicating a good load balancing for the example.

After the initial match task creation, map task i builds an index that maps a cube to a set of corresponding match tasks. Thereby, only cubes of whom the input partition i actually contains points, need to be considered. For each input point ω and each match task of the cube $C(\omega)$, the map function outputs a $(\mathsf{red\_task} \odot \mathsf{match\_task} \odot \mathsf{flag}, (\omega, \mathsf{flag}))$ pair. Again, the flag indicates to which of the match task's (possibly split) cubes ω belongs to. The partitioning is only based on the reduce task index. The sorting is performed on the entire key, whereas the grouping is done by match task index. 5(b) illustrates the dataflow for the running example. Note, that due to the enumeration of the match tasks and the sorting behavior, it is ensured that the largest match tasks are processed first. This makes it unlikely that larger delays occur at the end of the computation when most nodes are already idle.

Dataset	Source	Size	Features
DS_1	DBPedia	25,781	min/medium/max elevation
DS_2	DBPedia	475,000	latitude, longitude
DS_3	Linked Geo Data	500,000	latitude, longitude
DS_4	Linked Geo Data	6,000,000	latitude, longitude

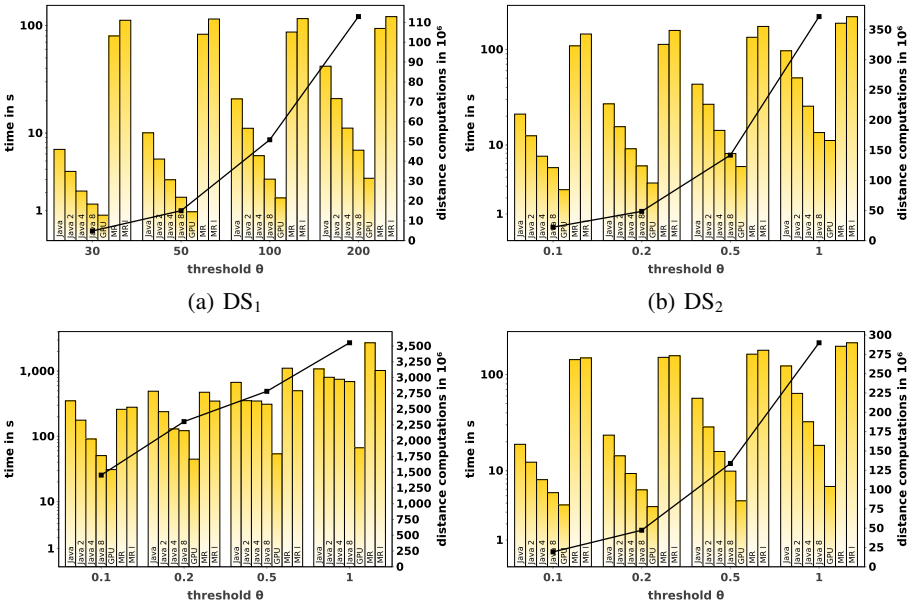
Fig. 6. Datasets used for evaluation

5 Evaluation

The aim of our evaluation was to discover break-even points for the use of parallel processor, GPU and cloud implementations of LD algorithms. For this purpose, we compared the runtimes of the implementations of $\mathcal{HR}^3$ presented in the previous sections on four data sets within two series of experiments. The goal of the first series of experiment was to compare the performance of the approaches for link discovery problems of common size. Thereafter, we carried out a scalability evaluation on a large dataset to detect break-even points of the implementations. In the following, we present the datasets we used as well as the results achieved by the different implementations.

5.1 Experimental Setup

We utilized the four datasets of different sizes shown in Figure 6. The small dataset DS_1 contains place instances having three elevation features. The medium-sized datasets DS_2 and DS_3 contain instances with geographic coordinates. For the scalability experiment we used the large dataset DS_3 and varied its size up to $6 \cdot 10^6$. Throughout all experiments we considered the Euclidean distance. Given the spectrum of implementations at hand, we ran our experiments on three different platforms. The *CPU experiments* (Java, $Java_2$, $Java_4$, $Java_8$ for 1, 2, 4 and 8 cores) were carried out on a 32-core server running JDK 1.7 on Linux 10.04. The processors were 8 quad core AMD Opteron 6128 clocked at 2.0 GHz. The *GPU experiments* (GPU) were performed on an average consumer workstation. The GPU was a AMD Radeon 7870 GPU with 20 compute units, each of which has the ability to schedule up to 64 parallel hardware threads. The host program was executed on a Linux workstation running Ubuntu 12.10 and AMD APP SDK 2.8. The machine had an Intel Core i7 3770 CPU and 8 GB of RAM. All C++ code was compiled with gcc 4.7.2. Given that C++ and Java are optimized differently, we also ran the Java code on this machine and computed a runtime ratio that allowed our results to remain compatible. The *MapReduce experiments* (basic: MR, load balanced: MR_l) were performed with the *Dedoop prototype* [11] on Amazon EC2 in EU-west location. For the first experiment we used 10 nodes of type c1.medium (2 virtual cores, 1.7 GB memory). For the large data set we employed 20 nodes of type c1.xlarge (8 virtual cores, 7 GB memory).



(a) DS_1 (b) DS_2

(c) DS_3 (d) DS_2–DS_3

Fig. 7. Comparison of runtimes for Experiment 1

5.2 Performance Comparison

The results of our performance comparison are shown in Figure 7. While the parallel implementation of $\mathcal{HR}^3$ on CPUs scales linearly for uniformly distributed data, the considerable skew in the DS_3 data led to the 8-core version being only 1.6 times faster than the mono-core implementation with a threshold of 1°. This impressively demonstrates the need for load balancing in all parallel link discovery tasks on skewed data. This need is further justified by the results achieved by MR and MR_l on DS_3. Here, MR_l clearly outperforms MR and is up to 2.7 times faster. Still, the most important result of this series of experiments becomes evident after taking a look at the GPU and Java runtimes on the workstation.

Most importantly, the massively parallel implementation outperforms all other implementations significantly. Especially, the GPU implementation outperforms the MR and MR_l by one to two orders of magnitude. Even the $Java_8$ implementation is outperformed by up to one order of magnitude. The performance boost of the GPU is partly due to the different hardware used in the experiments. To measure the effect of the hardware, we ran the server Java program also on the workstation. A comparison of the runtimes achieved during this rerun shows that the workstation is between 2.16 and 7.36 times faster than the server. Still, our results suggests that our massively parallel implementation can make an effective use of the underlying architecture to outperform all other implementations in the indexing phase. The added efficient implementation of float operations for the distance computation in C++ leads to an overall superior performance of the GPU. Here, the results can be regarded as conclusive with respect to

MR and MR_l and clearly suggest the use of local parallelism when dealing with small to average-sized link discovery problems.

The key observation that leads to conclusive results when comparing GPU and CPU results is that the generation of the cube index required between 29.3% (DS_1, $\theta = 50m$) and 74.5% (DS_3, $\theta = 1°$) of the total runtime of the algorithm during the deduplication tasks. Consequently, while running a parallel implementation on the CPU is advisable for small datasets with small thresholds for which the index computation makes up a small percentage of the total computation, running the approach on medium-sized datasets or with larger thresholds should be carried out on the GPU. This conclusion is yet only valid as long as the index fits into the memory of the GPU, which is in most cases 4 to 8 times smaller than the main memory of workstations. Medium-sized link discovery tasks that do not fit in the GPU memory should indeed be carried out on the CPUs. Our experiments suggest a break-even point between CPU and GPU for result set sizes around 10^8 pairs for 2-dimensional data. For higher-dimensional data where the index computation is more expensive, the break-even point is reached even for problems smaller than DS_1.

5.3 Scalability: Data Size

The strengths of the cloud are revealed in the second series of experiments we performed (see Figure 8). While the DFS and data transfer overhead dominates the total runtime of the LD tasks on the small datasets, running the scalability experiments on 20 nodes reveals that for tasks which generate more than 12 billion pairs as output, MR_l outperforms our local Java implementation. Moreover, we ran further experiments with more than 20 nodes on the 6 million data items. Due to its good scalability, the cloud implementation achieves the runtime of the GPU or performs even better for more nodes, e.g., for 30 (50) nodes MR_l requires approx. 32min (23min). It is important to remember here that the GPU implementation runs the comparisons in the CPU(s). Thus, the above suggested break-even point will clearly be reached for even smaller dataset sizes with more complex similarity measures such as the Levenshtein distance or the trigram similarity. Overall, our results hint towards the use of local massively parallel hardware being sufficient for a large number of link discovery tasks that seemed to require cloud infrastructures. Especially, numeric datasets can be easily processed locally as they require less memory than datasets in which strings play the central role. Still, for LD tasks whose intermediate results go beyond 10^{10} pairs, the use of the cloud still remains the most practicable solution. The clue for deciding which approach to use lies in having an accurate approximation function for the size of the intermediate results. $\mathcal{HR}^3$ provides such a function and can ensure that it can achieve an approximation below or equal to any possible error margin. Providing such guarantees for other algorithms would thus allow deciding effectively and conclusively when to reach for the cloud.

6 Related Work

Link discovery has become an important area of research over the last few years. Herein, we present a brief overview of existing approaches.[7] Overall, the two main problems

[7] See [17,10] for more extensive presentations of the state of the art.

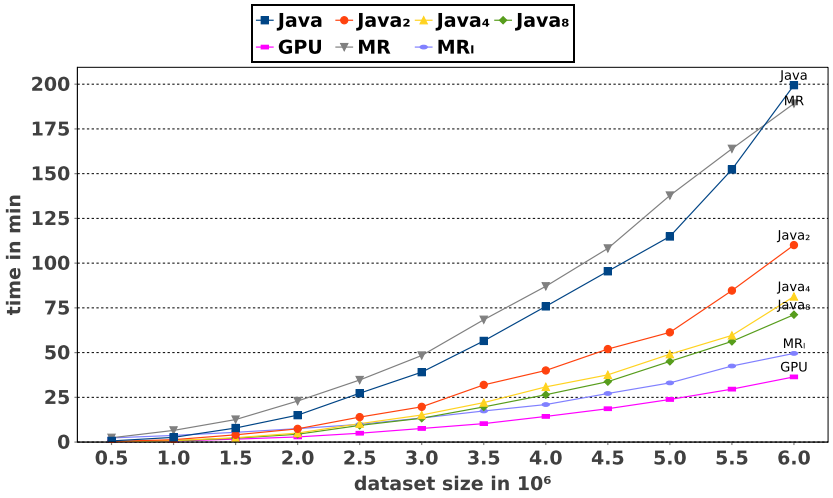


Fig. 8. Comparison of runtimes on DS4

time complexity and generation of link specifications have been at the core of the research on LD.

With regard to *time complexity*, time-efficient string comparison algorithms such as *PPJoin+* [26], *EDJoin* [25] that were developed for deduplication were integrated into several link discovery frameworks such as *LIMES* [18]. Moreover, dedicated time-efficient approaches were developed for LD. For example in [19], an approach based on the Cauchy-Schwarz inequality is presented. The approaches HYPPO [16] and $\mathcal{HR}^3$ [17] rely on space tiling in spaces with measures that can be split into independent measures across the dimensions of the problem at hand. Especially, $\mathcal{HR}^3$ was shown to be the first approach that can achieve a relative reduction ratio r' less or equal to any given relative reduction ratio $r > 1$. Standard blocking approaches were implemented in the first versions of SILK and later replaced with *MultiBlock* [9], a lossless multi-dimensional blocking technique. *KnoFuss* [22] also implements blocking techniques to achieve acceptable runtimes. Further LD frameworks have been participated in the ontology alignment evaluation initiative [4].

With regard to the *generation of link specifications*, some unsupervised techniques were newly developed (see, e.g., [22]), but most of the approaches developed so far abide by the paradigm of supervised machine learning. For example, the approach presented in [8] relies on large amounts of training data to detect accurate link specification using genetic programming. *RAVEN* [20] is (to the best of our knowledge) the first active learning technique for LD. The approach was implemented for linear or Boolean classifiers and shown to require a small number of queries to achieve high accuracy. Later, approaches combining active learning and genetic programming for LD were developed [10,21].

The entity resolution (ER) problem (see [14,3] for surveys) shares many similarities with link discovery. The MR programming model has been successfully applied for both ER and LD. [23] proposes a MR implementation of the *PPJoin+* algorithm

for large datasets. A first application for MR-based duplicate detection was presented in [24]. In addition, [7] as well as *Silk MapReduce*[8] implement MR approaches for LD. Several MR implementations for blocking-based ER approaches have been investigated so far. An MR implementation of the popular sorted neighborhood strategy is presented in [13]. Load balancing for clustering-based similarity computation with MR was considered in [12]. The ER framework *Dedoop* [11] allows to specify advanced ER strategies that are transformed to executable MR workflows and submitted to Hadoop clusters.

Load balancing and skew handling are well-known problems for parallel data processing but have only recently gained attention for MapReduce. *SkewTune* [15] is a generic load balancing approach that is invoked for a MapReduce job as soon as the first map (reduce) process becomes idle and no more map (reduce) tasks are pending. Then, the remaining keys (keygroups) of running tasks are tried to redistribute so that the capacity of the idle nodes is utilized. The approach in [5] is similar to our previous load balancing work [12] as it also relies on cardinality estimates determined during the map phase of the computation.

7 Conclusion and Future Work

In this paper, we presented a comparison of the runtimes of various implementations of the same link discovery approach on different types of parallel hardware. In particular, we compare parallel CPU, GPU and MR implementations of the $\mathcal{HR}^3$ algorithm. Our results show that the CPU implementation is most viable for two-dimensional problems whose result set size is in the order of 10^8. For higher-dimensional problems, massively parallel hardware preforms best even for problem with results set sizes in the order of 10^6. Cloud implementations become particularly viable as soon as the result set sizes reach the order of 10^{10}. Our results demonstrate that efficient resource management for link discovery demands the development of accurate approaches for determining the size of the intermediate results of link discovery frameworks. $\mathcal{HR}^3$ provides such a function. Thus, in future work, we will aim at developing such approximations for string-based algorithms. Moreover, we will apply the results presented herein to develop link discovery approaches that can make flexible use of the hardware landscape in which they are embedded.

References

1. Auer, S., Lehmann, J., Ngonga Ngomo, A.-C.: Introduction to Linked Data and Its Lifecycle on the Web. In: Polleres, A., d'Amato, C., Arenas, M., Handschuh, S., Kroner, P., Ossowski, S., Patel-Schneider, P. (eds.) Reasoning Web 2011. LNCS, vol. 6848, pp. 1–75. Springer, Heidelberg (2011)
2. Dean, J., Ghemawat, S.: MapReduce: Simplified Data Processing on Large Clusters. Commun. ACM 51(1), 107–113 (2008)
3. Elmagarmid, A.K., Ipeirotis, P.G., Verykios, V.S.: Duplicate record detection: A survey. IEEE Trans. Knowl. Data Eng. 19(1), 1–16 (2007)

[8] `https://www.assembla.com/spaces/silk/wiki/Silk_MapReduce`

4. Euzenat, J., Ferrara, A., van Hage, W.R., et al.: Results of the Ontology Alignment Evaluation Initiative 2011. In: OM (2011)
5. Gufler, B., Augsten, N., Reiser, A., Kemper, A.: Load Balancing in MapReduce Based on Scalable Cardinality Estimates. In: ICDE, pp. 522–533 (2012)
6. Heino, N., Pan, J.Z.: RDFS Reasoning on Massively Parallel Hardware. In: Cudré-Mauroux, P., et al. (eds.) ISWC 2012, Part I. LNCS, vol. 7649, pp. 133–148. Springer, Heidelberg (2012)
7. Hillner, S., Ngonga Ngomo, A.C.: Parallelizing LIMES for large-scale link discovery. In: I-SEMANTICS, pp. 9–16 (2011)
8. Isele, R., Bizer, C.: Learning Linkage Rules using Genetic Programming. In: OM (2011)
9. Isele, R., Jentzsch, A., Bizer, C.: Efficient Multidimensional Blocking for Link Discovery without losing Recall. In: WebDB (2011)
10. Isele, R., Jentzsch, A., Bizer, C.: Active Learning of Expressive Linkage Rules for the Web of Data. In: Brambilla, M., Tokuda, T., Tolksdorf, R. (eds.) ICWE 2012. LNCS, vol. 7387, pp. 411–418. Springer, Heidelberg (2012)
11. Kolb, L., Thor, A., Rahm, E.: Dedoop: Efficient Deduplication with Hadoop. PVLDB 5(12), 1878–1881 (2012)
12. Kolb, L., Thor, A., Rahm, E.: Load Balancing for MapReduce-based Entity Resolution. In: ICDE, pp. 618–629 (2012)
13. Kolb, L., Thor, A., Rahm, E.: Multi-pass Sorted Neighborhood blocking with MapReduce. Computer Science - R&D 27(1), 45–63 (2012)
14. Köpcke, H., Rahm, E.: Frameworks for entity matching: A comparison. Data Knowl. Eng. 69(2), 197–210 (2010)
15. Kwon, Y., Balazinska, M., Howe, B., Rolia, J.A.: SkewTune: Mitigating Skew in MapReduce Applications. In: SIGMOD Conference, pp. 25–36 (2012)
16. Ngonga Ngomo, A.C.: A Time-Efficient Hybrid Approach to Link Discovery. In: OM (2011)
17. Ngonga Ngomo, A.-C.: Link Discovery with Guaranteed Reduction Ratio in Affine Spaces with Minkowski Measures. In: Cudré-Mauroux, P., et al. (eds.) ISWC 2012, Part I. LNCS, vol. 7649, pp. 378–393. Springer, Heidelberg (2012)
18. Ngonga Ngomo, A.C.: On Link Discovery using a Hybrid Approach. Journal on Data Semantics 1, 203–217 (2012)
19. Ngonga Ngomo, A.C., Auer, S.: LIMES - A Time-Efficient Approach for Large-Scale Link Discovery on the Web of Data. In: IJCAI, pp. 2312–2317 (2011)
20. Ngonga Ngomo, A.C., Lehmann, J., Auer, S., Höffner, K.: RAVEN – Active Learning of Link Specifications. In: OM (2011)
21. Ngonga Ngomo, A.-C., Lyko, K.: EAGLE: Efficient Active Learning of Link Specifications Using Genetic Programming. In: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 149–163. Springer, Heidelberg (2012)
22. Nikolov, A., d'Aquin, M., Motta, E.: Unsupervised Learning of Data Linking Configuration. In: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 119–133. Springer, Heidelberg (2012)
23. Vernica, R., Carey, M.J., Li, C.: Efficient parallel set-similarity joins using mapreduce. In: SIGMOD Conference, pp. 495–506 (2010)
24. Wang, C., Wang, J., Lin, X., et al.: MapDupReducer: Detecting Near Duplicates over Massive Datasets. In: SIGMOD Conference, pp. 1119–1122 (2010)
25. Xiao, C., Wang, W., Lin, X.: Ed-Join: an efficient algorithm for similarity joins with edit distance constraints. PVLDB 1(1), 933–944 (2008)
26. Xiao, C., Wang, W., Lin, X., Yu, J.X.: Efficient similarity joins for near duplicate detection. In: WWW, pp. 131–140 (2008)

No Size Fits All – Running the Star Schema Benchmark with SPARQL and RDF Aggregate Views

Benedikt Kämpgen and Andreas Harth

Institute AIFB, Karlsruhe Institute of Technology, Karlsruhe, Germany
{benedikt.kaempgen,harth}@kit.edu

Abstract. Statistics published as Linked Data promise efficient extraction, transformation and loading (ETL) into a database for decision support. The predominant way to implement analytical query capabilities in industry are specialised engines that translate OLAP queries to SQL queries on a relational database using a star schema (ROLAP). A more direct approach than ROLAP is to load Statistical Linked Data into an RDF store and to answer OLAP queries using SPARQL. However, we assume that general-purpose triple stores – just as typical relational databases – are no perfect fit for analytical workloads and need to be complemented by OLAP-to-SPARQL engines. To give an empirical argument for the need of such an engine, we first compare the performance of our generated SPARQL and of ROLAP SQL queries. Second, we measure the performance gain of RDF aggregate views that, similar to aggregate tables in ROLAP, materialise parts of the data cube.

Keywords: Linked Data, OLAP, Star Schema Benchmark, View.

1 Introduction

Analytical queries using SPARQL on RDF have gained interest since large amounts of statistics have been published as Linked Data[1] and promise effective Extract-Transform-Load pipelines for integrating statistics. Online Analytical Processing (OLAP) has been proposed as a decision support method for analysing Linked Data describing *data cubes* [12,6].

OLAP engines translate OLAP queries into a target query language of a database storing the multidimensional data. The predominant way in industry is ROLAP since 1) it can be deployed on any of the widely-used relational databases, 2) industry-relevant data such as from accounting and customer relationship management often resemble star schemas [17] and 3) research has focused on optimising ROLAP approaches [15]. Instead of storing the data in a relational database, we have proposed to collect Statistical Linked Data reusing the RDF Data Cube Vocabulary (QB) and to transform OLAP into SPARQL queries [14]. Yet, there is little work on evaluating and optimising analytical

[1] http://wiki.planet-data.eu/web/Datasets

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 290–304, 2013.

queries on RDF data [4,5]. We expect that, similar to general-purpose relational databases, a "one size fits all" [17] triple store will not scale for analytical queries. In this paper, we intend to give an empirical argument in favor of creating a specialised OLAP engine for analytical queries on Statistical Linked Data. Contributions of this paper are centered around four analytical query approaches listed in the following table:

	No Materialisation	Materialisation
Relational data / SQL	RDBMS / ROLAP	ROLAP-M
Graph data / SPARQL	OLAP4LD-SSB/-QB [14]	OLAP4LD-QB-M

- We compare the performance of traditional relational approaches (RDBMS / ROLAP) and of using a triple store and an RDF representation closely resembling the tabular structure (OLAP4LD-SSB). We compare those approaches with our OLAP-to-SPARQL approach [14] reusing a standard vocabulary for describing statistics (OLAP4LD-QB). To use a credible benchmark, we extend our approach for multi-level dimension hierarchies.
- We measure the performance gain of the common ROLAP query optimisation approach to precompute parts of the data cube and to store those "views" in aggregate tables, since they do not fit in memory [15,10] (ROLAP-M). We apply materialisation to our approach, represent views in RDF (OLAP4LD-QB-M) and evaluate their performance gain.

In Section 2, we introduce an OLAP scenario and present our OLAP-to-SPARQL approach. Our optimisation approach of using RDF aggregate views we present in Section 3. In Section 4, we evaluate both OLAP-to-SPARQL approach and RDF aggregate views. In Section 5, we discuss the results, after which we describe related work in Section 6 and conclude in Section 7.

2 OLAP-to-SPARQL Scenario and Approach

We now shortly introduce an OLAP scenario, taken from the Star Schema Benchmark (SSB) [16]. We then use this scenario to explain our extended OLAP-to-SPARQL approach [14] for multi-level hierarchies. In Section 4, we will use the scenario and benchmark for a performance evaluation.

SSB describes a data cube of lineorders. Any lineorder (fact) has a value (member) for six dimensions: the time of ordering (*dates*), the served *customer*, the product *part*, the *supplier*, the ordered *quantity* and granted *discount*. Depending on the member for each dimension, a lineorder exhibits a value for measures having a certain aggregation function with which to compute its value, e.g., *sum profit*, computed by *sum revenue* minus *sum supplycost*.

Dimensions exhibit hierarchies of levels that group members and relate them to higher-level members, e.g., dates can be grouped starting from the lowest *dateLevel* over *yearmonthLevel* to *yearLevel*. Since a week can be spread over two months or years, there is a separate hierarchy where dates can be grouped by *weeknuminyear*, e.g. "199322". Customers and suppliers can be grouped into

cities, nations, and regions and parts into brands, categories and manufacturers. Any hierarchy implicitly has a special-type *ALL* member, which groups all members into one special-type *ALL* level.

SSB provides a workload of 13 queries on the data cube. Each query is originally provided in SQL. For instance, *Q2.1* computes per year the revenues (in USD) for product brands from product category MFGR#12 and of suppliers from AMERICA. Results from this query usually are shown in pivot tables such as the following:

Year\Brand	MFGR#121	MFGR#1210	...	MFGR#129
1992	667,692,830	568,030,008	...	614,832,897
...	...	...	...	...
1998	381,464,693	335,711,347	...	319,373,807

Filter: partCategory = "categoryMFGR#12"
AND supplierRegion = "AMERICA"

More information about the benchmark we provide on our benchmark website [13]. In subsequent sections we present and compare different logical representations of the SSB data cube on Scale 1. First, we describe SSB using an extended OLAP-to-SPARQL approach and sets of multidimensional elements such as *Dimension* and *Cube* [14]. Then, we describe an engine that translates OLAP queries on SSB into SPARQL queries.

Member. All 3,094 dates, 30,280 customer, 201,030 part and 2,280 supplier members from each level are represented as URIs. Any member, e.g., *rdfh:categoryMFGR-35*, links to members on the next lower level via *skos:narrower*, e.g., *rdfh:brand1MFGR-3527*. 51 quantity and 11 discount members we encode as RDF Literal values. Also, we define URIs representing the special-type *ALL* member for each dimension, e.g., customer *rdfh:lo_custkeyAllAll*. Those *ALL* members will later be needed for representing aggregate views.

Level. Every level is represented as a URI, e.g., *rdfh:lo_orderdateDateLevel*, has a *xkos:depth* within its hierarchy and links to a set of members via *skos:member*. The vocabulary *XKOS*[2] allows to represent hierarchy levels.

Hierarchy. Each dimension has one or two (dates) hierarchies. Every hierarchy is represented as a URI, e.g., *rdfh:lo_orderdateCodeList*. Levels with a depth link to the hierarchy via *skos:inScheme*.

Dimension. Every dimension such as dates is represented as an object property, e.g., *rdfh:lo_orderdate* and defines its hierarchy via *qb:codeList*. The simple dimensions *quantity*, *discount* are represented as datatype properties.

Measures. Every measure such as the sum of revenues is represented as a datatype property, e.g., *rdfh:lo_revenue*. The component specification of a measure defines the aggregation function, e.g., SUM, via *qb4o:hasAggregateFunction*, as proposed by Etcheverry and Vaismann [7]. Since there is no recommended way to represent more complex functions, for formulas, we use String Literals using measure URIs as variables.

DataCubeSchema. The data cube schema of the SSB data cube is represented as an instance *rdfh-inst:dsd* of *qb:DataStructureDefinition* and defines the dimensions and measures of the data cube.

[2] `https://github.com/linked-statistics/xkos`

Fact. Every possible lineorder can be represented as a *qb:Observation*. Any observation links for each dimension property to the URI of a member or a Literal value (quantity, discount), and for each measure property to a Literal value. Whereas *base facts* with each dimension on the lowest level are given by the SSB dataset, aggregated facts on higher levels of dimensions of the cube need to be computed.

DataCube. The SSB data cube is identified by the dataset *rdfh-inst:ds*. The dataset defines the schema *rdfh-inst:dsd* and has attached via *qb:dataSet* all base facts.

All queries of SSB can be formalised as OLAP queries on single data cubes with multi-level hierarchies as per Definition 1, e.g., *Q2.1* as follows with abbreviated names: `({yearLevel, ALL, brand1Level, ALL, ALL, ALL}, {categoryLevel = categoryMFGR-12, s_regionLevel = s_regionAMERICA}, {lo_revenue})`. *Q2.1* slices dimensions customer, supplier, discount, quantity, rolls up dates to years and part to product brands, dices for a specific product part category and supplier region and projects the revenues.

Definition 1 (OLAP Query). *Given a data cube* $c = (cs, C) \in DataCube$, *with* $cs = (?x, D, M) \in DataCubeSchema$, $C \in 2^{Fact}$. $D = \{D_1, D_2, \ldots D_d\} \subseteq$ *Dimension is an ordered list of dimensions with a set of levels* $L_i = \{l_1, l_2, \ldots\} \subseteq$ *Level, including the special-type* ALL *level. Each level* l_i *has memberno*(l_i) *members.* $M \subseteq$ *Measure is an ordered list of measures. We define an OLAP query on this cube with OLAP Query* $= \mathcal{SC} \times 2^{Fact}$ *with* $(c, SlicesRollups, Dices, Projections) \in \mathcal{SC}$, *with* $SlicesRollups \subseteq L_1 \times L_2 \times \ldots \times L_d$ *a level for each dimension in the same order (for roll-ups), including the special-type level ALL (for slices), with Dices a set of conditional terms on members of levels (for dice) and with* $Projections \subseteq M$ *a set of selected measures from a data cube (for projection). An OLAP query results in a set of facts from the data cube.*

Given *Member*, *Level*, *Hierarchy*, *Dimension*, *Measure*, *DataCubeSchema*, *Fact*, and *DataCube* as sets of multidimensional data, we define OLAP Engine $\subseteq$ OLAP Query $\times$ Target Query with OLAP Query as per Definition 1, Target Query a query in a target query language such as SQL and SPARQL. The following pseudocode algorithm implements an OLAP engine that transforms an OLAP query into a SPARQL query. The algorithm separately creates the WHERE, SELECT and GROUP BY clause. Note, in this pseudocode we disregard translating multidimensional elements to URI representations and variables, more efficient filters, complex measures and ordering:

```
1  Algorithm 1: OLAP-to-SPARQL
2  Input: OLAP Query (cube, SlicesRollups, Dices, Projections)
3  Output: SPARQL query string
4  begin
5    whereClause = "?obs qb:dataSet " + cube.ds.uri.
6    for level ∈ SlicesRollups do
7      levelHeight = level.getHeight()
8      dimension = level.getHierarchy().getDimension()
9      dimVar = makeUriToParameter(dimension)
10     hashMap.put(dimension, levelHeight)
11     for i = 0 to levelHeight - 1 do
```

```
12        rollUpsPath += dimVar + i + ". " + dimVar + i + " skos:narrower "
13      whereClause += "?obs " + dimension.uri + rollUpsPath + dimVar +
            levelHeight + ". "
14      whereClause += dimVar + levelHeight + " skos:member " + level.uri
15      selectClause, groupByClause += " "+ dimVar + levelHeight
16    for member ∈ Dices.getPositions().get(0).getMembers() do
17      if (dicieslevelHeight > slicesRollupsLevelHeight) do
18        dicesLevelHeight = member.getLevel().getHeight()
19        slicesRollupsLevelHeight = hashMap.get(dimension)
20        dimension = member.getLevel().getHierarchy().getDimension()
21        dimVar = makeUriToParameter(dimension)
22        for i = slicesRollupsLevelHeight to dicesLevelHeight - 1 do
23          dicesPath += dimVar + i + ". " + dimVar + i + " skos:narrower "
24        whereClause += "?obs " + dimension.uri + dicesPath + dimVar +
              dicesLevelHeight + ". "
25    whereClause += " Filter("
26    for position ∈ Dices.getPositions() do
27      for member ∈ position.getMembers() do
28    dimVar =
          makeUriToParameter(member.getLevel().getHierarchy().getDimension())
29    memberFilterAnd += "AND " + dimVar + dicieslevelHeight + " = " + member
30      memberFilterOr += "OR " + memberFilterAnd
31    whereClause += memberFilterOr + ") "
32    for measure ∈ Projections do
33      measVar = makeUriToParameter(measure)
34      selectClause += measure.getAggregationFunction() + "(" + measVar +
            ") "
35      whereClause += " ?obs " + measure.uri + " " + measVar + " ."
36    return selectClause + whereClause + groupByClause
```

We query for all observations of the cube (line 5). Then, for each level, we create a property path starting with ?obs and ending with a dimension variable at the respective level (line 6 to 14). Each level height we store in a map in order to later check whether graph patterns need to be added for dices (10). Then, we add the variables to the select and group by clause (15). Now, we add graph patterns for dices (16 to 31). We assume that the set of conditional terms on members of levels, *Dices*, can be translated into a set of positions with each position describing a possible combination of members for each diced dimension (16). Diced dimensions and levels are fixed for each position; therefore, we only use the first position for adding graph patterns (16). We assume furthermore that measures are only contained in *Projections* but not *SlicesRollups* and *Dices*. We only need to add graph patterns if the height of the diced level is larger than the level mentioned for the same dimension in *SlicesRollups* (17). Then, from the positions in *Dices*, we filter for one (OR, 30) of all possible combinations (AND, 29) of members for each diced dimension. Finally, for each measure in *Projections*, we add a variable with the aggregation function of the measure to the select clause and graph patterns to the where clause (34,35). The following listing shows the relevant parts of the SPARQL query for *Q2.1*:

```
1  SELECT ?rdfh_lo_orderdate ?rdfh_lo_partkey1 sum(?rdfh_lo_revenue) as
       ?lo_revenue
2  WHERE {
3  ?obs qb:dataSet rdfh-inst:ds; rdfh:lo_orderdate ?rdfh_lo_orderdate0.
4  ?rdfh_lo_orderdate1 skos:narrower ?rdfh_lo_orderdate0.
5  ?rdfh_lo_orderdate2 skos:narrower ?rdfh_lo_orderdate1.
6  ?rdfh_lo_orderdate skos:narrower ?rdfh_lo_orderdate2.
7  rdfh:lo_orderdateYearLevel skos:member ?rdfh_lo_orderdate.
8  ?obs rdfh:lo_partkey ?rdfh_lo_partkey0.
9  ?rdfh_lo_partkey1 skos:narrower ?rdfh_lo_partkey0.
10 ?rdfh_lo_partkey skos:narrower ?rdfh_lo_partkey1.
```

```
11  rdfh:lo_partkeyCategoryLevel skos:member ?rdfh_lo_partkey.
12  ?obs rdfh:lo_suppkey ?rdfh_lo_suppkey0.
13  ?rdfh_lo_suppkey1 skos:narrower ?rdfh_lo_suppkey0.
14  ?rdfh_lo_suppkey2 skos:narrower ?rdfh_lo_suppkey1.
15  ?rdfh_lo_suppkey skos:narrower ?rdfh_lo_suppkey2.
16  rdfh:lo_suppkeyRegionLevel skos:member ?rdfh_lo_suppkey.
17  ?obs rdfh:lo_revenue ?rdfh_lo_revenue.
18  FILTER(?rdfh_lo_partkey = rdfh:lo_partkeyCategoryMFGR-12 AND
        ?rdfh_lo_suppkey = rdfh:lo_suppkeyRegionAMERICA ).
19  } GROUP BY ?rdfh_lo_orderdate ?rdfh_lo_partkey1 ORDER BY
        ?rdfh_lo_orderdate ?rdfh_lo_partkey1
```

Here, *Dices*, {`categoryLevel = categoryMFGR-12, s_regionLevel = s_regionAMERICA`}, is translated into one position with one member for part category level and one member for supplier region level. The SPARQL query queries for all facts within the data cube (line 3), adds *skos:narrower* paths up to yearLevel, categoryLevel and s_regionLevel (4 to 16), selects lo_revenue as measure (17), filters for a certain member of part category and of supplier region (18) and groups by yearLevel and brand1Level (19). We assume all RDF data stored in a default graph.

3 RDF Aggregate Views

We now apply a common optimisation technique to the OLAP engine implementing our OLAP-to-SPARQL approach: data cube materialisation, i.e., precomputing of certain facts from the entire data cube and storing them for reuse.

Just as Harinarayan et al. [11], we assume that the cost of answering an OLAP query is proportional to the number of facts that need to be scanned, e.g., for validating a filter or calculating an aggregation. So far, any OLAP query to the SSB data cube needs to scan the 6,000,000 base facts. Intuitively, materialisation pre-computes facts with dimensions on higher levels, so that views contain fewer and already aggregated facts to be examined for filtering or further aggregation.

Definition 2 (Aggregate View). *We define a view in a data cube c as an OLAP query per Definition 1 $(c, SlicesRollups, Dices, Projections)$ with $SlicesRollups \subseteq L_1 \times L_2 \times \ldots \times L_d$, Dices the empty set, and Projections a set of measures. Thus, any fact within the view gives a value for each of its measures for a certain combination of level members. A view may be sparse and not contain facts for each possible combination of members. The maximum number of facts within a view is given by $\prod memberno(l_i), l_i \in L_i$. The number of views in the data cube is given by $\prod |L_i|$. The facts from an aggregate view can be generated by executing the OLAP query using an OLAP engine.*

The SSB data cube contains $6*5*5*5*2*2 = 3,000$ views with dates having six levels since the two hierarchies of dates contain the same lowest and *ALL* level. The advantage of aggregate views as per Definition 2 is that the entire set of views of a data cube with multi-level hierarchies can be represented as a *data cube lattice* [11], see Figure 1 for an illustration of the lattice of the SSB cube. Any view is represented by the level of each dimension, omitting any *ALL* levels. The single view on the lowest level corresponds to the OLAP query that contains all

base facts, i.e., the view returns all non-aggregated facts from the SSB dataset. The view contains maximum $2,555 * 30,000 * 200,000 * 2,000 * 51 * 11 \geq 1.7 * 10^{19}$ facts, however, SSB provides a sparse data cube with $6,000,000$ facts. From this lowest view one can reach higher views via *roll-up* operations on dimensions, e.g., the next higher view on the right side rolls up to the *ALL* level of quantity. The single view on the highest level in Figure 1 corresponds to the OLAP query that returns one single fact grouping by the special-type level *ALL* with the single member *ALL* for each dimension.

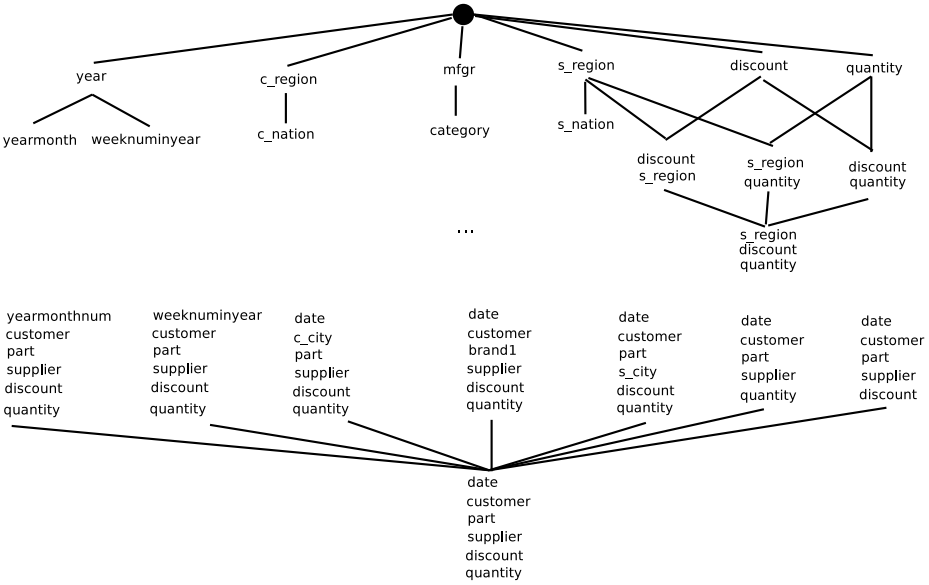


Fig. 1. Illustration of data cube lattice of SSB

The higher the view on a path in the lattice, the fewer facts it contains, since higher levels group lower-level members into groups of fewer members. For distributive aggregation functions such as *SUM*, and algebraic formulas such as `SUM(rdfh:lo_revenue - rdfh:lo_supplycost)`, we do not run into summarisability problems [9] and a view can be computed from any view on a lower level that can be reached via a *roll-up* path; for instance, the view grouping by quantity on the right upper corner can be computed from the view grouping discount and quantity, s_region and quantity, their collective child view grouping by s_region, discount, and quantity as well as from any other reachable lower level view not displayed. The holistic aggregation function `SUM(rdfh:lo_extendedprice * rdfh:lo_discount)` is not further aggregated from views, thus, *Q1.1* to *Q1.3* return correct results.

Summing up for each dimension the number of members on the lowest level, the numbers of members on each level per hierarchy, and the special-type member *ALL*, we can calculate the maximum number of facts in the entire data

cube: $3095 * 30281 * 201031 * 2281 * 52 * 12 > 2.6 * 10^{19}$. As materialising the entire data cube would 1) take too much time and 2) require too much hard disk space, we are concerned with deciding which views to materialise. We define for a given OLAP query (c, *SlicesRollups*, *Dices*, *Projections*) as per Definition 1 a single *closest* view in the lattice from which we can create the results by only scanning the facts in the view [11]: We create a view (c, *SlicesRollups*′, *Dices*′, *Projections*′) on the same cube that contains in *SlicesRollups*′ for each dimension the lowest level mentioned in *SlicesRollups* and *Dices*, contains an empty set for *Dices*′ and $M' = M$. The following term describes the *closest* view for *Q2.1*, the other views are translated, accordingly: (`{yearLevel, ALL, brand1Level, s_regionLevel, ALL, ALL}`, $\emptyset$, `{lo_revenue}`). The view contains maximum 35,000 facts and as such is considerably smaller than the SSB dataset with 6,000,000 facts. Note, *Q2.2* and *Q2.3* can use the same view as *Q2.1* and *Q3.3* can use the same view as *Q3.2*, resulting in less time and less space for creating the views. Though some views can contain as many facts as there are base facts in the data cube, they often do not due to sparsity, e.g., *Q4.3* with 4,178,699 facts. For views may still be large, in ROLAP, views are stored in *aggregate tables*. Similarly, we represent views as *RDF aggregate views* reusing QB and store the triples together with the other multidimensional data in the same triple store. See Figure 2 for an illustration of this approach for *Q2.1*.

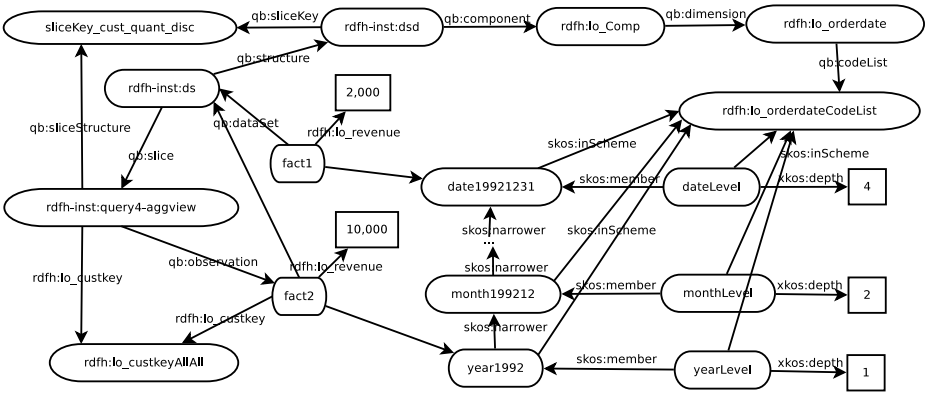


Fig. 2. Modelling RDF aggregate view rolling up to year level using QB

For dimensions on the *ALL* level, aggregate views only contain facts that fix those dimensions to the *ALL* member, e.g., *Q2.1* fixes customer. Therefore, we can represent *RDF aggregate views* as instances of *qb:Slice*, e.g., *rdfh-inst:query4-aggview*. *qb:SliceKeys* describe the structure of a slice, i.e., the sliced dimensions (not shown in figure). Slices explicitly state to what member a sliced dimension is fixed, e.g., *rdfh:lo_custkey* to *ALL*. In addition to base facts, e.g., *fact1* with member *date19921231* in the date level, facts are created that aggregate on the specific levels of the view. For instance, the view of Q2.1 contains via

qb:observation a *fact2* that rolls-up to the higher-level-member *year1992* in the year level of the dates hierarchy. The higher-level-member is connected to the lower-level-member in a *skos:narrower* path. Also, *fact2* rolls-up to the special-type *ALL* member of customer. The datatype property *xkos:depth* states for each level the depth of a level starting with 0 from the (implicit) *ALL* level. The following listing shows the relevant parts of a SPARQL INSERT query on the SSB data that populates the RDF aggregate view for *Q2.1*:

```
1  INSERT {
2  rdfh-inst:query4-aggview qb:observation _:obs.
3  _:obs rdfh:lo_orderdate ?d_year; rdfh:lo_custkey rdfh:lo_custkeyAllAll;
       rdfh:lo_partkey ?p_brand1; rdfh:lo_suppkey ?s_region;
       rdfh:lo_quantity rdfh:lo_quantityAllAll; rdfh:lo_discount
       rdfh:lo_discountAllAll; rdfh:lo_revenue ?lo_revenue.}
4  WHERE {{
5  SELECT ?d_year ?p_brand1 ?s_region sum(?rdfh_lo_revenue) as ?lo_revenue
       WHERE {
6  ?obs qb:dataSet rdfh-inst:ds.
7  ?obs rdfh:lo_orderdate ?d_date.
8  ?d_yearmonthnum skos:narrower ?d_date.
9  ?d_yearmonth skos:narrower ?d_yearmonthnum.
10 ?d_year skos:narrower ?d_yearmonth.
11 rdfh:lo_orderdateYearLevel skos:member ?d_year.
12 ?obs rdfh:lo_partkey ?p_part.
13 ?p_brand1 skos:narrower ?p_part.
14 rdfh:lo_partkeyBrand1Level skos:member ?p_brand1.
15 ?obs rdfh:lo_suppkey ?s_supplier.
16 ?s_city skos:narrower ?s_supplier.
17 ?s_nation skos:narrower ?s_city.
18 ?s_region skos:narrower ?s_nation.
19 rdfh:lo_suppkeyRegionLevel skos:member ?s_region.
20 ?obs rdfh:lo_revenue ?rdfh_lo_revenue.
21 } GROUP BY ?d_year ?p_brand1 ?s_region
22 }}
```

Here, we first create a SELECT query using our OLAP-to-SPARQL algorithm on the OLAP query (line 5), then this SELECT query is made a subquery of an INSERT query. Observations roll-up to members of specific levels and fix sliced dimensions (3). Resulting triples are stored in the default graph. We can now easily adapt our OLAP-to-SPARQL algorithm to use for an OLAP query the RDF aggregate views instead of the base facts from the SSB dataset. The following listing shows the SPARQL query for *Q2.1*.

```
1  SELECT ?d_year ?p_brand1 sum(?rdfh_lo_revenue) as ?lo_revenue
2  WHERE {
3  rdfh-inst:ds qb:slice ?slice.
4  ?slice qb:observation ?obs;
5      rdfh:lo_custkey rdfh:lo_custkeyAllAll;
6      rdfh:lo_quantity rdfh:lo_quantityAllAll;
7      rdfh:lo_discount rdfh:lo_discountAllAll.
8  ?obs rdfh:lo_orderdate ?d_year.
9  rdfh:lo_orderdateYearLevel skos:member ?d_year.
10 ?obs rdfh:lo_partkey ?p_brand1.
11 ?p_category skos:narrower ?p_brand1.
12 rdfh:lo_partkeyCategoryLevel skos:member ?p_category.
13 ?obs rdfh:lo_suppkey ?s_region.
14 rdfh:lo_suppkeyRegionLevel skos:member ?s_region.
15 ?obs rdfh:lo_revenue ?rdfh_lo_revenue.
16 FILTER(?p_category = rdfh:lo_partkeyCategoryMFGR-12 AND ?s_region =
       rdfh:lo_suppkeyRegionAMERICA ).
17 } GROUP BY ?d_year ?p_brand1 ORDER BY ?d_year ?p_brand1
```

Here, we query for observations from slices of *rdfh-inst:ds* that fix customer, quantity and discount to the *ALL* member, as indicated in *SlicesRollups* of *Q2.1* (lines 3 to 7). In comparison to the OLAP SPARQL query of *Q2.1* without views, we have a reduced set of triple patterns for rolled-up dimensions (*skos:narrower* paths) (8 to 14). To distinguish between observations from views slicing the same dimensions but rolling-up to different levels, we require for each member the correct level (9, 12, 14). And we add filters on diced dimensions (16).

4 Evaluation

We now give an overview of tested approaches and the reasons for their selection, then explain the design of the tests. See the benchmark website for this paper [13] for more background information about the tests:

Name	Data Format	Metadata	Query Language	Engine/Database	Pre-processing (s)	Rows / Triples
RDBMS	Relational	-	SQL	MySQL	22	6,234,555
ROLAP-M	Relational	XML	SQL	MySQL, Mondrian	4,507	14,975,472
OLAP4LD-SSB	Graph-based	-	SPARQL	Open Virtuoso	5,352	108,021,078
OLAP4LD-QB	Graph-based	RDF/QB	SPARQL	Open Virtuoso	5,744	116,832,479
OLAP4LD-QB-M	Graph-based	RDF/QB	SPARQL	Open Virtuoso	26,032	190,060,632

RDBMS and ROLAP-M represent the traditional approaches with a widely-used Open-Source relational database (MySQL 5.1 v5.1.63) and SQL. ROLAP-M uses aggregate tables for optimising queries. The other tests represent graph-based approaches with a widely-used Open-Source triple store (Open Virtuoso v06.01.3127) and SPARQL 1.1 for aggregate and sub-queries. Whereas OLAP4-LD-SSB represents SSB data without reusing a vocabulary, OLAP4LD-QB reuses QB which allows us to materialise parts of the data cube as RDF aggregate views in OLAP4LD-QB-M.

We use the Star Schema Benchmark [16], since SSB 1) refines the decision support benchmark TPC-H by deriving a pure star schema from the schema layout in order to evaluate analytical query engines [1], and 2) can be regarded as a realistic data source since statistics published as Linked Data are typically highly structured [4,3]. We run each approach on a Debian Linux 6.0.6, 2x Intel(R) Xeon(R) CPU E5-2670 @ 2.60GHz with 16 cores, 128GB RAM and 900GB RAID10 on 15k SAS disks for local data storage. We assume unlimited amount of space but configure the databases to only use an amount of memory clearly below 100% of the space the data files surmount to (400M for relational approaches, $<$ 650M for graph-based approaches), since storing all multidimensional data in main memory is often too costly. For each approach we 1) translate the SSB data cube at Scale 1 with 6,000,000 lineorders into the respective data format for storage in the database, 2) simulate an OLAP engine translating the SSB OLAP queries into the respective query language of the database 3) before each test, shut-down all other applications not needed and run the test once to populate the disk cache (warm-up) and 4) document the elapsed query time of each query in turn. Note, we do not consider data refreshes. For running the SSB benchmark and collecting the data about elapsed query times, we used the Business Intelligence Benchmark (BIBM)[3]. BIBM also ensured identical results

[3] `http://sourceforge.net/projects/bibm/`

for the approaches through qualification files (provided at website). We now describe for each approach how we stored SSB data in the database and translated SSB OLAP queries to the database query language.

RDBMS. We created a schema file for dimension and fact tables and populated the database with an SSB data generator. We setup column data types as recommended by SSB and indexes for each foreign key in the fact table, primary keys for the fact table comprising the dimension keys and primary keys for dimension tables in a standard Star Schema fashion. Loading of 6,234,555 rows of data took 22s. The SQL queries of SSB could be reused with minor MySQL-syntax-specific modifications. We switched off query cache so that MySQL after a warm-up would not read all queries from cache. Note, we have compared those SQL queries with SQL queries created by the widely-used Open-Source ROLAP engine Mondrian (v3.4.1). Mondrian stores data cube metadata in XML and would for example deliberately query for more data than requested by the query and cache the results for later use; however, SSB minimises overlap between queries, e.g., *Q1.1* uses discounts between 1 and 3, *Q2.1* between 4 and 6. Since the performance gain of using Mondrian-created SQL queries instead of the original SSB SQL queries showed small, we only include a Mondrian test in the benchmark website (ROLAP).

ROLAP-M. We created aggregate tables without indices and keys for the *closest* view to each query using SQL INSERT queries on the original tables from *RDBMS*. Time included 22s for preparing *RDBMS* with 6,234,555 rows and 4,485s for creating the aggregate tables with another 8,740,917 rows. For each OLAP query we created an SQL query using the *closest* aggregate table. Similarly, Mondrian would choose the aggregate table with the smallest number of rows and create an SQL query with comparable performance.

OLAP4LD-SSB. With BIBM we translated the SSB tabular data into RDF/TTL files using a vocabulary that strongly resembles the SSB tabular structure: A lineorder row is represented as a URI which links for each dimension via an object property, e.g., *rdfh:lo_orderdate*, to a URI representing a row from the respective dimension table, e.g., *rdfh:lo_orderdate19931201*. From this URI, datatype properties link to Literal values for members, e.g., month "199312". Quantity and discount are directly given using datatype properties from a lineorder. Each measure is attached to the lineorder URI using a datatype property. Translation took 48s, bulk loading of $108,021,078$ triples 5,304s. For each SSB OLAP query, we tried to build the most efficient SPARQL-pendant to the original SSB SQL queries, e.g., reducing the number of joins.

OLAP4LD-QB. We created RDF metadata for the SSB data cube using our extended OLAP-to-SPARQL approach and via a small script added links from each lineorder of *OLAP4LD-SSB* to *rdfh-inst:ds*. Using SPARQL INSERT queries for each dimension, we grouped dimension members into levels of hierarchies, and added them to the triple store. Creating the *OLAP4LD-SSB* data and adding links took 48s and 38s, the INSERT queries 14s; compressing and bulk loading

of 116,832,479 triples took 60s and 5,584s. Simulating our OLAP-to-SPARQL algorithm, we manually translated the SSB queries to SPARQL.

OLAP4LD-QB-M. For each SSB query, we created a *closest* RDF aggregate view using a SPARQL INSERT query. Setting up *OLAP4LD-QB* took 5,744s, the SPARQL INSERT queries 20,288s for another 73,228,153 triples. Also, we created SPARQL queries that use the *closest* views.

5 Presentation and Discussion of Results

In this section, we evaluate 1) the scalability of the OLAP-to-SPARQL approach and 2) the performance gain of RDF aggregate views. Table 1 lists performance-relevant SSB query features. *Filter factor* measures the ratio of fact instances that are filtered and aggregated. Filter factors are computed by multiplying the filter factors of each dice, e.g., for *Q2.1* the filter factor is 1/25 for part times 1/5 for supplier. *View factor* measures the ratio of fact instances that are contained in a view in relation to the 6M base facts. For example, from the filter factor and view factor, we see that query flight 4 (Q4) iteratively drills-down to more granular levels (up to 4,178,699 facts) but filters for fewer, more specific lineorders. With *RDBMS joins* we describe the number of joins between tables in the SQL representation of a query. Note, ROLAP-M does not need joins. With *SSB, QB and QB-M joins* we state the number of triple pattern joins, pairs of triple patterns mentioning the same variable. Table 2 lists the elapsed query times (s) which we now discuss.

Table 1. Overview of SSB queries and their performance-relevant features

Feature	Q1.1	Q1.2	Q1.3	Q2.1	Q2.2	Q2.3	Q3.1	Q3.2	Q3.3	Q3.4	Q4.1	Q4.2	Q4.3
Filter factor	.019	.00065	.000075	.008	.0016	.0002	.034	.0014	.000055	.00000076	.016	.0046	.000091
View factor	.00064	.0073	.0032	.0058	.0058	.0058	.0007	.0728	.0728	.5522	.0007	.0036	.6964
RDBMS joins	1	1	1	3	3	3	3	3	3	3	4	4	4
SSB joins	5	5	6	8	7	7	9	9	7	8	10	12	12
QB joins	8	6	6	15	13	14	16	16	16	12	22	22	22
QB-M joins	9	9	9	12	11	11	13	13	11	12	13	14	14

Table 2. Evaluation results with single and total elapsed query time (s)

Name	Q1.1	Q1.2	Q1.3	Q2.1	Q2.2	Q2.3	Q3.1	Q3.2	Q3.3	Q3.4	Q4.1	Q4.2	Q4.3	Total
RDBMS	1.6	1.1	1.1	16.1	15.7	15.4	10.4	7.8	7.6	3.1	11.0	5.3	5.0	101
ROLAP-M	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.6	0.1	0.1	0.8	2
OLAP4LD-SSB	22.5	0.8	0.2	16.1	0.9	0.2	28.5	2.1	1.0	0.4	N/A	36.8	9.6	119
OLAP4LD-QB	46.1	1.3	0.2	55.0	49.4	31.1	145.7	12.5	1.8	87.2	175.3	544.5	24.9	1175
OLAP4LD-QB-M	19.9	10.2	10.2	366.3	356.7	356.3	468.5	467.6	4.6	4.6	0.1	0.4	55.4	2121

ROLAP-M overall is 50 times faster than *RDBMS* for not requiring any joins and a reduced number of facts to scan for aggregation. Whereas RDBMS has to first scan 6M rows and then to aggregate, ROLAP-M only has to scan the view and to aggregate from there. Affirmatively, the views of Q3.4 and Q4.3 with very low selectivity show smaller benefits. However, preparing ROLAP-M takes 200 times longer than RDBMS.

Comparing *OLAP4LD-SSB* and RDBMS, we see that the graph database is as fast as the relational alternative for some of the queries (e.g., Q1.2, Q2.1), slower for other queries (e.g., Q1.1, Q3.1, Q4.2), and even faster for others (Q2.2, Q3.3). Over all queries, OLAP4LD-SSB is only slightly worse, however, Q4.1 for no known reason does not successfully complete. Differences can be explained by the number of joins; for instance, whereas RDBMS requires for Q3.1 and Q4.2 three and four joins, OLAP4LD-SSB requires nine and twelve joins, respectively. If the number of joins is less divergent, differences can be explained by the filter factor and the fact that after filtering facts still need to be aggregated. In general, the smaller the filter factor, the better the graph database seems in comparison to the relational database, for instance Q2.2, Q2.3 and Q3.3. For low-selective queries, the graph database performs worse, e.g., Q1.1, Q3.1, Q4.2. This aligns with our expectations that a graph database is more optimised for high-selectivity metadata queries without aggregations. OLAP4LD-SSB requires 243 times as much time for loading.

OLAP4LD-QB reusing QB requires up to twice as many joins than OLAP4LD-SSB (Q2.3), since hierarchies are explicitly represented through *skos:narrower* paths from higher-level to lower-level members, and consequently is 10 times slower. Both approaches require similar pre-processing time. Yet, only OLAP4LD-QB can represent hierarchies and be optimised using RDF aggregate views.

Although *OLAP4LD-QB-M* overall leads to 1.8 times slower queries and performs considerably worse for query flights 2 and 3 (Q2/3), it succeeds in optimising query flight 4 (Q4). Similar to ROLAP-M, the performance gain RDF aggregate views can be explained by a reduced number of joins, e.g., for Q4.1, Q4.2. However, for most queries OLAP4LD-QB-M performs worse, since RDF aggregate views – different from ROLAP-M with separately created aggregate tables – are stored in the same graph and do not reduce the number of facts scanned for a query. Thus, whereas OLAP4LD-QB needs to scan for 6M facts, OLAP4LD-QB-M also needs to scan over facts from the aggregate views, in total 14.98M facts. Queries need to compensate for the increased effort in scanning by the reduced number of joins, in which Q2.1 to Q3.2 apparently do not succeed.

6 Related Work

In this section, we describe related work on 1) evaluating analytical query execution on RDF data, 2) representing multidimensional data as RDF and 3) materialising aggregate views on RDF data.

In our OLAP-to-SPARQL approach we have chosen RDF reusing QB as a logical representation, SPARQL as a query language for computation, and materialised *closest* views from the data cube lattice that promise the largest performance gain. We compare analytical queries on RDF with common alternatives in a realistic scenario, according to Erling [4] a prerequisite for successful RDF use cases and targeted optimisations [5]. Most notably, the *Berlin SPARQL*

Benchmark BI Use Case allows quantifying analytical query capabilities of RDF stores, but, so far, no work compares the RDF performance with the industry-standard of relational star schemas.

Recent work discusses approaches to represent multidimensional data in RDF [12,7,6], however, no approach deals with the computation and selection of data cube slices and dices in RDF, in particular, considering the special-type *ALL* members and levels for uniquely identifying all possible facts of a data cube.

Several authors discuss views over RDF data. Castillo and Leser [2] have presented work on automatically creating views from a workload of SPARQL queries. In the evaluation, they use a dataset with 10M triples and disregard queries that exhibit a high selectivity. Also, Goasdoué et al. [8] have discussed the creation and selection of RDF views. Their evaluation is done on a 35M triple dataset. In contrast to these approaches, our approach considers more complex views based on aggregation functions and hierarchies, materialises views as RDF reusing QB in a triple store and evaluates the applicability for high- and low-selectivity queries on a $>$ 100M triple dataset.

7 Conclusion

We now give an empirical argument in favor of creating a specialised OLAP engine for analytical queries on RDF. Although a triple store has shown almost as fast as a relational database, OLAP scenarios such as from the Star Schema Benchmark used in our evaluation require results in seconds rather than minutes. Materialised views with aggregate tables overall reach 50 times faster queries. Queries by our OLAP-to-SPARQL approach on data reusing the RDF Data Cube Vocabulary (QB) overall are 10 times slower than queries on data without reusing QB, for a large number of joins are required for rolling-up on dimensions; yet, only QB metadata allows to explicitly represent dimension hierarchies and to materialise parts of the data cube. RDF aggregate views show the capability to optimise query execution, yet, overall still take six times longer for preprocessing and not nearly reach the performance gain of aggregate tables in ROLAP. The reason seems that the reduced number of joins for queries on RDF aggregate views often cannot compensate for the increased number of facts that are stored in the triple store and need to be scanned for query execution. We conclude that the query optimisation problem intensifies in many OLAP scenarios on Statistical Linked Data and that OLAP-to-SPARQL engines for selection and management of RDF aggregate views are needed.

Acknowledgements. This work is supported by the Deutsche Forschungsgemeinschaft (DFG) under the SFB/TRR 125 - Cognition-Guided Surgery and under the Software-Campus project. We thank Günter Ladwig, Andreas Wagner and the anonymous reviewers for helpful support and feedback.

References

1. Bog, A., Plattner, H., Zeier, A.: A mixed transaction processing and operational reporting benchmark. Information Systems Frontiers 13, 321–335 (2011)
2. Castillo, R., Leser, U.: Selecting Materialized Views for RDF Data. In: Daniel, F., Facca, F.M. (eds.) ICWE 2010 Workshops. LNCS, vol. 6385, pp. 126–137. Springer, Heidelberg (2010)
3. Duan, S., Kementsietsidis, A., Srinivas, K., Udrea, O.: Apples and Oranges: a Comparison of RDF Benchmarks and Real RDF Datasets. In: Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data (2011)
4. Erling, O.: Directions and Challenges for Semdata. In: Proceedings of Workshop on Semantic Data Management (SemData@VLDB 2010) (2010)
5. Erling, O.: Virtuoso, a Hybrid RDBMS/Graph Column Store. IEEE Data Eng. Bull. 35, 3–8 (2012)
6. Etcheverry, L., Vaisman, A.A.: Enhancing OLAP Analysis with Web Cubes. In: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 469–483. Springer, Heidelberg (2012)
7. Etcheverry, L., Vaisman, A.A.: QB4OLAP: A Vocabulary for OLAP Cubes on the Semantic Web. In: Proceedings of the Third International Workshop on Consuming Linked Data (2012)
8. Goasdoué, F., Karanasos, K., Leblay, J., Manolescu, I.: View Selection in Semantic Web Databases. PVLDB 5, 97–108 (2011)
9. Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., Venkatrao, M., Pellow, F., Pirahesh, H.: Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. Data Mining and Knowledge Discovery 1, 29–53 (1997)
10. Gupta, A., Mumick, I.S.: Maintenance of Materialized Views: Problems, Techniques, and Applications. In: Materialized Views. MIT Press (1999)
11. Harinarayan, V., Rajaraman, A.: Implementing Data Cubes Efficiently. In: Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data (1996)
12. Kämpgen, B., Harth, A.: Transforming Statistical Linked Data for Use in OLAP Systems. In: Proceedings of the 7th International Conference on Semantic Systems (2011)
13. Kämpgen, B., Harth, A.: Benchmark Document for No Size Fits All – Running the Star Schema Benchmark with SPARQL and RDF Aggregate Views (2012), `http://people.aifb.kit.edu/bka/ssb-benchmark/`
14. Kämpgen, B., O'Riain, S., Harth, A.: Interacting with Statistical Linked Data via OLAP Operations. In: Proceedings of Workshop on Interacting with Linked Data (2012)
15. Morfonios, K., Konakas, S., Ioannidis, Y., Kotsis, N.: ROLAP Implementations of the Data Cube. ACM Computing Surveys 39 (2007)
16. O'Neil, P., O'Neil, E., Chen, X.: Star Schema Benchmark - Revision 3. Tech. rep., UMass/Boston (2009), `http://www.cs.umb.edu/~poneil/StarSchemaB.pdf`
17. Stonebraker, M., Bear, C., Cetintemel, U., Cherniack, M., Ge, T., Hachem, N., Harizopoulos, S., Lifter, J., Rogers, J., Zdonik, S.: One Size Fits All? – Part 2: Benchmarking Results. In: Proceedings of the Third International Conference on Innovative Data Systems Research (2007)

Seven Commandments for Benchmarking Semantic Flow Processing Systems*

Thomas Scharrenbach[1], Jacopo Urbani[2], Alessandro Margara[2], Emanuele Della Valle[3], and Abraham Bernstein[1]

[1] University of Zurich, Switzerland
{scharrenbach,bernstein}@ifi.uzh.ch
[2] Vrije Universiteit Amsterdam, The Netherlands
jacopo@cs.vu.nl, a.margara@vu.nl
[3] Politecnico di Milano, Italy
emanuele.dellavalle@polimi.it

Abstract. Over the last few years, the processing of dynamic data has gained increasing attention in the Semantic Web community. This led to the development of several stream reasoning systems that enable on-the-fly processing of semantically annotated data that changes over time. Due to their streaming nature, analyzing such systems is extremely difficult. Currently, their evaluation is conducted under heterogeneous scenarios, hampering their comparison and an understanding of their benefits and limitations. In this paper, we strive for a better understanding of the key challenges that these systems must face and define a generic methodology to evaluate their performance. Specifically, we identify three *Key Performance Indicators* and seven *commandments* that specify how to design the stress tests for system evaluation.

1 Introduction

The processing of dynamic data is becoming an important research area in the Semantic Web community, and this is fueled by an increasing number of use cases where input data cannot be considered as static, but rather as a "flow" that continuously changes as computation takes place [16]. Examples range from information produced by on-line newspapers, blogs, and social networks to data generated by sensor networks for environmental monitoring, weather forecast, or traffic analysis in big cities, as well as stock prices for financial analysis.

This led to the definition of a number of stream reasoning systems [10, 11, 20] that combine the on-the-fly processing capabilities of Information Flow Processing (IFP) systems [15] with the use of semantically annotated data, in the form

* We would like to express our thanks to Srdjan Komazec for his valuable comments and discussions. The research leading to these results has received funding from the European Union Seventh Framework Programme FP7/2007-2011 under grant agreement no 296126, from the Dutch national program COMMIT, and from the Dept. of the Navy under Grant NICOP N62909-11-1-7065 issued by Office of Naval Research Global.

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 305–319, 2013.

of RDF triples. To avoid bias in terminology and in continuity with the definition of IFP systems, we collectively denote such systems *Semantic Flow Processing* (SFP) systems. Since in SFP scenarios data changes over time, query answers need to be updated to reflect such changes. This fact turns the entire query process upside-down: whilst "traditional" query engines operate on fixed data and changing queries, the SFP scenario evaluates fixed queries on changing data.

Empirical evaluation of systems is a significant challenge in computer science [23, 24]. Due to their complexity and heterogeneity this is especially true for SFP systems. Despite the number of SFP systems presented in literature, their evaluation is still conducted in incomparable and limited scenarios, without addressing a proper definition of the key performance indicators. This complicates (or even prevents) any meta-analysis comparing the different systems to understand their distinctive aspects, benefits, and limitations.

In this paper, we study the problem of *benchmarking SFP systems* with the purpose of better understanding the key challenges that these system must face and defining a generic methodology to evaluate their performance. We base our study upon a recent survey of IFP systems [15], the commandments for benchmarking databases [18], and our analysis of available benchmarks for testing SFP systems [21, 26]. Our study first *identifies the challenges that SFP systems must face*. Starting from these challenges, we discern the *key performance indicators* (KPIs) of SFP systems and introduce *seven commandments* on how to evaluate the performance of SFP systems according to these KPIs.

This work makes no effort towards defining yet another benchmark for evaluating the performance of SFP systems. On the contrary, *we identify as the main contribution a systematic guideline for assessing the KPIs of SFP systems*. Not only this is useful for a systematic evaluation of a concrete benchmarking framework at hand. By identifying the main KPIs for the abstract SFP scenario, our work can be also used for understanding the requirements of concrete applications as well as guide the design and configuration of an SFP system capable of satisfying them.

The paper is structured as follows: Section 2 provides background information on IFP and SFP systems, as well as on frameworks and methodologies for evaluating their performance. Section 3 investigates the main properties of SFP systems, which we use in Section 4 to present the main challenges in the domain. In Section 5 we discuss the most appropriate KPIs and stress tests for the evaluation. Finally, Section 6 summarizes our findings and concludes the paper.

2 Related Work

This section presents related work in the area of IFP and SFP systems, and in the area of benchmarks for flow-processing systems.

Flow Processing Systems. The last years have seen the development of a large number of IFP systems. These process continuous flows of information based on a set of pre-deployed rules or queries to produce and deliver timely

responses to interested parties. Despite their common goals, existing systems greatly differ in the language they use to define queries and on the adopted processing mechanisms [15]. Based on these aspects, we can roughly classify them into two main classes: Data Stream Managements Systems (DSMSs) [7] and Complex Event Processing (CEP) systems [22]. Note that there exist hybrid systems that combine features of DSMS and CEP.

DSMSs have been developed by the database community and exploit a processing model that is similar to that of traditional DBMSs. More in particular, they adopt *window* operators to isolate the portions of streams that are relevant for processing and logically operate on these portions using relational algebra operators. This processing model is described in [5] and, despite some differences, it represents the common ground of all DSMSs [1, 8, 13].

CEP systems [2, 12, 14] take a different approach. While DSMSs use relational operators to *transform* input streams, CEP rules *define* higher level information (in the form of composite events) from patterns of primitive events observed from the external environment.

SFP systems extend the IFP domain by considering semantically annotated data, based on the RDF data model. They extend IFP systems by inference mechanisms that reach from simple RDFS inference to supporting the OWL2 profiles.[1] Most SFP systems [10, 11, 20] use the query model of DSMSs, enriching it with the possibility to perform reasoning over streaming data. Only few approaches [4] take a different direction and combine RDF data with the processing model of CEP systems.

Stream Benchmarking. In the following, we first present the Linear Road Benchmark and the Fast Flower Delivery use case—the accepted means to compare DSMSs and CEP systems—and then SR-Bench and the SLD-Bench – the two existing proposals for benchmarking SFP systems.

Linear Road (LR) This benchmark [6] was proposed by groups at MIT, Brandeis University, Brown University, and Stanford University to compare the performance characteristics of different DSMSs and of alternative (e.g., Relational Database) systems. LR simulates a variable tolling system for highways. Toll charges are determined dynamically considering traffic congestion and accident proximity. The benchmark does not specify a solution but describes the requirements of the tolling system both functionally (e.g., how to determine the level of traffic congestion or to detect accidents) and non-functionally (e.g., the vehicle must receive toll notifications at most five seconds after moving from one road segment to the following one). LR comes with a simulator, an environment that validates the results of the system being benchmarked, and a set of software sensors to measure response time and supported query load.

Fast Flower Delivery (FFD) evolved from a running example [17] to a must-to-implement showcase for commercial CEPs. It proposes a logistic scenario, where independent van drivers are asked to deliver flowers from the city's flower stores to their destinations. The use case is divided into five phases: *1*) a bid phase,

[1] `http://www.w3.org/TR/owl2-profiles/`

when a store offers highly rated drivers nearby to deliver flowers to a destination within a given time; *2*) an assignment phase, when the system (manually or automatically) decides which driver shall deliver the flowers; *3*) a deliver process, when the system monitors the delivery process; *4*) a ranking evaluation, when the system increases or decreases each driver's ranking based on the ability to deliver flowers on time; and *5*) an activity monitoring, when the system evaluates drivers ranking over time.

SR-Bench (SR) is defined on measurements of sensors and a fixed (i.e. non-parameterized) set of queries [26]; some of which require RDFS reasoning capabilities. Each graph points to a) the sensor, b) the timestamp of the observation, and c) the actual observation. Each of the above refers to a complex object, where the sensor, the timestamp and the observation follow a pre-defined fixed schema. Observations are considered as flow-data whereas the schema and the background knowledge are considered fixed. SR-Bench comprises 17 queries that can be divided in sub-categories to test different kinds of use-cases: 1) query only flow-data (Q1-Q7), 2) query both flow and background data (Q8-Q11), and 3) additionally query the GeoNames and DBpedia datasets (Q12-Q17). Some of these queries require inference capabilities (Q3, Q15-17).

SLD-Bench [21] is defined on three synthetically generated social streams (i.e., a stream of GPS position of the social media users, a stream of micro-posts, and a stream of uploaded images), a synthetically generated social graph, and a fixed (i.e., non-parameterized) set of queries. Emphasis is on processing social streams against a large dataset of static data. SLD-Bench includes 12 queries: some challenge only flow data (Q1, Q4, Q8, Q10-Q11), others joining flow and static data (Q2, Q3, Q5-Q7, Q9), none requiring inference capabilities.

3 Properties of SFP Systems

Following the terminology for IFP systems [15] Figure 1 shows the abstract architecture of an SFP system. It receives flows (or streams) of *information items* from external sources and processes them on-the-fly to produce results for a set of connected *sinks*. All existing SFP systems use RDF triples for representing information items. Processing is governed by a set of *rules* or *queries* deployed into the system. It is performed by one or more interconnected *processors* and may consider (semi)static *background data* in addition to the information flowing

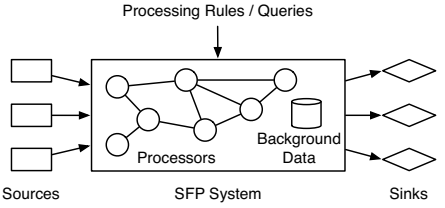


Fig. 1. Abstract Architecture of an SFP System

from sources. Processors cooperate to generate final results for sinks by producing and sharing *partial results* (e.g. variable bindings that are not yet complete).

With reference to this architecture, we identified *seven* main properties of SFP system. Note that these properties are not unique, but rather those useful to determine the list of challenges for an SFP system. A complete classification of SFP systems is however beyond the scope of this paper.

[P1] Support of Background Data. It defines the feature of considering background data during processing. An SFP system can either support or ignore background data; assume that such data is fixed and available ex-ante; or allow (infrequent) changes to this data.

[P2] Inference Support. The usage of semantically annotated data allows the SFP system to infer implicit information. This process is broadly referred as *inference* or *reasoning*. The ability of performing inference is feature unique to SFP system and not available in IFP systems. We make, however, no assumption about the expressive power of the inference mechanism.

[P3] Quality of Service (QoS). The QoS property identifies whether an SFP system performs best effort processing or guarantees some specific levels of performance. The two main metrics to measure QoS for SFP systems are completeness and soundness of results along with the response time. Completeness measures whether the system guarantees a certain proportion of all correct answers, while soundness measures the number of incorrect results due, for example, to approximation.

[P4] Time Model. In flow-processing applications time plays a central role. Information items are situated in time and an SFP system may provide time for each data item either explicitly or implicitly. In the first case, time is explicitly present in the data-flow while in the latter case the system assigns some timestamp or interval to each incoming item. Current SFP systems either encode time using RDF (by using an RDF node), or add a timestamp or interval to information items, which thus become quads or quintuples instead of triples.

[P5] Time Semantics. Time can be modeled using point-based semantics or interval-based semantics. The point based semantics associates each information item in the data-flow a single point in time (e.g. the occurrence of the event or the incoming time in the system). In contrast, interval-based semantics defines an interval of validity for the associated information.

[P6] Query Model. In the context of SFP, the query model is a discriminating property between systems. Systems like EP-SPARQL [4] define *pattern matching* queries through a set of primitive operators (e.g. sequences). Conversely, systems like C-SPARQL [10] extend *declarative languages* like SQL, augmenting them with operators like windows to limit the scope of processing. The query model also defines when queries are evaluated. The evaluation can be either reactive (the query is triggered when new data arrives), or periodic (the query is executed at a specified interval of time).

	C1: Managing Background Data	C2: Inference Expressivity	C3: Time Modeling	C4: Querying	C5: Managing Bursts
P1: Background Data	•	•		•	
P2: Inference Support		•			
P3: Quality of Service		•	•	•	•
P4: Time Model		•	•	•	
P5: Time Semantics		•	•		
P6: Query Model	•			•	
P7: Distribution	•			•	•

Fig. 2. Relations between the challenges and SFP properties

[P7] Distribution. To better support large scale scenarios, with sources of information potentially distributed over a wide geographical area, SFP systems may enable processors (see Figure 1) to be distributed among different physical machines. Distribution enables the concurrent execution of different queries at different nodes, but also the incremental evaluation of the building blocks of a single query on different machines. In the latter case, distribution can be used to push filtering operators as close as possible to the sources of the streaming information, to reduce the volume of data propagated over the network.

4 Challenges

Defining the challenges we rely on the following assumptions. They clearly define the scope of our analysis, and thus the area of validity of our results.

- **SFP systems distinguish between stream data and background data.** Stream data changes at high frequency while background data is static.
- **Streamed data does not affect the schema; no schema information is present in the stream.** In the Semantic Web, schema is defined by ontologies describing a conceptualization for a domain of interest. Since SFP systems assume that schema information does not change frequently it is not present in the stream. Note that this does not contradict the Semantic Web's Open-World-Assumption: an SFP system's inference process may still discover new schema statements as long as the reasoning remains monotone.
- **Only deductive and analytical processing is considered.** To limit the scope of our paper, we do not consider inductive processing (e.g., inductive reasoning). It is based on completely different methods and therefore introduces new challenges and requires a separate evaluation methodology.

We identified five classes of challenges that affect both the design and the development of a SFP system: *Managing Background Data*, *Inference Expressivity*, *Time Modeling*, and *Querying*, *Managing Bursts*. Each of them relate to one or more properties of the SFP systems, as shown in Figure 2.

[C1] Managing Background Data. Several challenges are connected to handling background data (P1) next to streaming information. First of all, storing

and manipulating background data might be difficult due to the *size* of the data, which can stress the machine resources. Data that greatly exceeds the size of main memory, requires algorithms to govern the data transfer between disk and memory.

Additional challenges derive from the complexity of queries over background data (P6). Queries may require to combine, i.e., join large portions of background data together with the elements in the flow. This poses strict timing constraints to processing, thus demanding for the definition and maintenance of suitable data structures for efficient retrieval and processing of information. This may involve changes to the background data, that needs to be timely propagated to processors. Partial results from the flow computation might become invalid, if in the meantime the background data has changed—an even more challenging aspect in the case of parallel and distributed processing (P7).

An SFP system must develop efficient mechanisms to handle all these issues, and their design and implementation is certainly not trivial. Therefore, efficient mechanisms for storing, accessing, and updating background data are crucial and should be properly considered in the evaluation of such systems.

[C2] Expressive Power of Inference. The support for inference (P2) is the distinguishing feature of SFP over IFP systems and introduces serious challenges.

The super-linearity of reasoning (quadratic for RDFS to super-exponential for OWL 2) requires to carefully balance the expressive power of the inference mechanism and performance. Even though inference can be limited to become tractable the fast change rate inherently present in a data-flow imposes strict constraints on the inference process (P3).

Inference requires a frequent interaction between background and stream data (P1), as all SFP systems store schema independent from the flow-data. Efficient mechanisms for storing as well as accessing the schema guarantee fast inference over the flow-data. Entailment regimes like RDFS produce many duplicates requiring an SFP system to handle repeatedly inserted information.

One additional challenge in the inference process is connected with the validity of the information in the system (and this strictly relates it to the properties P4 and P5). If a triple, for example, is no longer valid (e.g., because the active window has moved), then the inference process might have to be repeated to verify whether some conclusions still hold or should be retracted.

[C3] Time Modeling. This challenge differs from the others because it relates to the design of the system while the others primarily affect its execution. In fact, choosing a specific model –and a corresponding semantics– for representing time (P4 and P5) can significantly impact the performance of the system (P3). For example, it has been proven in [25] that the use of an interval-based semantics rather than a point-based semantics may negatively impact the tractability of some time-based operators (e.g., next, sequences). Therefore, the designer of an SFP system must carefully analyze the requisites of the system in order to choose an appropriate time model in order not to jeopardize its performance.

The current RDF data model includes no notion of time, which led existing SFP systems to extend RDF in several ways to handle time, e.g., by

time-stamping the triples —with potential serious consequences for complex processing tasks such as reasoning on the data. Suppose that the data exploits the RDFS semantics which allows reasoning by an exhaustive application of if-then rules. If the RDF triples used to derive some conclusions are no longer valid, then it is unclear what happens to the derivation. All these uncertainties can be clarified by a formal definition of the model and semantics of time associated to RDF data, but currently there is no clear consensus on this aspect, and this hampers an understanding of the consequences of the processing of SFP systems.

[C4] Querying. The query model determines the processing strategy (P6). A key challenge is its definition for stream and background data (P1) that can satisfy application level requirements on expressive power and ease of use, while keeping the processing as simple and efficient as possible (P3).

An important challenge for CEP-inspired languages is *the choice of an appropriate strategy for storing, accessing, and discarding partial results*. This is even more important when dealing with aggregates, in particular under non-shrinking semantics [9], i.e., when we are not only interested in the number of items in an aggregate but also in the items themselves. (cf. Section 5, S4).

Languages may include *operators that implicitly determine the scope of processing*, e.g., time-constrained sequences. Similarly, in DSMS-inspired transforming languages, the type and size of windows determines the portion of flow-data considered for processing (P4). In both cases, isolating the elements that are relevant for processing is a key challenge. An inappropriate choice may negatively impact the performance of an SFP system: A window too small may never contain enough information to provide the desired results; a window too large may hamper the system's response time. Unsuitable strategies for storing and pruning partial results may further negatively influence response time.

Other challenges rises from the *mechanism for triggering queries* and the *management of multiple queries*. Increasing the frequency of query evaluation may decrease the system's response time while too infrequent evaluations may prevent the detection of critical situations—both resulting in decreasing system performance. SFP systems must be able to develop techniques for sharing the state of partial results that are common to multiple queries, thus reducing memory requirements and processing effort. The effort of managing multiple queries increases in presence of distributed settings (P7) by the necessity of concerting the distribution of operations over available resources with respect to processing capabilities, connectivity, and their geographical location.

[C5] Managing Bursts. SPF systems must be able to continuously provide timely answers to queries even in presence of sudden bursts. This strictly relates to the property P3: indeed, depending from the QoS agreements between the system and the users, it may be acceptable to sacrifice completeness of results for the sake of guaranteeing lower response times. Moreover, managing bursts also requires a careful design of the mapping of processing tasks to available processing components, enabling load balancing and avoiding bottlenecks. This issue becomes even more relevant in parallel and distributed systems (P7).

5 Seven Commandments of Benchmarking SFP Systems

The evaluation of a system performance is done by changing the environment and/or the system parameters and observing the behavior of some measurable Key Performance Indicators (KPIs) as these changes occur. The goal of a benchmark consists in designing a number of stress tests so that the user can measure how different systems react to the same changes, considering the same KPIs. These stress tests should properly create situations when the system is called to deal with the challenges of the domain. The LR benchmark, for example, "is designed to measure how well a system can meet real-time query response requirements in processing high-volume streaming and historical data." [6].

In this section, we first define a number of KPIs to evaluate SFP systems with respect to the challenges identified in Section 4. Then, we design some stress tests to measure and compare the performance of various systems. We thereby analyze to what extent current benchmarking tools cover such stress tests (see Table 1), and provide guidelines how the missing parts can be implemented.

Note that *we provide no unified benchmark but a unified model for systematically benchmarking aspects of SFP systems by stress tests.* An actual implementation of these stress tests will depend first on the actual SFP system and second on the use-cases at hand, and is beyond the scope of this paper.

5.1 Key Performance Indicators

In contrast to offline systems, *SFP systems are reactive.* A delay exists between the points in time when the system consumes an input element and it reports the results of its processing. If the system load exceeds available resources either this delay compromises system reactivity or the system has to drop data.

All benchmarks for SFP systems use throughput as their KPI. This choice yet ignores other criteria that were reported for IFP systems in [15]. We hence identified the following three KPIs as the most suitable regarding our context. Interestingly, they were also used used for the evaluation (yet not benchmarking) of most the principal current SFP systems.

- **Response time** over all queries (Average/x^{th}Percentile/Maximum).
- **Maximum input throughput** in terms of number of data element in the input stream consumed by the system per time unit.
- **Minimum time to accuracy and the minimum time to completion** for all queries [19].[2]

Stressing a system means exploring the input space and identifying best, average, and –most importantly– worst cases for its performance, i.e., the conditions under which the system performs how in relation to the KPIs.

[2] This includes recall, precision and error rate in relation to processing time.

Table 1. Stress tests existing benchmarks support. P indicates a potential support or a partial implementation for stress testing. (a) load balancing, (b) simple, (c) sequential or (d) temporal joins flow-flow data, (e) joins on flow-background data, aggregates under shrinking (f) and non-shrinking semantics (g), (h) out-of-order or (i) missing data, (j) inference, and finally (k) changes in background data.

Benchmark	**S1** (a)	**S2** (b) / (c) / (d)	**S3** (e)	**S4** (f) / (g)	**S5** (h) / (i)	**S6** (j)	**S7** (k)
LR	P	Yes/ Yes/ P	No	No/ Yes	P/ P	No	No
FFD	P	Yes/ P/ Yes	No	No/ Yes	No/ P	P	No
SR	P	Yes/ P/ P	Yes	P/ Yes	P/ P	Yes	P
SLD	P	Yes/ P/ P	Yes	P/ Yes	P/ P	P	P

5.2 Stress Tests

After identifying the KPIs, the definition of stress tests first involves diagnosing *which* parameters to manipulate to change the input of the system. In the case of SFP systems, these parameters have some impact on background data, streaming data, input rate, etc. It is important to devise *how* to change these parameters to achieve the purpose of the test, i.e. to properly impact on the desired KPIs.

In this section we present the *seven commandments* we worked out based on our study of the challenges in Section 4. Each commandment represents one of the stress tests that in our opinion best suit the evaluation of SFP systems. We show how the current benchmarks address these tests in Table 1. We observe that all the benchmarks identified in Section 2 either implemented one or more of these stress tests or could implement them (indicated by "P"). However, no existing benchmark fully implements all of them.

[S1] Load Balancing [Relates to C5]. SFP systems usually consider multiple input flows of information, with possible bursts (C5). Therefore, the SFP system must implement a proper mapping of operators over available processors and good load balancing strategies.

Finding potential bottlenecks in settings in which many queries are deployed and multiple processors are available is extremely difficult. However, benchmarks can empirically evaluate a system under various conditions by repeatedly applying a set of changes to the input. In particular, it is possible to stress the system by (*i*) changing the load of every stream relative to the others at random, (*ii*) creating bursts on an increasing number of input streams, and by (*iii*) dynamically switching data sources to provide their input on some other data flow. All current benchmarks identified in Section 2 provide streaming data from sensors, and therefore implement variants of this stress test. However, the sensors in SR can only emit data on regular stable intervals. SLD and LR offer support for skewed distributions for the generation rate of different streams, although the specifications do not clearly state to what extent the skew can be controlled.

[S2] Joins and Inference on Flow Data Only [Relates to C3, C4]. In order to stress the joins between bindings of flow data we need to distinguish

between simple, sequential, and temporal joins. Simple joins put no further constraints on the join but the join-equality. Sequential joins add a sequential constraint (like the `SEQ`-operator [4]). Temporal joins further extend sequential joins by enabling advanced temporal constraints such as Allen's intervals [3]. Note that both sequential and temporal joins require that the system defines an ordering of the flow-data (C3) as well as a proper extension of the query language (C4).

A stress test to measure the performance of data joins has to consider increasingly complex cascades of joins. For testing sequential and temporal joins a benchmark will have to add further constraints on the joins, which have to be reflected in the data. The current benchmarks LR and SLD provide data and use-cases for sequential joins but at the moment none of them implements stress tests for temporal joins—although all datasets would allow to. Therefore, a full implementation of this stress-test is currently unavailable in these benchmarks.

[S3] Joins and Inference in Flow and Background Data [Relates to C1, C4]. In contrast to joins on flow data only, joining stream and background data is not subject to any ordering and hence always results in simple joins. These can be stress-tested by considering single joins and increasingly complex cascades thereof. Notice that systems often exploit the combination of flow and background data to perform inference. In this context, the ability of the system to manage background data (C1) is crucial, since complex reasoning tasks (C4) can require frequent and repeated access to background data.

Currently, both the SR and SLD benchmarks only provide a few fixed queries. They are not parameterized, and thus do not allow an exhaustive assessment of join performance. Furthermore, only SR and SLD can stress an SFP by considering the accesses to background data that is stored in the disk. Conversely, the background data of LR and FFD easily fits into the main memory.

[S4] Aggregates [Relates to C3, C4]. Aggregates enable computation on groups of entities or literals. Such computations include statistics such as counts, averages but also any other arithmetic operation on groups nodes that fulfill a grouping constraint. We distinguish between aggregating over entities and literals. In contrast to literal aggregates, entities aggregates refer to groups of actual entities and not data values. Consider, for example, detecting situations where more than n people with similar interest are watching the same show.

We refer to detecting the sole event as shrinking semantics, i.e., we are *not* interested in the actual people but only some statistics about them. Referring to the actual entities taking part in the aggregate (i.e., the actual people watching the show) is called non-shrinking semantics [9]. We may assess both types by testing *a*) how the system scales with an increasing number of groups (lots of shows, n small), *b*) by increasing the complexity of the grouping constraints (complex definition of similar interests) and *c*) by adjusting the data such that there will be a lot of candidates for groups of which only a small number will finally fulfill the grouping criterion (lots of shows with a number of viewers just below the threshold n).

In contrast to shrinking semantics, non-shrinking semantics are not directly supported by standard SPARQL and also not implemented by any of the existing

benchmarks. All of the benchmarks test aggregates in a limited scope, e.g., by implementing single queries (SR, SLD) or the expected outcome (LR).

[S5] Unexpected Data [Relates to C3, C4]. In distributed settings, SFP systems have to deal with out-of-order arrival of information and data loss. This may affect the correctness of query answers, especially (C3) when temporal operators and constraints are involved. We can measure the ability to handle out-of-order observations by (*i*) increasing the number of events arriving not in the expected order; (*ii*) by testing the amount of time or data which can be handled until some out-of-order observation will be no longer considered for processing. SPARQL OPTIONAL operators, for example allow answering a query even if some data is still missing (C4). In both cases the benchmark should measure precision and recall of the amount of missing data. Interestingly, none of the current benchmarks implements tests for out-of-order events or missing data.

While the ability to deal with noisy data is a relevant problem it is our form believe that this must be handled outside the core query processing. Consequently, we did not add stress-tests for handling noisy data.

[S6] Schema [Relates to C1, C2]. Since the schema of both the stream and the background data is known ex-ante, we can only evaluate the system's ability to handle (*i*) an increasing number of statements in the schema (i.e., axioms of the system's ontology), and (*ii*) statements that generate a more complex reasoning. In this last case the system needs to provide inference services (C2).

Number of Axioms. When testing an SFP system by increasing the number of axioms in its ontology it is fundamental to add *new* axioms that could not have been deduced from existing ones. Moreover, the expressive power should not increase as this will spoil the results of this test. SR and SLD are the only benchmarks with ontology schemata. In spite of the several thousand axioms the ontologies comprise the number of axioms involved in these benchmarks' queries is roughly one per cent of that number.

Expressive Power. Increasing the expressive power of the schema not only for the background data but also of the flow data may stress an SFP system significantly [9]. Evaluating the impact of expressive power requires changing the constraints or rules applied by the reasoner, while leaving the ontology unchanged, e.g., by implementing different combinations of the RDFS inference rules or different profiles of OWL 2. The variation in complexity must have some effect on the performance of the inference engine. Adding, for example, disjunction to the reasoner only makes sense in case the ontology contains disjunctive axioms.

In spite of missing features like negation, testing variations of the expressive power is possible in SR and SLD as they refer to some OWL 2-DL ontologies. Currently, they only test *whether* RDFS subclass reasoning is possible but do not measure the impact on KPIs when varying the expressive power. On the other hand, works like [9] provide a stress test for inference on transitive properties under RDFS semantics.

[S7] Changes in Background-Data [Relates to C1, C2] Nearly all systems identified in [15] consider background data in answering queries and this by precompiling the query. When the background data changes (C1), those parts have

to be re-compiled and in this intermediate state processing may be delayed or corrupted,[3] further worsened by the presence of inference services (C2). Stress-testing changes in background data should aim at varying the update frequency and the sheer amount of data that is subject to an update. Addressing the query-related part of the background data it should force the system to access background-data from disk as much as possible.

Currently, no benchmark implements this stress test, and only benchmarks that use datasets with rich background data can properly implement it, which is not the case for LR and FFD. The SLD and SR benchmarks do support such data and are suitable for this task. In particular for the SR system, we can simply increase the background data by all those datasets in the LOD cloud for which we may establish links to the GeoNames dataset.

6 Conclusion and Future Work

SFP systems are becoming increasingly popular for processing flows of semantically annotated data on-line and on large scale. Yet, the field of SFP lacks a classification scheme such as [15] for understanding and comparing existing systems. Even more significanlty, there is a lack of common agreement of which are the key performance indicators in the field, and they can be evaluated. A few good proposals for benchmarking SFP systems were published recently [21, 26], but none of them has (yet) come up with a pair of simulator/validator systems comparable to what the LR benchmark provides for IFP systems.

In this paper we diagnosed this research gap and approached the problem of benchmarking SFP systems from another perspective, following a top-down approach. We identified those properties of SFP systems relevant for understanding the key challenges SFP system face and defining the key performance indicators that allow to assess such challenges.

Starting from this analysis, we proposed seven *commandments* for defining a set of benchmarks that comprehensively stress test SFP systems in relation to precisely defined KPIs. We worked out these commandments as currently the most important for benchmarking current SFP systems. With new features for SFP systems this list will certainly have to be extended. For the same reasons as the LR benchmark, we provided no algorithm for implementing a benchmark nor did we address the definition of a common protocol for running a concrete benchmark on different systems. Instead we provide clear guidelines that specify how concrete benchmarks can implement relevant stress tests for SFP systems.

It is our firm belief that following these guidelines will enable implementing new or adjusting existing benchmarks, thus making it possible to realize a thorough evaluation and comparison of SFP systems, clearly spotting their strenghts and weaknesses. The tale of understanding SFP systems by systematic evaluation and comparison has only just begun.

[3] Note that a change in background data does *not* allow for a change in the schema.

References

[1] Abadi, D., Carney, D., Çetintemel, U.U.G., Cherniack, M., Convey, C., Lee, S., Stonebraker, M., Tatbul, N., Zdonik, S.: Aurora: a new model and architecture for data stream management. VLDB J. 12(2), 120–139 (2003)

[2] Agrawal, J., Diao, Y., Gyllstrom, D., Immerman, N.: Efficient pattern matching over event streams. In: Wang, J.T.L. (ed.) Proc. SIGMOD 2008, p. 147. ACM (2008)

[3] Allen, J.F.: Maintaining knowledge about temporal intervals. Commun. of the ACM 26(11), 832–843 (1983)

[4] Anicic, D., Fodor, P., Rudolph, S., Stojanovic, N.: EP-SPARQL: a unified language for event processing and stream reasoning. In: Srinivasan, S., Ramamritham, K., Kumar, A., Ravindra, M.P., Bertino, E., Kumar, R. (eds.) Proc. WWW 2011, pp. 635–644. ACM (2011)

[5] Arasu, A., Babcock, B., Babu, S., Datar, M., Ito, K., Nishizawa, I., Rosenstein, J., Widom, J.: STREAM: The Stanford Stream Data Manager. IEEE Data Eng. Bull., 19–26 (2003)

[6] Arasu, A., Cherniack, M., Galvez, E., Maier, D., Maskey, A.S., Ryvkina, E., Stonebraker, M., Tibbetts, R.: Linear Road: A Stream Data Management Benchmark. VLDB J. (2004)

[7] Babcock, B., Babu, S., Datar, M., Motwani, R., Widom, J.: Models and issues in data stream systems. In: Popa, L., Abiteboul, S., Kolaitis, P.G. (eds.) Proc. PODS 2002, pp. 1–16. ACM (2002)

[8] Bai, Y., Thakkar, H., Wang, H., Luo, C., Zaniolo, C.: A data stream language and system designed for power and extensibility. In: Yu, P.S., Tsotras, V.J., Fox, E.A., Liu, B. (eds.) Proc. CIKM 2006, pp. 337–346. ACM (2006)

[9] Barbieri, D.F., Braga, D., Ceri, S., Della Valle, E., Grossniklaus, M.: Incremental reasoning on streams and rich background knowledge. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010, Part I. LNCS, vol. 6088, pp. 1–15. Springer, Heidelberg (2010)

[10] Barbieri, D.F., Braga, D., Ceri, S., Della Valle, E., Grossniklaus, M.: C-SPARQL: A Continuous Query Language for RDF Data Streams. Int. J. of Semantic Computing 4(1), 3–25 (2010)

[11] Barbieri, D., Braga, D., Ceri, S., Della Valle, E., Grossniklaus, M.: C-SPARQL: SPARQL for continuous querying. In: Quemada, J., León, G., Maarek, Y.S., Nejdl, W. (eds.) Proc. WWW 2009, pp. 1061–1062. ACM (2009)

[12] Brenna, L., Demers, A., Gehrke, J., Hong, M., Ossher, J., Panda, B., Riedewald, M., Thatte, M., White, W.: Cayuga: A High-Performance Event Processing Engine. In: Chan, C.Y., Ooi, B.C., Zhou, A. (eds.) Proc. SIGMOD 2007, pp. 1100–1102. ACM (2007)

[13] Chandrasekaran, S., Cooper, O., Deshpande, A., Franklin, M.J., Hellerstein, J.M., Hong, W., Krishnamurthy, S., Madden, S., Reiss, F., Shah, M.A.: TelegraphCQ: Continuous Dataflow Processing. In: Halevy, A.Y., Ives, Z.G., Doan, A. (eds.) Proc. SIGMOD 2003, p. 668. ACM (2003)

[14] Cugola, G., Margara, A.: Complex event processing with T-REX. J. Syst. Softw. 85(8), 1709–1728 (2012)

[15] Cugola, G., Margara, A.: Processing Flows of Information: from Data Stream to Complex Event Processing. ACM Comput. Surv. 44(3), 1–62 (2012)

[16] Della Valle, E., Ceri, S., Milano, P., Van Harmelen, F.: It's a Streaming World! Reasoning upon Rapidly Changing Information. J. Intell. Syst., IEEE (2009)

[17] Etzion, O., Niblett, P.: Event Processing In Action. Manning Publications Co., Greenwich (2010)
[18] Gray, J.: The Benchmark Handbook for Database and Transaction Systems, 2nd edn. Morgan Kaufmann (1993)
[19] Hellerstein, J.M., Haas, P.J., Wang, H.J.: Online Aggregation. In: Peckham, J. (ed.) Proc. SIGMOD 1997, pp. 171–182. ACM (1997)
[20] Le-Phuoc, D., Dao-Tran, M., Parreira, J.X., Hauswirth, M.: A Native and Adaptive Approach for Unified Processing of Linked Streams and Linked Data. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 370–388. Springer, Heidelberg (2011)
[21] Le-Phuoc, D., Dao-Tran, M., Pham, M.-D., Boncz, P., Eiter, T., Fink, M.: Linked Stream Data Processing Engines: Facts and Figures. In: Cudré-Mauroux, P., et al. (eds.) ISWC 2012, Part II. LNCS, vol. 7650, pp. 300–312. Springer, Heidelberg (2012)
[22] Luckham, D.: The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems. Addison-Wesley (2002)
[23] Tichy, W.F., Lukowicz, P., Prechelt, L., Heinz, E.A.: A Quantitative Evaluation Study in Computer Science. J. Syst. and Softw. 28(1), 9–18 (1995)
[24] Wainer, J., Novoa Barsottini, C.G., Lacerda, D., Magalhães de Marco, L.R.: Empirical evaluation in Computer Science research published by ACM. J. Inform. and Softw. Tech. 51(6), 1081–1085 (2009)
[25] White, W., Riedewald, M., Gehrke, J., Demers, A.: What is "next" in event processing? In: Libkin, L. (ed.) Proc. PODS 2007, pp. 263–272. ACM (2007)
[26] Zhang, Y., Duc, P.M., Corcho, O., Calbimonte, J.-P.: SRBench: A Streaming RDF/SPARQL Benchmark. In: Cudré-Mauroux, P., et al. (eds.) ISWC 2012, Part I. LNCS, vol. 7649, pp. 641–657. Springer, Heidelberg (2012)

Graph-Based Ontology Classification in OWL 2 QL

Domenico Lembo, Valerio Santarelli, and Domenico Fabio Savo

Dipartimento di Ing. Informatica, Automatica e Gestionale "Antonio Ruberti"
Sapienza Università di Roma
Via Ariosto 25, I-00186 Roma, Italy
{lembo,santarelli,savo}@dis.uniroma1.it

Abstract. Ontology classification is the reasoning service that computes all subsumption relationships inferred in an ontology between concept, role, and attribute names in the ontology signature. OWL 2 QL is a tractable profile of OWL 2 for which ontology classification is polynomial in the size of the ontology TBox. However, to date, no efficient methods and implementations specifically tailored to OWL 2 QL ontologies have been developed. In this paper, we provide a new algorithm for ontology classification in OWL 2 QL, which is based on the idea of encoding the ontology TBox into a directed graph and reducing core reasoning to computation of the transitive closure of the graph. We have implemented the algorithm in the QuOnto reasoner and extensively evaluated it over very large ontologies. Our experiments show that QuOnto outperforms various popular reasoners in classification of OWL 2 QL ontologies.

1 Introduction

Ontology classification is the problem of computing all subsumption relationships inferred in an ontology between predicate names in the ontology signature, i.e., named concepts (a.k.a. classes), roles (a.k.a. object-properties), and attributes (a.k.a. data-properties). It is considered a core service for ontology reasoning, which can be exploited for various tasks, at both design-time and run-time, ranging from ontology navigation and visualization to query answering.

Devising efficient ontology classification methods and implementations is a challenging issue, since classification is in general a costly operation. Most popular reasoners for Description Logic (DL) ontologies, i.e., OWL ontologies, such as Pellet [22], Racer [11], FACT++ [23], and HermiT [9], offer highly optimized classification services for expressive DLs. Various experimental studies show that such reasoners have reached very good performances through the years. However, they are still not able to efficiently classify very large ontologies, such as the full versions of GALEN [21] or of the FMA ontology [10].

Whereas the above tools use algorithms based on model construction through tableau (or hyper-tableau [9]), the CB reasoner [14] for the Horn-$\mathcal{SHIQ}$ DL is a *consequence-driven* reasoner. The use of this technique allows CB to obtain an impressive gain on very large ontologies, such as full GALEN. However, the

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 320–334, 2013.

current implementation of the CB reasoner is rather specific for particular fragments of Horn-$\mathcal{SHIQ}$ (and incomplete for the general case) [14]. For example, it does not allow for classification of properties.

Other recently developed tools, such as Snorocket [17], ELK [15], and JCEL [19], are specifically tailored to intensional reasoning over logics of the $\mathcal{EL}$ family, and show excellent performances in classification of ontologies specified in such languages, which are the logical underpinning of OWL 2 EL, one of the tractable profile of OWL 2 [20].

Instead, to the best of our knowledge, ontology classification in the other OWL 2 profiles has received so far little attention. In particular, classification in OWL 2 RL has been investigated only in [16], whereas, to date, no techniques have been developed that are specifically tailored to intensional reasoning in OWL 2 QL, the "data oriented" profile of OWL 2, nor for any logic of the *DL-Lite* family [6], which constitutes the logical underpinning of OWL 2 QL. Our aim is then to contribute to fill this lack on OWL 2 QL, encouraged also by the fact that such language, like all logics of the *DL-Lite* family, allows for tractable intensional reasoning, and in particular for PTime ontology classification, as it immediately follows from the results in [6].

In this paper, we thus provide a new method for ontology classification in the OWL 2 QL profile. In our technique, we encode the ontology terminology (TBox) into a graph, and compute the transitive closure of the graph to then obtain the ontology classification. The analogy between simple inference rules in DLs and graph reachability is indeed very natural: consider, for example, an ontology containing the subsumptions $A_1 \sqsubseteq A_2$ and $A_2 \sqsubseteq A_3$, where A_1, A_2, and A_3 are class names in the ontology signature. We can then associate to this ontology a graph having three nodes labeled with A_1, A_2, and A_3, respectively, an edge from A_1 to A_2 and an edge from A_2 to A_3. It is straightforward to see that A_3 is reachable from A_1, and therefore an edge from A_1 to A_3 is contained in the transitive closure of the graph. This corresponds to the inferred subsumption $A_1 \sqsubseteq A_3$. On the other hand, things become soon much more complicated when complex (OWL) axioms come into play.

In this respect, we will show that for an OWL 2 QL ontology it is possible to easily construct a graph whose closure constitutes the major sub-task in ontology classification, because it allows us to obtain all subsumptions inferred by the "positive knowledge" specified by the TBox. We will show that the computed classification misses only "trivial" subsumptions inferred by unsatisfiable predicates, i.e., named classes (resp. properties) that always have an empty interpretation in every model of the ontology, and that are therefore subsumed by every class (resp. property) in the ontology signature. We therefore provide an algorithm that, exploiting the transitive closure of the graph, computes all unsatisfiable predicates, thus allowing us to obtain a complete ontology classification. We notice that the presence of unsatisfiable predicates in an ontology is mainly due to errors in the design. However, it is not rare to find such predicates, especially in very large ontologies or in ontologies that are still "under construction". In particular, we could find unsatisfiable concepts even in some

benchmark ontologies we used in our experiments (cf. Section 4). Of course, already debugged ontologies might not present such predicates [13,12]. In this case, one can avoid executing our algorithm for computing unsatisfiable predicates.

We have implemented our technique in a new module of QuOnto [1], the reasoner at the base of the Mastro [5,7] system, and have carried out extensive experimentation, focusing in particular on very large ontologies. We have considered well-known ontologies, often used as benchmark for ontology classification, and we have suitably approximated them in OWL 2 QL.

QuOnto showed better performances, in some cases corresponding to enormous gains, with respect to tableau-based reasoners (in particular, Pellet, Fact++, and HermiT). We also obtained comparable or better results with respect to the CB reasoner, for almost all ontologies considered, but, differently from CB reasoner, we were always able to compute a complete classification. We finally compared QuOnto with ELK, one of the most performing reasoner for $\mathcal{EL}$, for those approximated ontologies that turned out to be both in OWL 2 QL and OWL 2 EL, obtaining similar performances in almost all cases.

We conclude by noticing that, even though we refer here to OWL 2 QL, our algorithms and implementations can be easily adapted to deal with all logics of the *DL-Lite* family mentioned in [6], excluding those allowing for the use of conjunction in the left-hand side of inclusion assertions or the use of n-ary relations instead of binary roles.

The rest of the paper is organized as follows. In Section 2, we provide some preliminaries. In Section 3, we describe our technique for ontology classification in OWL 2 QL. In Section 4, we describe our experimentation, and finally, in Section 5, we conclude the paper.

2 Preliminaries

In this section, we present some basic notions on DL ontologies, the formal underpinning of the OWL 2 language, and on OWL 2 QL. We also recall some notions of graph theory needed later on.

Description Logic Ontologies. We consider a signature Σ, partitioned in two disjoint signatures, namely, Σ_P, containing symbols for predicates, i.e., atomic concepts, atomic roles, atomic attributes, and value-domains, and Σ_C, containing symbols for individual (object and value) constants. Complex concept, role, and attribute expressions are constructed starting from predicates of Σ_P by applying suitable constructs, which vary in different DL languages. Given a DL language $\mathcal{L}$, an $\mathcal{L}$-TBox (or simply a TBox, when $\mathcal{L}$ is clear) over Σ contains universally quantified first-order (FOL) assertions, i.e., axioms specifying general properties of concepts, roles, and attributes. Again, different DLs allow for different axioms. An $\mathcal{L}$-ABox (or simply an ABox, when $\mathcal{L}$ is clear) is a set of assertions on individual constants, which specify extensional knowledge. An $\mathcal{L}$-ontology $\mathcal{O}$ is constituted by both an $\mathcal{L}$-TBox $\mathcal{T}$ and an $\mathcal{L}$-ABox $\mathcal{A}$, denoted as $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$.

The semantics of a DL ontology $\mathcal{O}$ is given in terms of FOL interpretations (cf. [3]). We denote with $Mod(\mathcal{O})$ the set of models of $\mathcal{O}$, i.e., the set of FOL-interpretations that satisfy all TBox axioms and ABox assertions in $\mathcal{O}$, where

the definition of satisfaction depends on the DL language in which $\mathcal{O}$ is specified. An ontology $\mathcal{O}$ is *satisfiable* if $Mod(\mathcal{O}) \neq \emptyset$. A FOL-sentence ϕ is *entailed* by an ontology $\mathcal{O}$, denoted $\mathcal{O} \models \phi$, if ϕ is satisfied by every model in $Mod(\mathcal{O})$. All the above notions naturally apply to a TBox $\mathcal{T}$ alone.

Traditional intensional reasoning tasks with respect to a given TBox are verification of subsumption and satisfiability of concepts, roles, and attributes [3]. More precisely, a concept C_1 is *subsumed in* $\mathcal{T}$ by a concept C_2, written $\mathcal{T} \models C_1 \sqsubseteq C_2$, if, in every model I of $\mathcal{T}$, the interpretation of C_1, denoted C_1^I, is contained in the interpretation of C_2, denoted C_2^I, i.e., $C_1^I \subseteq C_2^I$ for every $I \in Mod(\mathcal{T})$. Furthermore, a concept C in $\mathcal{T}$ is unsatisfiable, which we wrote as $\mathcal{T} \models C \sqsubseteq \neg C$, if the interpretation of C is empty in every model of $\mathcal{T}$, i.e., $C^I = \emptyset$ for every $I \in Mod(\mathcal{T})$. Analogous definitions hold for roles and attributes.

Strictly related to the previous reasoning tasks is the classification inference service, which we focus on in this paper. Given a signature Σ_P and a TBox $\mathcal{T}$ over Σ_P, such a service allows to determine subsumption relationships in $\mathcal{T}$ between concepts, roles, and attributes in Σ_P. Therefore, classification allows to structure the terminology of $\mathcal{T}$ in the form of a subsumption hierarchy that provides useful information on the connection between different terms, and can be used to speed up other inference services. Here we define it more formally.

Definition 1. *Let $\mathcal{T}$ be a satisfiable $\mathcal{L}$-TBox over Σ_P. We define the $\mathcal{T}$-classification of Σ_P (or simply $\mathcal{T}$-classification when Σ_P is clear from the context) as the set of inclusion assertions defined as follows:*

Let S_1 and S_2 be either two concepts, roles, or attributes in Σ_P. If $\mathcal{T} \models S_1 \sqsubseteq S_2$ then $S_1 \sqsubseteq S_2$ belongs to the $\mathcal{T}$-classification of Σ_P.

The OWL 2 QL Language. We now present OWL 2 QL. We use the German notation for describing its constructs and axioms, and refer the reader to [20] for the OWL functional style syntax.

Expressions in OWL 2 QL are formed according to the following syntax:

$$
\begin{array}{lll}
B \longrightarrow A \mid \exists Q \mid \delta(U) & R \longrightarrow Q \mid \neg Q & E \longrightarrow \rho(U) \\
C \longrightarrow B \mid \neg B \mid \exists Q.A \mid \delta_F(U) & V \longrightarrow U \mid \neg U & F \longrightarrow T_1 \mid \cdots \mid T_n \\
Q \longrightarrow P \mid P^- & &
\end{array}
$$

where: A, P, and U are symbols in Σ_P denoting respectively an *atomic concept*, an *atomic role*, and an *atomic attribute*; P^- denotes the inverse of P; $\exists Q$, also called *unqualified existential role*, denotes the set of objects related to some object by the role Q; $\delta(U)$ denotes the *domain* of U, i.e., the set of objects that U relates to values; $\rho(U)$ denotes the *range* of U, i.e., the set of values related to objects by U; $T_1, \ldots, T_n$ denote n unbounded value-domains (i.e., datatypes); the concept $\exists Q.A$, or *qualified existential role*, denotes the *qualified domain* of Q with respect to A, i.e., the set of objects that Q relates to some instance of A. Similarly, $\delta_F(U)$ denotes the *qualified domain* of U with respect to a value-domain F, i.e., the set of objects that U relates to some value in F. In the following, we call B a *basic concept*, and Q a *basic role*.

An OWL 2 QL TBox $\mathcal{T}$ is a finite set of axioms of the form:

$$B \sqsubseteq C \qquad Q \sqsubseteq R \qquad U \sqsubseteq V \qquad E \sqsubseteq F$$

From left to right, the above axioms denote subsumptions between concepts, roles, attributes, and value-domains, respectively. We call *positive inclusions* axioms of the form $B_1 \sqsubseteq B_2$, $B_1 \sqsubseteq \exists Q.A$, $B_1 \sqsubseteq \delta_F(U)$, $Q_1 \sqsubseteq Q_2$, and $U_1 \sqsubseteq U_2$, *value-domain inclusions* axioms of the form $E \sqsubseteq F$, and *negative inclusions* axioms of the form $B_1 \sqsubseteq \neg B_2$, $Q_1 \sqsubseteq \neg Q_2$ and $U_1 \sqsubseteq \neg U_2$.

We notice that also other constructs and axioms are in fact allowed in OWL 2 QL. In particular, it allows for the use of $\delta_F(U)$ in the left-hand side of subsumptions, or in the right-hand side of negative inclusions, the use of "top" constructs in the left hand-side of subsumptions, corresponding to rdfs:Literal, owl:Thing, owl:topObjectProperty, and owl:topDataProperty, and the use of reflexivity and irreflexivity on roles (i.e., object-properties). For the sake of presentation, in this paper we prefer to not consider such aspects of OWL 2 QL, since their presence requires to burden our algorithms with some technicalities, which represent very minor contributions of our approach. Also, such constructs and axioms are rarely used in the practice, and in particular are never used in the benchmark ontologies considered in our experimentations (cf. Section 4). We notice however, that all the techniques presented in the following sections can be extended to full OWL 2 QL with minimal adaptations. Other constructs, such us symmetric or asymmetric roles, even though not explicitly mentioned, can be easily expressed by the OWL 2 QL syntax we consider.

As for OWL 2 QL ABoxes, we do not present them here, since we concentrate on intensional reasoning, and refer the interested reader to [20].

The semantics of OWL 2 QL ontologies and TBoxes is given in the standard way [20,3]. We only recall here that, datatypes, i.e., value-domains, have a fixed predefined interpretation, i.e., each datatype T_i is interpreted always in the same way, denoted $val(T_i)$, in every interpretation of the ontology. Notice also that OWL 2 QL supports only OWL datatypes such that the intersection of the value spaces of any set of these datatypes is either infinite or empty, i.e., for each $i, j \in \{1, \ldots, n\}$, it holds either that $val(T_i) \cap val(T_j)$ is infinite or $val(T_i) \cap val(T_j) = \emptyset$.

Graph Theory Notions. In this paper we use the term *digraph* to refer to a directed graph. We assume that a digraph $\mathcal{G}$ is a pair $(\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is a set of elements called *nodes*, and $\mathcal{E}$ is a set of ordered pairs (s, t) of nodes in $\mathcal{N}$, called *arcs*, where s is denoted the *source* of the arc, and t the *target* of the arc.

The transitive closure $\mathcal{G}^* = (\mathcal{N}, \mathcal{E}^*)$ of a digraph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is a digraph such that there is an arc in $\mathcal{E}^*$ having a node s as source and a node t as target if and only if there is a path from s to t in $\mathcal{G}$ [4]. Let $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ be a digraph, and let n be a node in $\mathcal{N}$. We denote with $\mathsf{predecessors}(n, \mathcal{G})$ the set of nodes p_n in $\mathcal{N}$ such that there exists in $\mathcal{E}$ an arc (p_n, n).

3 $\mathcal{T}$-Classification in OWL 2 QL

In this section we describe our approach to computing, given a signature Σ_P and an OWL 2 QL TBox $\mathcal{T}$ over Σ_P, the $\mathcal{T}$-classification of Σ_P.

In OWL 2 QL, a subsumption relation between two concepts, roles, or attributes in Σ_P, can be inferred by a TBox $\mathcal{T}$ if and only if (*i*) $\mathcal{T}$ contains such subsumption; (*ii*) $\mathcal{T}$ contains a set of positive inclusion assertions that together entail the subsumption; or (*iii*), trivially, the subsumed concept, role, or attribute is unsatisfiable in $\mathcal{T}$. The above observation is formalized as follows.

Theorem 1. *Let $\mathcal{T}$ be an OWL 2 QL TBox containing only positive inclusions, and let S_1 and S_2 be two atomic concepts, two atomic roles, or two atomic attributes. $S_1 \sqsubseteq S_2$ is entailed by $\mathcal{T}$ if and only if at least one of the following conditions holds:*

1. *a set $\mathcal{P}$ of positive inclusions exists in $\mathcal{T}$, such that $\mathcal{P} \models S_1 \sqsubseteq S_2$;*
2. *$\mathcal{T} \models S_1 \sqsubseteq \neg S_1$.*

Proof. (*sketch*) ($\Leftarrow$) This is trivially proven.
($\Rightarrow$) Assume $\mathcal{T} \models S_1 \sqsubseteq S_2$. Towards a contradiction, suppose that both statements 1 and 2 are false. If $\mathcal{T} \models S_1 \sqsubseteq S_2$ then the following cases are conceivable:

(a) $S_1 \sqsubseteq S_2 \in \mathcal{T}$, but this implies that statement 1 is true (contradiction);
(b) $S_1 \sqsubseteq S_2 \notin \mathcal{T}$ and S_1 is satisfiable. Since statement 1 does not hold, it remains that there exists a subset $\mathcal{T}'$ of $\mathcal{T}$ formed by positive inclusions and at least one negative inclusion such that $\mathcal{T}' \models S_1 \sqsubseteq S_2$. It can be shown that in OWL 2 QL negative inclusions do not concur in the entailment of positive inclusions [6], and therefore $S_1 \sqsubseteq S_2$ follows only from the positive inclusions of $\mathcal{T}'$, which contradicts that statement 1 is false;
(c) $S_1 \sqsubseteq S_2 \notin \mathcal{T}$ and S_1 is unsatisfiable. But then statement 2 is true (contradiction). ∎

Given a OWL 2 QL TBox $\mathcal{T}$ over a signature Σ_P, we use $\Phi_\mathcal{T}$ and $\Omega_\mathcal{T}$ to denote two sets of positive inclusions of the form $S_1 \sqsubseteq S_2$, with $S_1, S_2 \in \Sigma_P$, such that $\Phi_\mathcal{T}$ contains only positive inclusions for which statement 1 holds, and $\Omega_\mathcal{T}$ contains only positive inclusions for which statement 2 holds. It is easy to see that $\Phi_\mathcal{T}$ and $\Omega_\mathcal{T}$ are not disjoint. From Definition 1 and Theorem 1 it follows that the $\mathcal{T}$-classification coincides with the union of the sets $\Phi_\mathcal{T}$ and $\Omega_\mathcal{T}$.

In the following, we describe our approach to the computation of the $\mathcal{T}$-classification by firstly computing the set $\Phi_\mathcal{T}$, and then computing the set $\Omega_\mathcal{T}$.

Computation of $\Phi_\mathcal{T}$. Given an OWL 2 QL TBox $\mathcal{T}$, in order to compute $\Phi_\mathcal{T}$, we encode the set of positive inclusions in $\mathcal{T}$ into a digraph $\mathcal{G}_\mathcal{T}$ and compute the transitive closure of $\mathcal{G}_\mathcal{T}$ in such a way that each subsumption $S_1 \sqsubseteq S_2$ in $\Phi_\mathcal{T}$ corresponds to an arc (S_1, S_2) in such transitive closure, and vice versa. The following constructive definition describes the appropriate manner to obtain the digraph TBox representation for our aims.

Definition 2. *Let $\mathcal{T}$ be an OWL 2 QL TBox over a signature Σ_P. We call the digraph representation of $\mathcal{T}$ the digraph $\mathcal{G}_\mathcal{T} = (\mathcal{N}, \mathcal{E})$ built as follows:*

1. *for each atomic concept A in Σ_P, $\mathcal{N}$ contains the node A;*
2. *for each atomic role P in Σ_P, $\mathcal{N}$ contains the nodes P, P^-, $\exists P$, $\exists P^-$;*

3. *for each atomic attribute* U *in* Σ_P, $\mathcal{N}$ *contains the nodes* U *and* $\delta(U)$;
4. *for each concept inclusion* $B_1 \sqsubseteq B_2 \in \mathcal{T}$, $\mathcal{E}$ *contains the arc* (B_1, B_2);
5. *for each role inclusion* $Q_1 \sqsubseteq Q_2 \in \mathcal{T}$, $\mathcal{E}$ *contains the arcs* (Q_1, Q_2), (Q_1^-, Q_2^-), $(\exists Q_1, \exists Q_2)$, *and* $(\exists Q_1^-, \exists Q_2^-)$;
6. *for each attribute inclusion* $U_1 \sqsubseteq U_2 \in \mathcal{T}$, $\mathcal{E}$ *contains the arcs* (U_1, U_2) *and* $(\delta(U_1), \delta(U_2))$;
7. *for each concept inclusion* $B_1 \sqsubseteq \exists Q.A \in \mathcal{T}$, $\mathcal{E}$ *contains the arc* $(B_1, \exists Q)$;
8. *for each concept inclusion* $B_1 \sqsubseteq \delta_F(U) \in \mathcal{T}$, $\mathcal{E}$ *contains the arc* $(B_1, \delta(U))$.

The idea is that each node in the digraph $\mathcal{G}_\mathcal{T}$ represents a basic concept, a basic role or an attribute, and each arc models a positive inclusion, i.e., a subsumption, contained in $\mathcal{T}$, where the source node of the arc represents the left-hand side of the subsumption and the target node of the arc represents the right-hand side of the subsumption. Observe that for each role inclusion assertion $P_1 \sqsubseteq P_2$ in the TBox $\mathcal{T}$, we also represent as nodes and arcs in the digraph $\mathcal{G}_\mathcal{T}$ the entailed positive inclusions $P_1^- \sqsubseteq P_2^-$, $\exists P_1 \sqsubseteq \exists P_2$, and $\exists P_1^- \sqsubseteq \exists P_2^-$. We operate in a similar fashion for positive inclusions on attributes in $\mathcal{T}$.

Let $\mathcal{T}$ be an OWL 2 QL TBox and let $\mathcal{G}_\mathcal{T} = (\mathcal{N}, \mathcal{E})$ be its digraph representation. We denote with $\mathcal{G}_\mathcal{T}^* = (\mathcal{N}, \mathcal{E}^*)$ the transitive closure of $\mathcal{G}_\mathcal{T}$. Note that by definition of digraph transitive closure, for each node $n \in \mathcal{N}$ there exists in $\mathcal{E}^*$ an arc (n, n). Moreover, in what follows, we denote with $\alpha(\mathcal{E}^*)$ the set of arcs $(S_1, S_2) \in \mathcal{E}^*$ such that both terms S_1 and S_2 denote in $\mathcal{T}$ either two atomic concepts, two atomic roles, or two attributes. Then, the following property holds.

Theorem 2. *Let* $\mathcal{T}$ *be an OWL 2 QL TBox and let* $\mathcal{G}_\mathcal{T} = (\mathcal{N}, \mathcal{E})$ *be its digraph representation. Let* S_1 *and* S_2 *be two atomic concepts, two atomic roles, or two atomic attributes. An inclusion assertion* $S_1 \sqsubseteq S_2$ *belongs to* $\Phi_\mathcal{T}$ *if and only if there exists in* $\alpha(\mathcal{E}^*)$ *an arc* (S_1, S_2).

Proof. (*sketch*) ($\Leftarrow$) This is trivially proven.
($\Rightarrow$) To prove the thesis we need to introduce the notion of chase for an OWL 2 QL ontology, which is analogous to the notion of chase given in [6,8]. We first note that every positive inclusion in the TBox can be formulated as a FOL implication of the form

$$\forall \boldsymbol{x}, \boldsymbol{y}.S(\boldsymbol{x}, \boldsymbol{y}) \rightarrow \exists \boldsymbol{z}.\psi(\boldsymbol{x}, \boldsymbol{z}) \tag{1}$$

where S is an atomic concept, an atomic attribute, or an atomic role, ψ is a single atom or a conjunction of two atoms constructed on predicates of Σ_P, $\boldsymbol{x}$ is a vector of one or two variables, $\boldsymbol{y}$ and $\boldsymbol{z}$ are vectors of one or zero variables, i.e., they may be missing. For example, a positive inclusion of the form $A_1 \sqsubseteq A_2$ is written as $\forall x.A_1(x) \rightarrow A_2(x)$, the positive inclusion $\exists P_1^- \sqsubseteq \exists P_2.A$ is written as $\forall x, y.P_1(x, y) \rightarrow \exists z.P_2(y, z) \wedge A(z)$, or the inclusion $U_1 \sqsubseteq U_2$ is written as $\forall x, y.U_1(x, y) \rightarrow U_2(x, y)$.

Now, let $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ be an OWL 2 QL ontology. Our notion of chase is given inductively as follows. We pose $chase_0(\mathcal{O}) = \mathcal{A}$, and for every non-negative integer i, we define $chase_{i+1}(\mathcal{O})$ as the set of ABox assertions obtained from $chase_i(\mathcal{O})$ by applying the following rule:

CHASE RULE. Let I be a positive inclusion in $\mathcal{T}$ of the form (1). Let h be a homomorphism from $S(\boldsymbol{x}, \boldsymbol{y})$ to $chase_i(\mathcal{O})$ such that $h(S(\boldsymbol{x}, \boldsymbol{y})) = S(\boldsymbol{a}, \boldsymbol{b})$, and such that there is no extension of h to a homomorphism h' from $S(\boldsymbol{x}, \boldsymbol{y}) \wedge \psi(\boldsymbol{x}, \boldsymbol{z})$ to $chase_i(\mathcal{O})$ (we say in this case that I is applicable to $S(\boldsymbol{a}, \boldsymbol{b})$). Then $chase_{i+1}(\mathcal{O}) = chase_i(\mathcal{O}) \cup \{\psi(\boldsymbol{a}, n)\}$, where n is a fresh constant, i.e., a constant in Σ_C not occurring in $chase_i(\mathcal{O})$, if $\boldsymbol{z}$ is a single variable in (1), or $chase_{i+1}(\mathcal{O}) = chase_i(\mathcal{O}) \cup \{\psi(\boldsymbol{a})\}$, if $\boldsymbol{z}$ is absent in (1). We say that $chase_{i+1}(\mathcal{O})$ is obtained from $chase_i(\mathcal{O})$ via application of the positive inclusion I to $S(\boldsymbol{a}, \boldsymbol{b})$.

We assume that the chase rule is always executed in such a way that if a positive inclusion I becomes applicable to an ABox assertion β in a certain $chase_i(\mathcal{O})$, then there exists $j > i$ such that $chase_j(\mathcal{O})$ is obtained from $chase_{j-1}(\mathcal{O})$ via application of I to α. Then, we call *chase of* $\mathcal{O}$, denoted $chase(\mathcal{O})$, the set of ABox assertions obtained as the infinite union of all $chase_i(\mathcal{O})$, i.e., $chase(\mathcal{O}) = \bigcup_{i \in \mathbb{N}} chase_i(\mathcal{O})$. Associated to the chase, we consider the so-called *canonical interpretation of* $\mathcal{O}$, denoted $can(\mathcal{O})$, in which every constant is interpreted by itself, and for every predicate S, we have that $S^{can(\mathcal{O})} = \{\boldsymbol{a} \mid S(\boldsymbol{a}) \in chase(\mathcal{O})\}$. It is possible to show that $can(\mathcal{O})$ is a model of $\mathcal{O}$ [6].

Let us now turn back to our proof, and show that from the fact that an arc $(A_1, A_2) \notin \alpha(\mathcal{E}^*)$, where A_1 and A_2 are atomic concepts, it follows that there does not exist a set $\mathcal{P}$ of positive inclusions in $\mathcal{T}$ such that $\mathcal{P} \models A_1 \sqsubseteq A_2$. The cases of arcs between nodes corresponding to roles or attributes can be proved analogously. Let us consider any set $\mathcal{P} \subseteq \mathcal{T}$ of positive inclusions. To prove the thesis we construct a model I of $\mathcal{P}$ and show that if $(A_1, A_2) \notin \alpha(\mathcal{E}^*)$, I is not a model of $A_1 \sqsubseteq A_2$. To this aim, we consider the ABox $\mathcal{A}_{A_1} = \{A_1(d)\}$, where d is a constant in Σ_C, and the canonical interpretation $can(\mathcal{O}_P)$ of the ontology $\mathcal{O}_P = \langle \mathcal{P}, \mathcal{A}_{A_1} \rangle$, i.e., the model associated to $chase(\mathcal{O}_P)$. Since $can(\mathcal{O}_P)$ is a model of $\mathcal{O}_P$, it is also a model of $\mathcal{P}$. We show now that $can(\mathcal{O}_P)$ is not a model of $A_1 \sqsubseteq A_2$. Let us denote with $chase_i(\mathcal{O}_P)$ the chase obtained after i applications of the chase rule. We can now show that $chase_i(\mathcal{O}_P)$ contains an ABox assertion of the form $A(d)$ (resp. $P(d, n)$, $P(n, d)$, or $U(d, n)$) if and only if there exists an arc from A_1 to A (resp. to $\exists P$, $\exists P^-$, or $\delta(U)$) in $\mathcal{G}^*_{\mathcal{P}}$. The if direction of this property can be easily verified. For the only if direction we proceed by induction on the construction of the chase. The base step is indeed trivial. As for the inductive step, various cases are possible. We consider here the case in which $chase_{i+1}(\mathcal{O}_P)$ contains the fact $A(d)$ that is generated from $chase_i(\mathcal{O}_P)$ by applying the axiom $A' \sqsubseteq A$ of $\mathcal{P}$ (in fact its FOL version, according to our definition of chase). This means that $chase_i(\mathcal{O}_P)$ contains the ABox assertion $A'(d)$, and, by the inductive hypothesis, $\mathcal{G}^*_{\mathcal{P}}$ contains the arc (A_1, A'). It is easy then to see that $\mathcal{G}^*_{\mathcal{P}}$ contains the arc (A_1, A). Other possible cases can be proved in an analogous way. It is now very easy to conclude that $can(\mathcal{O}_P)$ is not a model of $A_1 \sqsubseteq A_2$, since the arc (A_1, A_2) is not in $\alpha(\mathcal{E}^*)$. ∎

We can then easily construct an algorithm, called ComputeΦ, that, taken as input an OWL 2 QL TBox $\mathcal{T}$, first builds the digraph $\mathcal{G}_{\mathcal{T}} = (\mathcal{N}, \mathcal{E})$ according

to Definition 2, then computes its transitive closure, and finally returns the set $\Phi_{\mathcal{T}}$, which contains an inclusion assertion $S_1 \sqsubseteq S_2$ for each arc $(S_1, S_2) \in \alpha(\mathcal{E}^*)$.

According to Theorem 2, ComputeΦ is sound and complete with respect to the problem of computing $\Phi_{\mathcal{T}}$ for any OWL 2 QL TBox $\mathcal{T}$ containing only positive inclusions.

Computation of $\Omega_{\mathcal{T}}$. In OWL 2 QL, unsatisfiability of concepts, roles, and attributes can mainly arise due to a malicious interaction of negative and positive inclusions. However, also disjoint value-domains, i.e., datatypes having empty intersection of their value spaces, can cause unsatisfiability. This can happen, due to the presence in the TBox of ill-defined value-domain inclusions, which can make one derive contradictory information. For instance, consider the TBox $\mathcal{T}$ containing the assertions $\rho(U) \sqsubseteq$ `xsd:dateTime` and $\rho(U) \sqsubseteq$ `xsd:integer`. Since the `xsd:dateTime` and `xsd:integer` datatypes are disjoint, we have that $\mathcal{T} \models U \sqsubseteq \neg U$. The detection of the situation above described is rather technical, and does not add particular value to our overall technique for identification of unsatisfiable predicates. Furthermore, this situation is quite rare in the practice (for example, no ill-typed attributes are present in the benchmark ontologies used in Section 4). Therefore, for the sake of presentation, we prefer here to not consider this case, and assume that the TBox does not contain value-domain inclusions. Furthermore, since under such assumption the treatment of attributes and roles is analogous, we limit here our attention to the case where the TBox does not contain axioms involving attributes. All results given below apply however also to full-fledged OWL 2 QL TBoxes.

We first observe that, according Definition 2, no node corresponding to a qualified existential role is created in the TBox digraph representation. This kind of node is indeed not useful for computing $\Phi_{\mathcal{T}}$. Differently, if one aims to identify every cause of unsatisfiability, the creation of nodes corresponding to a qualified existential role is needed. This is due to the fact that a TBox may entail that a qualified existential role $\exists P.A$ is unsatisfiable, even in case of satisfiability of $\exists P$. Specifically, this may occur in two instances: (i) if the TBox $\mathcal{T}$ entails the assertion $\exists P^- \sqsubseteq \neg A$, and (ii), the TBox $\mathcal{T}$ entails $A \sqsubseteq \neg A$. Clearly, in both cases the concept $\exists P.A$ is unsatisfiable. We therefore modify here Definition 2 by substituting Rule 7 with the following one:

7*. for each concept inclusion $B_1 \sqsubseteq \exists Q.A \in \mathcal{T}$, $\mathcal{N}$ contains the node $\exists Q.A$, and $\mathcal{E}$ contains the arches $(B_1, \exists Q.A)$ and $(\exists Q.A, \exists Q)$;

From now on, we adopt the digraph representation built according to Definition 2, where rule 7* replaces rule 7, and, according to the above assumptions, we consider only OWL 2 QL TBoxes that do not contain axioms involving attributes in Σ_P. Given one such TBox $\mathcal{T}$ over a signature Σ_P, the algorithm computeUnsat given in Figure 1 returns all unsatisfiable concepts and roles in Σ_P, by exploiting the transitive closure of the digraph representation of $\mathcal{T}$.

Before describing the algorithm, we recall that, given a digraph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ and a node $n \in \mathcal{N}$, the set predecessors$(n, \mathcal{G}^*)$ contains all those nodes n' in $\mathcal{N}$ such that $\mathcal{G}^*$ contains the arc (n', n), which means that there exists a path from n'

Algorithm: computeUnsat
Input: an OWL 2 QL TBox $\mathcal{T}$
Output: a set of concept and role expressions
foreach negative inclusion $S_1 \sqsubseteq \neg S_2 \in \mathcal{T}$ **do** /* step 1 */
  foreach $n_1 \in \mathsf{predecessors}(S_1, \mathcal{G}^*_\mathcal{T})$ **do**
    foreach $n_2 \in \mathsf{predecessors}(S_2, \mathcal{G}^*_\mathcal{T})$ **do**
      if $n_1 = n_2$
      then $\mathsf{Emp} \leftarrow \mathsf{Emp} \cup \{n_1\}$;
      if ($n_1 = \exists Q^-$ **and** $n_2 = A$) **or** ($n_2 = \exists Q^-$ **and** $n_1 = A$)
      then $\mathsf{Emp} \leftarrow \mathsf{Emp} \cup \{\exists Q.A\}$;
$\mathsf{Emp}' \leftarrow \emptyset$;
while $\mathsf{Emp} \neq \mathsf{Emp}'$ **do** /* step 2 */
  $\mathsf{Emp}' \leftarrow \mathsf{Emp}$;
  foreach $S \in \mathsf{Emp}'$ **do**
    foreach $n \in \mathsf{predecessors}(S, \mathcal{G}^*_\mathcal{T})$ **do**
      $\mathsf{Emp} \leftarrow \mathsf{Emp} \cup \{n\}$;
      if $n = P$ **or** $n = P^-$ **or** $n = \exists P$ **or** $n = \exists P^-$
      then $\mathsf{Emp} \leftarrow \mathsf{Emp} \cup \{P, P^-, \exists P, \exists P^-\}$;
      if there exists $B \sqsubseteq \exists Q.n \in \mathcal{T}$
      then $\mathsf{Emp} \leftarrow \mathsf{Emp} \cup \{\exists Q.n\}$;
return Emp.

Fig. 1. The algorithm $\mathsf{computeUnsat}(\mathcal{T})$

to n in $\mathcal{G}$. Also, it can be shown that $\mathcal{G}^*_\mathcal{T}$ allows in fact to obtain all subsumptions between satisfiable *basic* concepts or roles, in the sense that the TBox $\mathcal{T}$ infers one such subsumption $S_1 \sqsubseteq S_2$ if and only if there is an arc (S_1, S_2) in $\mathcal{E}^*$. Then, the two steps that compose the algorithm proceed as follows:

Step 1. Let S be either a concept expression or a role expression. We have that for each $S^i \in \mathsf{predecessors}(S, \mathcal{G}^*_\mathcal{T})$ the TBox $\mathcal{T}$ entails $S^i \sqsubseteq S$. Hence, given a negative inclusion assertion $S_1 \sqsubseteq \neg S_2$, for each $S_1^i \in \mathsf{predecessors}(S_1, \mathcal{G}^*_\mathcal{T})$ and for each $S_2^j \in \mathsf{predecessors}(S_2, \mathcal{G}^*_\mathcal{T})$, $\mathcal{T} \models S_1^i \sqsubseteq \neg S_2^j$. Therefore, for each negative inclusion $S_1 \sqsubseteq \neg S_2 \in \mathcal{T}$, the algorithm computes the set $\mathsf{predecessors}(S_1, \mathcal{G}^*_\mathcal{T})$ and $\mathsf{predecessors}(S_2, \mathcal{G}^*_\mathcal{T})$ and is able to: (i) recognize as unsatisfiable all those concepts and roles whose corresponding nodes occur in both the set $\mathsf{predecessors}(S_1, \mathcal{G}^*_\mathcal{T})$ and $\mathsf{predecessors}(S_2, \mathcal{G}^*_\mathcal{T})$, and (ii) identify those unsatisfiable qualified existential roles $\exists Q.A$ whose inverse existential role node $\exists Q^-$ occurs in $\mathsf{predecessors}(S_1, \mathcal{G}^*_\mathcal{T})$ (resp. $\mathsf{predecessors}(S_2, \mathcal{G}^*_\mathcal{T})$) and whose concept node A occurs in $\mathsf{predecessors}(S_2, \mathcal{G}^*_\mathcal{T})$ (resp. $\mathsf{predecessors}(S_1, \mathcal{G}^*_\mathcal{T})$), which indeed implies $\exists Q^- \sqsubseteq \neg A$ and therefore unsatisfiability of $\exists Q.A$.

Step 2. Further unsatisfiable concepts and roles are identified by the algorithm through a cycle in which: (i) if a concept or role S is in Emp, then all the expressions corresponding to the nodes in $\mathsf{predecessors}(S, \mathcal{G}^*_\mathcal{T})$ are in Emp. This captures propagation of unsatisfiability through chains of positive inclusions; (ii) if at least one of the expressions $P, P^-, \exists P, \exists P^-$ is in Emp, then all four expressions are in Emp; (iii) for each expression $\exists Q.A$ in $\mathcal{N}$, if $A \in \mathsf{Emp}$,

then $\exists Q.A \in \mathsf{Emp}$. We notice that the algorithm stops cycling when no new expressions of the form $\exists Q$ or $\exists Q.A$ are added to Emp (indeed, in this case only a single further iteration may be needed).

It easy to see that, by virtue of the fact that the size of the set $\mathcal{N}$ of the digraph representation of the TBox $\mathcal{T}$ is finite, $\mathsf{computeUnsat}(\mathcal{T})$ terminates, and that the number of executions of the while cycle is less than or equal to $|\mathcal{N}|$.

The following theorem shows that algorithm computeUnsat can be used for computing the set containing all the unsatisfiable concepts and roles in $\mathcal{T}$.

Theorem 3. *Let $\mathcal{T}$ be an OWL 2 QL TBox without axioms involving attributes and let S be either an atomic concept or an atomic role in Σ_P. $\mathcal{T} \models S \sqsubseteq \neg S$ if and only if $S \in \mathsf{computeUnsat}(\mathcal{T})$.*

As already said, it is easy to extend computeUnsat in such a way that it returns all unsatisfiable atomic concepts, atomic roles, and attributes occurring in general OWL 2 QL TBoxes. Therefore, we can restate Theorem 3 considering OWL 2 QL ontologies with also attributes and value-domain inclusions, and S that can be also an attribute. As an immediate consequence of this, we can compute the set $\Omega_{\mathcal{T}}$ of all "trivial" inclusion assertions inferred by an OWL 2 QL ontology $\mathcal{T}$, by means of the unsatisfiable predicates identified by computeUnsat. We call $\mathsf{Compute}\Omega$ the algorithm that, taken $\mathcal{T}$ as input, returns $\Omega_{\mathcal{T}}$ by making use of computeUnsat.

The following theorem, which is a direct consequence of Theorem 2 and of (the generalized version of) Theorem 3, states that our technique is sound and complete with respect to the problem of classifying an OWL 2 QL TBox.

Theorem 4. *Let $\mathcal{T}$ be an OWL 2 QL TBox and let S_1 and S_2 be either two atomic concepts, two atomic roles, or two attributes. $\mathcal{T} \models S_1 \sqsubseteq S_2$ if and only if $S_1 \sqsubseteq S_2 \in \mathsf{Compute}\Phi(\mathcal{T}) \cup \mathsf{Compute}\Omega(\mathcal{T})$.*

4 Implementation and Evaluation

By exploiting the results presented in Section 3, we have developed a Java-based OWL 2 QL classification module for the QuOnto reasoner [1,5,7].

This module computes the classification of an OWL 2 QL TBox $\mathcal{T}$ by adopting the technique described in Section 3. In this implementation the transitive closure of the digraph $\mathcal{G}_{\mathcal{T}}$ is based on a breadth first search through $\mathcal{G}_{\mathcal{T}}$.

We have performed comparative experiments, where QuOnto was tested against several popular ontology reasoners. Specifically, during our test we compared ourselves with the Fact++ [23], Hermit [9], and Pellet [22] OWL reasoners, and with the CB [14] Horn $\mathcal{SHIQ}$ reasoner, and with the ELK [15] reasoner for those ontologies that are also in OWL 2 EL.

The ontology suite used during testing includes twenty OWL ontologies, assembled from the TONES Ontology Repository[1] and from other independent sources. The six reasoners exhibited negligible differences in performance for the

[1] `http://owl.cs.manchester.ac.uk/repository/`

Table 1. In the table the Original and OWL 2 QL axioms fields indicate respectively the total number of axioms in the original version of the ontology and in the OWL 2 QL-approximated version. The Negative inclusion field reports the number of negative inclusions in the OWL 2 QL-approximated version.

Ontology	Concepts	Roles	Attributes	Original DL fragment	Original axioms	Owl 2 QL axioms	Negative inclusions
Mouse	2753	1	0	*ALE*	3463	3463	0
Transportation	445	89	4	*ALCH(D)*	931	931	317
DOLCE	209	313	4	*SHOIN(D)*	1736	1991	45
AEO	760	47	16	*SHIN(D)*	3449	3432	1957
Gene	26225	4	0	*SH*	42655	42655	3
EL-Galen	23136	950	0	*ELH*	46457	48026	0
Galen	23141	950	0	*ALEHIF+*	47407	49926	0
FMA 1.4	6488	165	0	*ALCOIF*	18612	18663	0
FMA 2.0	41648	148	20	*ALCOIF(D)*	123610	118181	0
FMA 3.2.1	84454	132	67	*ALCOIF(D)*	88204	84987	0
FMA-OBO	75139	2	0	*ALE*	119558	119558	0

majority of the smaller tested ontologies, so we will only discuss the ontologies which offered interesting results, meaning those on which reasoning times are significantly different for at least a subset of the reasoners.

These ontologies include: the Mouse ontology; the Transportation ontology[2]; the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [18]; the Athletic Events Ontology (AEO)[3]; the Gene Ontology (GO) [2]; two versions of the GALEN ontology [21]; and four versions of the Foundational Model of Anatomy Ontology (FMA) [10].

Because QuOnto is an OWL 2 QL reasoner, each benchmark ontology was preprocessed prior to classification in order to fit OWL 2 QL expressivity. Therefore, every OWL expression which cannot be expressed by OWL 2 QL axioms was approximated from the ontology specifications. This approximation follows this procedure: each axiom in the ontology is fed to an external reasoner, specifically Hermit, and every OWL 2 QL-compliant axiom that is implied from that axiom, between the ontology symbols that appear in it, is added to the OWL 2 QL-approximated ontology. For instance, the OWL assertion EquivalentClasses(ObjectUnionOf(:Male :Female) :Person) is approximated by the two assertions SubClassOf(:Male :Person) and SubClassOf(:Female :Person). Note that, as is the case in this example, the OWL 2 QL-approximated ontology may contain a greater number of axioms than the original ontology.

During the tests for each reasoner, classification was performed on the OWL 2 QL-compliant versions of the ontologies resulting from the above described preprocessing. Metrics about the ontologies are reported in Table 1.

All tests were performed on a DELL Latitude E6320 notebook with Intel Core i7-2640M 2.8Ghz CPU and 4GB of RAM, running Microsoft Windows 7 Premium operating system, and Java 1.6 with 2GB of heap space. Classification timeout was set at one hour, and aborting if maximum available memory was

[2] `http://www.daml.org/ontologies/409`

[3] `http://www.boemie.org/deliverable_d_3_5`

Table 2. Classification times of benchmark OWL 2 QL ontologies by QuOnto and other tested reasoners

Ontology	QuOnto	FaCT++	HermiT	Pellet	CB	ELK
Mouse	0.156	0.282	0.296	0.179	0.159	0.246
Transportation	0.150	0.045	0.163	0.151	0.195	0.343
DOLCE	1.327	0.245	25.619	1.696	1.358	—
AEO	0.650	0.743	0.920	0.647	0.605	—
Gene	1.255	1.400	3.810	2.803	1.918	1.419
EL-Galen	2.788	109.835	7.966	50.770	2.446	1.205
Galen	4.600	145.485	34.608	*timeout*	2.505	—
FMA 1.4	0.688	*timeout*	93.781	*timeout*	1.243	—
FMA 2.0	4.111	*out of memory*	*out of memory*	*timeout*	7.142	—
FMA 3.2.1	4.146	4.576	11.518	24.117	4.976	—
FMA-OBO	4.827	*timeout*	50.842	16.852	7.433	4.078

exhausted. All figures reported in Table 2 are in seconds, and, because classification results are subject to minor fluctuation, particularly when dealing with large ontologies, are the average of 3 classifications of the respective ontologies with each reasoner. The following versions of the OWL reasoners were tested: Fact++ v.1.5.3[4], HermiT v.1.3.6[5], Pellet v.2.3.0[6], CB v.12[7], and ELK v.0.3.2[8].

In our test configuration, the classifications of the FMA 2.0 ontology by the Hermit and FaCT++ reasoners terminate due to an out-of-memory error. In [9], classification of this ontology by the Hermit reasoner is performed successfully, but classification time far exceeds the one registered by QuOnto.

The results of the experiments are summarized in Table 2. These results confirm that the performance offered by QuOnto compares favorably to other reasoners for almost all tested ontologies. Classification for even the largest of the tested ontologies, i.e., the FMA-OBO and FMA 3.2.1 ontologies, is performed in under 5 seconds, and memory space issues were never experienced during our tests with QuOnto. For some test cases, the gap in performance between QuOnto and other reasoners is sizeable: for instance, classification by Pellet of the Galen and FMA (1.4 and 2.0) and by FaCT++ of the FMA (1.4 and OBO) ontologies exceeds the predetermined timeout limit of one hour.

Detailed analysis of the results provided in Table 2 shows that only the CB and ELK reasoners consistently display comparable performances to QuOnto, which is fastest for all ontologies which feature only positive inclusions, with the exception of the EL-Galen, Galen, and FMA-OBO ontologies. The CB reasoner, which is the best-performing reasoner for the Galen ontology, does not however always perform complete classification. For instance, it does not compute property hierarchies. The ELK reasoner instead is slower than QuOnto for three out of the five ontologies also in OWL 2 EL, showing instead markedly better performance for EL-Galen.

[4] `http://code.google.com/p/factplusplus/`
[5] `http://hermit-reasoner.com/`
[6] `http://clarkparsia.com/pellet`
[7] `http://code.google.com/p/cb-reasoner/`
[8] `http://code.google.com/p/elk-reasoner/`

Furthermore, if, as it is usually the case, an ontology does not present unsatisfiable predicates, the computation of such predicates through the exploration of all negative inclusions can be avoided. This is the case for ontologies such as DOLCE and AEO, for which computation of the set $\Phi_{\mathcal{T}}$ of positive inclusion assertions resulting from the transitive closure of $\mathcal{G}_{\mathcal{T}}$ is performed respectively in 0.347 and 0.384 seconds, fastest among tested reasoners. Instead, for ontologies such as Pizza and Transportation, which feature respectively 2 and 62 unsatisfiable atomic concepts, the identification of all such predicates is unavoidable, and the resulting set of trivial inclusion assertions must be added to $\Omega_{\mathcal{T}}$.

5 Conclusions

The research presented in this paper can be extended in various directions. First of all, in the implementation of our technique we have adopted a *naive* algorithm for computing the digraph transitive closure. We are currently experimenting more sophisticated and efficient techniques for this task. We are also working to optimize the procedure through which we identify unsatisfiable predicates. Finally, we are working to extend our technique to compute all inclusions that are inferred by the TBox (which, in OWL 2 QL, are a finite number). In this respect, we notice that through $\mathcal{G}^*_{\mathcal{T}}$ it is already possible to obtain the classification of all basic concepts, basic roles, and attributes, and not only that of predicates in the signature, and that, with slight modifications of computeUnsat, we can actually obtain the set of all negative inclusions inferred by an OWL 2 QL TBox. The remaining challenge is to devise an efficient mechanism to obtain all inferred positive inclusions involving qualified existential roles and attribute domains.

Acknowledgments. This research has been partially supported by the EU under FP7 project Optique – Scalable End-user Access to Big Data (grant n. FP7-318338).

References

1. Acciarri, A., Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Palmieri, M., Rosati, R.: QuOnto: Querying Ontologies. In: Veloso, M., Kambhampati, S. (eds.) Proc. of AAAI 2005, pp. 1670–1671. AAAI Press/The MIT Press (2005)
2. Ashburner, M., Ball, C., Blake, J., Botstein, D., Butler, H., Cherry, J., Davis, A., Dolinski, K., Dwight, S., Eppig, J., et al.: Gene Ontology: tool for the unification of biology. Nature Genetics 25(1), 25 (2000)
3. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation and Applications, 2nd edn. Cambridge University Press (2007)
4. Bang-Jensen, J., Gutin, G.Z.: Digraphs: Theory, Algorithms and Applications, 2nd edn. Springer (2008)
5. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., Rodriguez-Muro, M., Rosati, R., Ruzzi, M., Savo, D.F.: The MASTRO system for ontology-based data access. Semantic Web J. 2(1), 43–53 (2011)

6. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. J. of Automated Reasoning 39(3), 385–429 (2007)
7. De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., Rosati, R., Ruzzi, M., Savo, D.F.: MASTRO: A reasoner for effective ontology-based data access. In: Proc. of ORE 2012. CEUR, vol. 858 (2012), `ceur-ws.org`
8. Fagin, R., Kolaitis, P.G., Miller, R.J., Popa, L.: Data exchange: Semantics and query answering. In: Calvanese, D., Lenzerini, M., Motwani, R. (eds.) ICDT 2003. LNCS, vol. 2572, pp. 207–224. Springer, Heidelberg (2002)
9. Glimm, B., Horrocks, I., Motik, B., Shearer, R., Stoilos, G.: A novel approach to ontology classification. J. of Web Semantics 14, 84–101 (2012)
10. Golbreich, C., Zhang, S., Bodenreider, O.: The foundational model of anatomy in OWL: Experience and perspectives. J. of Web Semantics 4(3), 181–195 (2006)
11. Haarslev, V., Möller, R.: RACER system description. In: Goré, R., Leitsch, A., Nipkow, T. (eds.) IJCAR 2001. LNCS (LNAI), vol. 2083, pp. 701–706. Springer, Heidelberg (2001)
12. Ji, Q., Haase, P., Qi, G., Hitzler, P., Stadtmüller, S.: RaDON — repair and diagnosis in ontology networks. In: Aroyo, L., et al. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 863–867. Springer, Heidelberg (2009)
13. Kalyanpur, A., Parsia, B., Sirin, E., Hendler, J.A.: Debugging unsatisfiable classes in OWL ontologies. J. of Web Semantics 3(4), 268–293 (2005)
14. Kazakov, Y.: Consequence-driven reasoning for Horn $\mathcal{SHIQ}$ ontologies. In: Boutilier, C. (ed.) Proc. of IJCAI 2009, pp. 2040–2045. AAAI Press (2009)
15. Kazakov, Y., Krötzsch, M., Simančík, F.: Concurrent classification of $\mathcal{EL}$ ontologies. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 305–320. Springer, Heidelberg (2011)
16. Krötzsch, M.: The not-so-easy task of computing class subsumptions in OWL RL. In: Cudré-Mauroux, P., et al. (eds.) ISWC 2012, Part I. LNCS, vol. 7649, pp. 279–294. Springer, Heidelberg (2012)
17. Lawley, M., Bousquet, C.: Fast classification in Protégé: Snorocket as an OWL 2 EL reasoner. In: Meyer, T., Orgun, M., Taylor, K. (eds.) Proc. of AOW 2010. CRPIT, vol. 122, pp. 45–50. ACS (2010)
18. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A., Schneider, L.: The wonderweb library of foundational ontologies and the DOLCE ontology. Technical Report D17, WonderWeb (2002)
19. Mendez, J., Ecke, A., Turhan, A.: Implementing completion-based inferences for the $\mathcal{EL}$-family. In: Proc. of DL 2011. CEUR, vol. 745 (2011), `ceur-ws.org`
20. Motik, B., Grau, B.C., Horrocks, I., Wu, Z., Fokoue, A., Lutz, C.: OWL 2 Web Ontology Language – Profiles, 2nd edn. W3C Recommendation, World Wide Web Consortium (December 2012), `http://www.w3.org/TR/owl2-profiles/`
21. Rogers, J., Rector, A.: The GALEN ontology. In: Medical Informatics Europe, MIE 1996, pp. 174–178 (1996)
22. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoner. J. of Web Semantics 5(2), 51–53 (2007)
23. Tsarkov, D., Horrocks, I.: FaCT++ description logic reasoner: System description. In: Furbach, U., Shankar, N. (eds.) IJCAR 2006. LNCS (LNAI), vol. 4130, pp. 292–297. Springer, Heidelberg (2006)

RDFS with Attribute Equations via SPARQL Rewriting

Stefan Bischof[1,2] and Axel Polleres[1]

[1] Siemens AG Österreich, Siemensstraße 90, 1210 Vienna, Austria
[2] Vienna University of Technology, Favoritenstraße 9, 1040 Vienna, Austria

Abstract. In addition to taxonomic knowledge about concepts and properties typically expressible in languages such as RDFS and OWL, implicit information in an RDF graph may be likewise determined by arithmetic equations. The main use case here is exploiting knowledge about functional dependencies among numerical attributes expressible by means of such equations. While some of this knowledge can be encoded in rule extensions to ontology languages, we provide an arguably more flexible framework that treats attribute equations as first class citizens in the ontology language. The combination of ontological reasoning and attribute equations is realized by extending query rewriting techniques already successfully applied for ontology languages such as (the DL-Lite-fragment of) RDFS or OWL, respectively. We deploy this technique for rewriting SPARQL queries and discuss the feasibility of alternative implementations, such as rule-based approaches.

1 Introduction

A wide range of literature has discussed completion of data represented in RDF with implicit information through ontologies, mainly through taxonomic reasoning within a hierarchy of concepts (classes) and roles (properties) using RDFS and OWL. However, a lot of implicit knowledge within real world RDF data does not fall into this category: a large amount of emerging RDF data is composed of numerical attribute-value pairs assigned to resources which likewise contains a lot of implicit information, such as functional dependencies between numerical attributes expressible in the form of simple mathematical equations. These dependencies include unit conversions (e.g. between Fahrenheit and Celsius), or functional dependencies, such as the population density that can be computed from total population and area. Such numerical dependencies between datatype properties are not expressible in standard ontology languages such as RDFS or OWL. Rule based approaches also fail to encode such dependencies in the general case.

Example 1. Sample RDF data about cities, aggregated from sources such as DBPedia or Eurostat,[1] may contain data of various levels of completeness and using numerical attributes based on different units like

```
:Jakarta :tempHighC 33 .          :New_York :tempHighF 84 .
:New_York :population 8244910 .   :New_York :area_mile2 468.5 .
:Vienna :population 1714142 .     :Vienna :area_km2 414.6 .
:Vienna :populationDensity 4134 .    ...
```

[1] cf. `http://dbpedia.org/`, `http://eurostat.linked-statistics.org/`

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 335–350, 2013.

Users familiar with SPARQL might expect to be able to ask for the population density, or for places with temperatures over 90°F with queries like

```
SELECT ?C ?P WHERE { ?C :populationDensity ?P  }  or
SELECT ?C WHERE { ?C :tempHighF ?TempF FILTER(?TempF > 90) }
```

However, implicit answers from mathematical knowledge such as the following equations would not be returned by those queries:

$$tempHighC = (tempHighF - 32) \cdot {}^5\!/_9$$
$$populationDensity = population \div area_{km2}$$

One might ask why such equations cannot be directly added to the terminological knowledge modeled in ontologies? We aim to show that it actually can; further, we present an approach how to extend the inference machinery for SPARQL query answering under ontologies to cater for such equations. Inspired by query rewriting algorithms for query answering over DL-Lite [3], we show how similar ideas can be deployed to extend a DL-Lite fragment covering the core of RDFS with so-called equation axioms.

We focus on query rewriting techniques rather than e.g. rule-based approaches such as SWRL [13], where the equations from Example 1 could encoded as

$$tempHighC(X, C) \Leftarrow tempHighF(X, F), C = (F - 32) \cdot {}^5\!/_9 \quad (1)$$
$$populationDensity(X, PD) \Leftarrow population(X, P), area_{km2}(X, A), PD = P \div A \quad (2)$$

given respective arithmetic built-in support in a SWRL reasoner. However, note that these rules are not sufficient: (i) rule (1) is in the "wrong direction" for the query in Example 1, that is, we would need different variants of the rule for converting from $tempHighC$ to $tempHighF$ and vice versa; (ii) the above rules are not *DL safe* (i.e., we do not suffice to bind values only to explicitly named individuals, as we want to compute *new* values) which potentially leads to termination problems in rule-based approaches (and as we will see it actually does in existing systems). Our approach addresses both these points in that (i) equations are added as first class citizens to the ontology language, where variants are considered directly in the semantics, (ii) the presented query rewriting algorithm always terminates and returns finite answers; we also discuss reasonable completeness criteria.

In the remainder of this paper, we first define our ontology language DL^E_{RDFS} which extends the RDFS fragment of DL-Lite by simple equations (Sect. 2). In Sect. 3 we define SPARQL queries over DL^E_{RDFS} and present our query rewriting algorithm along with a discussion of considerations on soundness and completeness. Alternative implementation approaches with DL reasoners and rules are discussed briefly in Sect. 4, followed by the discussion of a use case experiment in Sect. 5. We wrap up with a discussion of related and future work as well as conclusions (Sects. 6 and 7).

2 Extending Description Logics by Equations

We herein define a simple, restricted form of arithmetic equations and extend a lightweight fragment of DL-Lite by such equations.

Definition 1. *Let $\{x_1, \ldots, x_n\}$ be a set of variables. A* simple equation *E is an algebraic equation of the form $x_1 = f(x_2, \ldots, x_n)$ such that $f(x_2, \ldots, x_n)$ is an arithmetic*

expression over numerical constants and variables $x_2, \ldots, x_n$ where f uses the elementary algebraic operators $+, -, \cdot, \div$ and contains each x_i exactly once. $\mathrm{vars}(E)$ *is the set of variables $\{x_1, \ldots, x_n\}$ appearing in E.*

That is, we allow non-polynomials for f – since divisions are permitted – but do not allow exponents (different from ± 1) for any variable; such equations can be solved uniquely for each x_i by only applying elementary transformations, assuming that all x_j for $j \neq i$ are known: i.e., for each x_i, such that $2 \leq i \leq n$, an equivalent equation E' of the form $x_i = f'(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$ is uniquely determined. Note that since each variable occurs only once, the standard procedure for solving single variable equations can be used, we write $\mathrm{solve}(x_1 = f(x_2, \ldots, x_n), x_i)$ to denote E'.[2]

2.1 The Description Logic $\mathbf{DL}^{\boldsymbol{E}}_{\boldsymbol{RDFS}}$

When we talk about Description Logics (DL), we consider a fragment of DL-Lite$_{\mathcal{A}}$ [18] with basic concepts, existential quantification, attributes over concrete value domains, role/attribute inclusions, and inverse roles which we extend by simple attribute equations. We we call this fragment DL^{E}_{RDFS}, i.e., it is just expressive enough to capture (the DL fragment of) the RDFS semantics [11] extended with equations. In contrast to DL-Lite$_{\mathcal{A}}$, DL^{E}_{RDFS} leaves out role functionality, as well as concept and role negation, and we restrict ourselves to a single value domain for attributes, the set of rational numbers $\mathbb{Q}$.[3]

Definition 2. *Let A be an atomic concept name, P be an atomic role name, and U be an atomic attribute name. As usual, we assume the sets of atomic concept names, role name, and attribute names to be disjoint. Then DL* concept expressions *are defined as* $C ::= A \mid \exists P \mid \exists P^- \mid \exists U$

In the following, let Γ be an infinite set of constant symbols (which, in the context of RDF(S) essentially equates to the set I of IRIs).

Definition 3. *A DL^{E}_{RDFS} knowledge base (KB) $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ consists of a finite set of terminological axioms $\mathcal{T}$ (TBox) and assertions $\mathcal{A}$ (ABox). For A, P_i, U_i and C denoting atomic concepts, roles, attributes, and concept expressions, resp., $\mathcal{T}$ can contain:*

$$
\begin{aligned}
C &\sqsubseteq A && (\textit{concept inclusion axiom})\\
P_1 &\sqsubseteq P_2 && (\textit{role inclusion axiom})\\
U_1 &\sqsubseteq U_2 && (\textit{attribute inclusion axiom})\\
U_0 &= f(U_1, \ldots, U_n) && (\textit{equation axiom})
\end{aligned}
$$

A set of role (attribute, resp.) inclusion axioms is called a role hierarchy *(*attribute hierarchy*, resp.). For $a, b \in \Gamma$, and $q \in \mathbb{Q}$, an* ABox *is a set of* concept assertions $C(a)$, role assertions $R(a, b)$, *and* attribute assertions $U(a, q)$. *Finally, by $\Gamma_{\mathcal{K}}$ (and $\Gamma_A, \Gamma_P, \Gamma_U$, resp.), we denote the (finite) sets of constants from Γ appearing in $\mathcal{K}$ (as concepts, roles, and attributes, resp.).*

[2] In analogy to notation used by computer algebra systems (such as Mathematica or Maxima).

[3] Note that since we only consider this single type of attributes, we also do not introduce value-domain expressions from [18]. Further, instead of $\delta(U)$ in [18] we just write $\exists U$.

Rows 1–6 of Table 1 show the obvious correspondence between DL^{E}_{RDFS} syntax and the essential RDFS terminological vocabulary. As for line 7, we can encode equation axioms in RDF by means of a new property definedByEquation and write the respective arithmetic expressions $f(U_1, \ldots, U_n)$ as plain literals (instead of e.g. breaking down the arithmetic expressions into RDF triples). ABox assertions are covered in rows 8–10, where we note that we use datatype literals of the type owl:rational from OWL2 for rational numbers (which however subsumes datatypes such as xsd:integer, xsd:decimal more commonly used in real world RDF data).

As mentioned before in the context of Definition 1, we consider equations that result from just applying elementary transformations as equivalent. In order to define the semantics of equation axioms accordingly, we will make use of the following definition.

Definition 4. *Let* $E\colon U_0 = f(U_1, \ldots, U_n)$ *be an equation axiom then, for any* U_i *with* $0 \leq i \leq n$ *we call the equation axiom* $\mathrm{solve}(E, U_i)$ *the* U_i-variant *of* E.

Note that the DL defined herein encompasses the basic expressivity of RDFS (subproperty, subclassOf, domain, range)[4] and in fact, rather than talking about a restriction of DL-Lite$_{\mathcal{A}}$, we could also talk about an extension of DL-Lite$_{RDFS}$ [1].[5]

Definition 5 (Interpretation). *An* interpretation $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ *consists of a non-empty set* $\Delta^{\mathcal{I}}$ *called the object domain, and an interpretation function* $\cdot^{\mathcal{I}}$ *which maps*

- *each atomic concept* A *to a subset of the domain* $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$,
- *each atomic role* P *to a binary relation over the domain* $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$,
- *each attribute* U *to a binary relation over the domain and the rational numbers* $U^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \mathbb{Q}$, *and*
- *each element of* Γ *to an element of* $\Delta^{\mathcal{I}}$.

For concept descriptions the interpretation function is defined as follows:

Table 1. DL^{E}_{RDFS} axioms in RDFS

	DL^{E}_{RDFS}	RDFS
1	$A_1 \sqsubseteq A_2$	$\mathsf{A_1}$ rdfs:subClassOf $\mathsf{A_2}$
2	$\exists P \sqsubseteq A$	P rdfs:domain A
3	$\exists P^- \sqsubseteq A$	P rdfs:range A
4	$\exists U \sqsubseteq A$	U rdfs:domain A
5	$P_1 \sqsubseteq P_2$	$\mathsf{P_1}$ rdfs:subPropertyOf $\mathsf{P_2}$
6	$U_1 \sqsubseteq U_2$	$\mathsf{U_1}$ rdfs:subPropertyOf $\mathsf{U_2}$
7	$U_0 = f(U_1, \ldots, U_n)$	$\mathsf{U_0}$ definedByEquation "$\mathsf{f(U_1, \ldots, U_n)}$"
8	$A(x)$	x rdf:type A
9	$R(x, y)$	x R y
10	$U(x, q)$	x U "q"^^owl:rational

[4] Leaving out subtleties such as e.g. those arising from non-standard use [2] of the RDF vocabulary.

[5] DL-Lite$_{RDFS}$ actually also allows to write axioms of the form $P_1 \sqsubseteq P_2^-$ which we do not allow since these in fact are beyond the basic expressivity of RDFS.

- $(\exists R)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \exists y.(x,y) \in R^{\mathcal{I}}\}$
- $(\exists R^{-})^{\mathcal{I}} = \{y \in \Delta^{\mathcal{I}} \mid \exists x.(x,y) \in R^{\mathcal{I}}\}$
- $(\exists U)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \exists q \in \mathbb{Q}.(x,q) \in U^{\mathcal{I}}\}$

Definition 6 (Model). *An interpretation $\mathcal{I}$ satisfies an axiom of the form*

- $C \sqsubseteq A$ *if* $C^{\mathcal{I}} \subseteq A^{\mathcal{I}}$
- $P_1 \sqsubseteq P_2$ *if* $P_1^{\mathcal{I}} \subseteq P_2^{\mathcal{I}}$
- $U_1 \sqsubseteq U_2$ *if* $U_1^{\mathcal{I}} \subseteq U_2^{\mathcal{I}}$
- $U_0 = f(U_1, \ldots, U_n)$ *if*
  $\forall x, y_1, \ldots, y_n(\bigwedge_{i=1}^{n}(x, y_i) \in U_i^{\mathcal{I}}) \wedge \mathrm{defined}(f(U_1/y_1, \ldots, U_n/y_n)$
  $\Rightarrow (x, \mathrm{eval}(f(U_1/y_1, \ldots, U_n/y_n)) \in U_0^{\mathcal{I}}$

where, by $\mathrm{eval}(f(U_1/y_1, \ldots, U_n/y_n))$ *we denote the actual value in* $\mathbb{Q}$ *from evaluating the arithmetic expression* $f(U_1, \ldots, U_n)$ *after substituting each* U_i *with* y_i*, and by* $\mathrm{defined}(f(U_1/y_1, \ldots, U_n/y_n))$ *we denote that this value is actually defined (i.e., does not contain a division by zero). Analogously,* $\mathcal{I}$ *satisfies an ABox assertion of the form*

- $C(a)$ *if* $a^{\mathcal{I}} \in C^{\mathcal{I}}$
- $P(a,b)$ *if* $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in P^{\mathcal{I}}$
- $U(a,q)$ *if* $(a^{\mathcal{I}}, q) \in U^{\mathcal{I}}$

Finally, an interpretation $\mathcal{I}$ *is called a* model *of a KB* $\mathcal{K} = (\mathcal{T}, \mathcal{A})$*, written* $\mathcal{I} \models \mathcal{K}$*, if* $\mathcal{I}$ *satisfies all (role, attribute and concept) inclusion axioms in* $\mathcal{T}$*, all* variants *of equation axioms in* $\mathcal{T}$*, and all assertions in* $\mathcal{A}$*.*

Finally, we define conjunctive queries (with assignments) over DL_{RDFS}^{E}.

Definition 7. *A* conjunctive query *(CQ) is an expression of the form*

$$q(\boldsymbol{x}) \leftarrow \exists \boldsymbol{y}.\phi(\boldsymbol{x}, \boldsymbol{y})$$

where $\boldsymbol{x}$ *is a sequence of variables called* distinguished variables*,* $\boldsymbol{y}$ *is a sequence of variables called* non-distinguished variables*, and* ϕ *is a conjunction of* ***class***, ***role*** *or* ***attribute atoms*** *of the forms* $C(x), P(x,y)$*, and* $U(x,z)$*, respectively, and* ***assignments*** *of the form* $x_0 = f(x_1, \ldots, x_n)$ *representing simple equations, where* x, y *are constant symbols from* Γ *or variables (distinguished or non-distinguished), and* z *is either a value from* $\mathbb{Q}$ *or a variable, and the* x_i *are variables such that for all* $i \geq 1$*,* x_i *appears in an atom of the form* $U(x, x_i)$ *within* ϕ*. A set of queries with the same head* $q(\boldsymbol{x})$ *is a* union of conjunctive queries *(UCQ).*

For an interpretation $\mathcal{I}$, we denote by $q^{\mathcal{I}}$ the set of tuples $\boldsymbol{a}$ of domain elements and elements of $\mathbb{Q}$ which makes ϕ true[6] when $\boldsymbol{a}$ is assigned to distinguished variables $\boldsymbol{x}$ in q.

Definition 8. *For a conjunctive query* q *and a KB* $\mathcal{K}$ *the* answer to q over $\mathcal{K}$ *is the set* $\mathrm{ans}(q, \mathcal{K})$ *consisting of tuples* $\boldsymbol{a}$ *of constants from* $\Gamma_{\mathcal{K}} \cup \mathbb{Q}$ *such that* $\boldsymbol{a}^{\mathcal{M}} \in q^{\mathcal{M}}$ *for every model* $\mathcal{M}$ *of the KB* $\mathcal{K}$*.*

[6] We mean true in the sense of first-order logic, where we assume that the interpretation of arithmetic expressions is built-in with the usual semantics for arithmetics over the rational numbers $\mathbb{Q}$, and that equality "=" is false for expressions that yield division by zero on the RHS.

Note that, as opposed to most DL-Lite variants (such as [3]), $\mathrm{ans}(q, \mathcal{K})$ in our setting is not necessarily finite, as shown by the following example.

Example 2. Let $\mathcal{K}_1 = (\mathcal{T}_1, \mathcal{A}_1)$ with $\mathcal{A}_1 = u_1(o_1, 1), u_2(o_1, 1), u_3(o_1, 1), \mathcal{T}_1 = \{e\colon u_1 = u_2 + u_3\}$ and $q(x) \leftarrow u_1(o_1, x)$ then $\mathrm{ans}(q, \mathcal{K})$ contains any value from $\mathbb{N}$.

3 SPARQL over $\mathbf{DL}^{E}_{RDFS}$

The semantics of SPARQL is defined as usual based on matching of basic graph patterns (BGPs), more complex patterns are defined as per the usual SPARQL algebra and evaluated on top of basic graph pattern matching, cf. for instance [16, 19]. In order to remain compatible with the notion of CQs in DL^{E}_{RDFS}, we only allow restricted BGPs.[7]

Definition 9. *Let V be an infinite set of variables, I be the set of IRIs, I_{RDF} = {rdfs:subClassOf, rdfs:subPropertyOf, rdfs:domain, rdfs:range, rdf:type, definedByEquation}, and $I' = I \setminus I_{RDF}$, then* basic graph patterns (BGPs) *are sets of RDF triple patterns (s, p, o) from $((I' \cup V) \times I' \times (I' \cup \mathbb{Q} \cup V)) \cup ((I' \cup V) \times \{\mathsf{rdf:type}\} \times I')$*

More complex graph patterns can be defined recursively on top of basic graph patterns, i.e., if P_1 and P_2 are graph patterns, $v \in V$, $g \in I \cup V$, R is a filter expression, and $Expr$ an arithmetic expression over constants and variables in V, then (i) $\{\{P_1\}\{P_2\}\}$ (conjunction), (ii) $\{P_1\}$ UNION $\{P_2\}$ (disjunction), (iii) P_1 OPTIONAL $\{P_2\}$ (left-outer join), (iv) P_1 FILTER(R) (filter), and (v) P_1 BIND $(Expr$ AS $v)$ (assignment) are graph patterns; as a syntactic restriction we assume that $v \notin \mathrm{vars}(P_1)$. The evaluation semantics of complex patterns builds up on basic graph pattern matching,[8] which we define in our setting simply in terms of conjunctive query answering over the underlying DL.

Following the correspondence of Table 1 and the restrictions we have imposed on BGPs, any BGP P can trivially be mapped to a (non-distinguished-variable-free) conjunctive query of the form $q_P\colon q(\mathrm{vars}(P)) \leftarrow \phi(P)$, where $\mathrm{vars}(P)$ is the set of variables occurring in P.

Example 3. Within the SPARQL query

```
SELECT ?X WHERE { { :o1 :u1 ?X } FILTER ( ?X > 1 ) }
```

the BGP `{ :o1 :u1 ?X }` corresponds to the CQ from Example 2. FILTERs and other complex patterns are evaluated on top of BGP matching:

Definition 10 (Basic graph pattern matching for $\mathbf{DL}^{E}_{RDFS}$). *Let G be an RDF representation of a DL^{E}_{RDFS} KB (cf. Table 1) $\mathcal{K}$. Then, the solutions of a BGP P for G, denoted (analogously to [16]) as $[\![P]\!]_G = \mathrm{ans}(q_P, \mathcal{K})$.*

[7] We note though, that soundness of our query rewriting approach would not be affected if we allowed arbitrary BGPs.

[8] For simplicity we leave our GRAPH graph patterns or other new features except BIND introduced in SPARQL1.1.

Note that here we slightly abused notation using $\mathrm{ans}(q_P, \mathcal{K})$ synonymous for what would be more precisely "the set of SPARQL variable mappings corresponding to $\mathrm{ans}(q_P, \mathcal{K})$". As for the semantics of more complex patterns, we refer the reader to [16, 19] for details, except for the semantics of BIND which is newly introduced in SPARQL 1.1 [10], which we define as:

$$[\![P \,\mathsf{BIND}\, (\mathit{Expr}\, \mathsf{AS}\, v)]\!]_G = \{\mu \cup \{v \to \mathrm{eval}(\mu(\mathit{Expr}))\} \mid \mu \in [\![P]\!]_G\}$$

Here, by $\mathrm{eval}(\mu(\mathit{Expr}))$ we denote the actual value in $\mathbb{Q}$ from evaluating the arithmetic expression Expr after applying the substitutions from μ.

3.1 Adapting PerfectRef to $\mathrm{DL}^{\boldsymbol{E}}_{\boldsymbol{RDFS}}$

Next, we extend the PerfectRef algorithm [3] which reformulates a conjunctive query to directly encode needed TBox assertions in the query. The algorithm $\mathsf{PerfectRef}_\mathsf{E}$ in Algorithm 1 extends the original PerfectRef by equation axioms and conjunctive queries containing assignments, as defined before, following the idea of query rewriting

Algorithm 1. Rewriting algorithm $\mathsf{PerfectRef}_\mathsf{E}$

Input: Conjunctive query q, TBox $\mathcal{T}$
Output: Union (set) of conjunctive queries

$P := \{q\}$
repeat
  $P' := P$
  foreach $q \in P'$ **do**
    foreach *g in q* **do** // expansion
      foreach *inclusion axiom I in* $\mathcal{T}$ **do**
        if *I is applicable to g* **then**
          $P := P \cup \{q[g/\mathrm{gr}(g, I)]\}$
      foreach *equation axiom E in* $\mathcal{T}$ **do**
        if $g = U^{\mathrm{adn}(g)}(x, y)$ *is an (adorned) attribute atom and* $\mathrm{vars}(E) \cap \mathrm{adn}(g) = \emptyset$ **then**
          $P := P \cup \{q[g/\mathrm{expand}(g, E)]\}$
until $P' = P$
return P

Table 2. Semantics of $\mathrm{gr}(g, I)$ of Algorithm 1

g	I	$\mathrm{gr}(g/I)$
$A(x)$	$B \sqsubseteq A$	$B(x)$
	$\exists P \sqsubseteq A$	$P(x, \_)$
	$\exists P^- \sqsubseteq A$	$P(\_, x)$
	$\exists U \sqsubseteq A$	$U(x, \_)$
$P_1(x, y)$	$P_2 \sqsubseteq P_1$	$P_2(x, y)$
$U_1^{\mathrm{adn}(g)}(x, y)$	$U_2 \sqsubseteq U_1$	$U_2^{\mathrm{adn}(g)}(x, y)$

by "expanding" a conjunctive query (CQ) Q to a union of conjunctive queries (UCQ) Q_0 that is translated to a regular SPARQL 1.1 query which is executed over an RDF Store.

$\mathsf{PerfectRef_E}$ first expands atoms using inclusion axioms (lines 6–8) as in the original PerfectRef algorithm. Here, an DL^{E}_{RDFS} inclusion axiom I is *applicable* to a query atom g if the function gr (Table 2) is defined.[9] The only new thing compared to [3] in Table 2 is the "adornment" $\mathrm{adn}(g)$ of attribute atoms which we explain next, when turning to the expansion of equation axioms.

The actually new part of $\mathsf{PerfectRef_E}$ that reformulates attribute atoms in terms of equation axioms is in lines 9–11. In order to avoid infinite expansion of equation axioms during the rewriting, the algorithm "adorns" attribute atoms in a conjunctive query by a set of attribute names. That is, given an attribute atom $U(x,z)$ and a set of attribute names $\{U_1,\ldots,U_k\}$ we call $g = U^{U_1,\ldots,U_k}(x,z)$ an *adorned attribute atom* and write $\mathrm{adn}(g) = \{U_1,\ldots,U_k\}$ to denote the set of adornments. For an unadorned $g = U(x,z)$, obviously $\mathrm{adn}(g) = \emptyset$. Accordingly, we call an *adorned conjunctive query* a CQ where adorned attribute atoms are allowed.

The function $\mathrm{expand}(g,E)$ returns for $g = U^{\mathrm{adn}(g)}(x,y)$ and $E' : U = f(U_1,\ldots,U_n)$ being the U-variant of E the following conjunction:

$$U_1^{\mathrm{adn}(g)\cup\{U\}}(x,y_1) \wedge \ldots \wedge U_n^{\mathrm{adn}(g)\cup\{U\}}(x,y_n) \wedge y = f(y_1,\ldots,y_n)$$

where $y_1,\ldots,y_n$ are fresh variables. Here, the condition $\mathrm{vars}(E)\cap\mathrm{adn}(g) = \emptyset$ ensures that U is not "re-used" during expansion to compute its own value recursively. The adornment thus prohibits infinite recursion.

We note that we leave out the *reduction* step of the original PerfectRef algorithm from [3][Fig.2, step (b)], since it does not lead to any additional applicability of inclusion axioms in the restricted Description Logic DL^{E}_{RDFS}. As we may extend $\mathsf{PerfectRef_E}$ to more expressive DLs as part of future work, this step may need to be re-introduced accordingly.

Finally, just as before we have defined how to translate a SPARQL BGP P to a conjunctive query, we translate the result of $\mathsf{PerfectRef_E}(q_P,\mathcal{T})$ back to SPARQL by means of a recursive translation function $\mathrm{tr}(\mathsf{PerfectRef_E}(q_P,\mathcal{T}))$. That is, for $\mathsf{PerfectRef_E}(q_P,\mathcal{T}) = \{q_1,\ldots q_m\}$ and each q_i being of the form $\bigwedge_{j=0}^{k_i} atom_j$, we define tr as follows:

$\mathrm{tr}(\{q_1,\ldots q_m\})$	{ $\mathrm{tr}(q_1)$ } UNION ...UNION { $\mathrm{tr}(q_m)$ }
$\mathrm{tr}(\bigwedge_j = 0^{k_i} atom_j)$	$\mathrm{tr}(atom_1)$ ....$\mathrm{tr}(atom_{k_i})$
$\mathrm{tr}(A(x))$	$\mathrm{tr}(x)$ rdf : type A
$\mathrm{tr}(P(x,y))$	$\mathrm{tr}(x)$ P $\mathrm{tr}(y)$
$\mathrm{tr}(U(x,y))$	$\mathrm{tr}(x)$ U $\mathrm{tr}(y)$
$\mathrm{tr}(y = f(y_1,\ldots,y_n))$	BIND(f(tr(y_1),...,tr(y_n)) AS $\mathrm{tr}(y)$)
$\mathrm{tr}(x)$, for $x \in V$	$?x$
$\mathrm{tr}(x)$, for $x \in \Gamma$	x
$\mathrm{tr}(x)$, for $x \in \mathbb{Q}$	"x"^^owl:rational

[9] With DL^{E}_{RDFS} we cover only a very weak DL, but we expect that our extension is applicable to more complex DLs such as the one mentioned in [3], which we leave for future work.

The following proposition follows from the results in [3], since (a) $\mathsf{PerfectRef}_\mathsf{E}$ is a restriction of the original PerfectRef algorithm as long as no equation axioms are allowed, and (b) any DL^E_{RDFS} KB is consistent.

Proposition 1. *Let q be a conjunctive query without assignments and $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL^E_{RDFS} KB without equation axioms. Then $\mathsf{PerfectRef}_\mathsf{E}$ is sound and complete, i.e.*

$$\mathrm{ans}(q, \mathcal{K}) = \mathrm{ans}(\mathsf{PerfectRef}_\mathsf{E}(q, \mathcal{T}), \langle \emptyset, \mathcal{A} \rangle)$$

The following corollary follows similarly.

Corollary 1. *Let q be a conjunctive query without assignments and without attribute axioms and let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be an arbitrary DL^E_{RDFS} KB. Then $\mathsf{PerfectRef}_\mathsf{E}$ is sound and complete.*

As for arbitrary DL^E_{RDFS} knowledge bases, let us return to Example 2.

Example 4. Given the knowledge base $\mathcal{K}_1 = \langle \mathcal{T}_1, \mathcal{A}_1 \rangle$ and query q from Example 2. The query $\mathsf{PerfectRef}_\mathsf{E}(q, \mathcal{T})$ is

$$\{\, q(x) \leftarrow u_1(o_1, x), q(x) \leftarrow u_2^{u_1}(o_1, x_2), u_3^{u_1}(o_1, x_3), x = x_2 + x_3 \}$$

which only has the certain answers $x = 1$ and $x = 2$, showing that $\mathsf{PerfectRef}_\mathsf{E}$ is incomplete in general. As a variant of $\mathcal{K}_1$, lets consider $\mathcal{K}_2 = \langle \mathcal{T}_1, \mathcal{A}_2 \rangle$ with the modified ABox $\mathcal{A}_2 = \{u_1(o_1, 2), u_2(o_1, 1), u_3(o_1, 1)\}$. In this case, notably $\mathsf{PerfectRef}_\mathsf{E}$ delivers complete results for $\mathcal{K}_2$, i.e., $\mathrm{ans}(q, \mathcal{K}_2) = \mathrm{ans}(\mathsf{PerfectRef}_\mathsf{E}(q, \mathcal{T}_1), \langle \emptyset, \mathcal{A}_2 \rangle)$ with the single certain answer $x = 2$. Finally, the rewritten version of the SPARQL query in Example 3 is

```
SELECT ?X WHERE {
    { { :o1 :u1 ?X } UNION
      { :o1 :u2 ?X2 . :o1 :u3 ?X3 . BIND(?X2+?X3 AS ?X ) } }
    FILTER ( ?X > 1 )  }
```

In order to capture a class of DL^E_{RDFS} KBs, where completeness can be retained, we will use the following definition.

Definition 11. *An ABox $\mathcal{A}$ is* data-coherent *with $\mathcal{T}$, if there is no pair of ground atoms $U(x, d'), U(x, d)$ with $d \neq d'$ entailed by $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$*

The following result is obvious.

Lemma 1. *Whenever $\mathcal{A}$ is* data-coherent *with $\mathcal{T}$, any conjunctive query has a finite number of certain answers.*

Proof (Sketch). Assume that the certain answers to q are infinite. From Corollary 1 we can conclude that infiniteness can only stem from distinguished variables that occur as attribute value y in some attribute atom $U(x, y)$ in the query. However, that would in turn mean that there is at least one x with an infinite set of findings for y, which contradicts the assumption of data-coherence.

The following stronger result (which for our particular use case of BGP matching in SPARQL we only consider for non-distinguished-variable-free conjunctive queries) states that data-coherence in fact implies completeness.

Theorem 1. *If $\mathcal{A}$ is* data-coherent *with $\mathcal{T}$, the for any non-distinguished-variable-free conjunctive query q* $\mathsf{PerfectRef}_\mathsf{E}$ *is sound and complete.*

Proof (Sketch). The idea here is that whenever $\mathcal{A}$ is *data-coherent* with $\mathcal{T}$ for any fixed x any certain value y for $U(x, y)$ will be returned by $\mathsf{PerfectRef}_\mathsf{E}$: assuming the contrary, following a shortest derivation chain $U(x, y)$ can be either (i) be derived by only atoms $U_i(x, y_i)$ such that any U_i is different from U, in which case this chain would have been "expanded" by $\mathsf{PerfectRef}_\mathsf{E}$, or (ii) by a derivation chain that involves an instance of $U(x, z)$. Assuming now that $z \neq y$ would violate the assumption of data-coherence, whereas if $z = y$ then $U(x, y)$ was already proven by a shorter derivation chain.

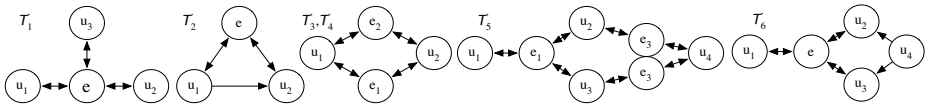
In what follows, we will define a fragment of DL^E_{RDFS} KBs where data-coherence can be checked efficiently. First, we note that a data-coherent ABox alone, such as for instance in $\mathcal{K}_2$ in Example 4 above, is in general not a guarantee for data-coherence. To show this, let us consider the following additional example.

Example 5. Consider the TBox $\mathcal{T}_2 = \{e\colon u_1 = u_2 + 1, u_2 \sqsubseteq u_1\}$ As easily can be seen, any ABox containing an attribute assertion for either u_1 or u_2 is data-incoherent with $\mathcal{T}_2$.

The example also shows that considering equation axioms only is not sufficient to decide data-coherence, but we also need to consider attribute inclusion axioms. Following this intuition, we define a dependency graph over $\mathcal{T}$ as follows.

Definition 12. *A TBox dependency graph is $G_\mathcal{T} = \langle N, E\rangle$ is constructed from nodes for each attribute and each equation axiom $N = \{e \mid e \text{ is an equation axiom in } \mathcal{T}\} \cup \Gamma_U$. There exist edges (e, v) and (v, e) between every equation e and its variables $v \in \mathrm{vars}(e)$. Furthermore there exists an edge (u, v) for each attribute inclusion axiom $u \sqsubseteq v$. If G contains no (simple) cycle with length greater than 2, then we call $\mathcal{T}$* attribute-acyclic.

Example 6. Given $\mathcal{T}_1, \mathcal{T}_2$ from Examples 2 and 5, let further $\mathcal{T}_3 = \{e_1\colon u_1 = u_2 + 1, e_2\colon u_2 = u_1 + 1\}$, $\mathcal{T}_4 = \{e_1\colon u_1 = u_2 + 1, e_2\colon u_2 = u_1 - 1\}$, and $\mathcal{T}_5 = \{e_1\colon u_1 = u_2 - u_3, e_2\colon u_4 = u_2\, e_3\colon u_4 = u_3\}$ $\mathcal{T}_6 = \{e\colon u_1 = u_2 - u_3, u_4 \sqsubseteq u_2\, u_4 \sqsubseteq u_3\}$ then the resp. dependency graphs are as follows where the graphs for $\mathcal{T}_2$–$\mathcal{T}_5$ are cyclic.



Notably, since e_2 is a variant of e_1 in $\mathcal{T}_4$, $\mathcal{T}_4$ is actually equivalent to an acyclic TBox (removing either e_1 or e_2), whereas this is not the case for $\mathcal{T}_3$; more refined notions of acyclicity, which we leave for future work, might capture this difference. Therefore, as shown in Examples 2 and 4 for $\mathcal{T}_1, \mathcal{T}_2$. Further, let us point out the subtle difference

between $\mathcal{T}_5$ and $\mathcal{T}_6$. In $\mathcal{T}_5$, when e_1–e_3 are viewed as equation system, partially solving this system would result in the new equation $u_1 = 0$, independent of the ABox. Since $\mathsf{PerfectRef_E}$ does not solve any equation systems (but only instantiates equations with values from the ABox), it would not detect this. On the contrary, in $\mathcal{T}_6$, only when a concrete "witness" for u_4 is available in the ABox, this constrains the value of u_1 to be 0, which could be correctly detected by means of $\mathsf{PerfectRef_E}$: for attribute-acyclic TBoxes, data-coherence indeed (finitely) depends on the ABox and we can define a procedure to check data-coherence (and thus completeness) by means of $\mathsf{PerfectRef_E}$ itself.

Proposition 2. *Let $\mathcal{T}$ be an attribute-acyclic TBox, and $\Gamma_U = \{u_1, \ldots, u_m\}$. The following SPARQL query $Q^{\mathcal{T}}_{check}$*

$$\begin{aligned}&\mathsf{ASK}\ \{\ \{\ \mathrm{tr}(\mathsf{PerfectRef_E}(q_{P_1}, \mathcal{T}))\ \mathsf{FILTER}(\ ?Y_1\ !{=}\ ?Z_1)\ \}\\ &\qquad \mathsf{UNION} \ldots\ \mathsf{UNION}\\ &\qquad \{\ \mathrm{tr}(\mathsf{PerfectRef_E}(q_{P_m}, \mathcal{T}))\ \mathsf{FILTER}(\ ?Y_1\ !{=}\ ?Z_1)\ \}\ \}\end{aligned}$$

where $P_i = \{\ ?X\ u_i\ ?Y_1\ .\ ?X\ u_i\ ?Z_2\ \}$ determines data-coherence in the following sense: an ABox $\mathcal{A}$ is data-coherent with $\mathcal{T}$ if Q returns "no".

The idea here is that since $\mathcal{T}$ is attribute-acyclic, and due to the restriction that variable occurs at most once in simple equations, finite witnesses for data-incoherences can be acyclically derived from the ABox, and thus would be revealed by $\mathsf{PerfectRef_E}$.

4 Discussion of Alternative Implementation Approaches

Our approach relies on standard SPARQL1.1 queries and runs on top of any off-the-shelf SPARQL1.1 implementation by first extracting the TBox and then rewriting BGPs in each query according to the method described in the previous section. In order to compare this rewriting to alternative approaches, we have looked into DL reasoners as well as rule-based reasoners, namely, Racer, Pellet, and Jena Rules. We discuss the feasibility of using either of these for query answering under DL^{E}_{RDFS} separately.

Racer [8] provides no SPARQL interface but uses its own functional query language *new Racer Query Language* (nRQL). The system allows for modeling some forms of equation axioms, cf. examples modeling unit conversions in [9], but Racer only uses these for satisfiability testing and not for query answering (which is orthogonal to our approach, as due to the lack of negation there is no inconsistency in DL^{E}_{RDFS}).

SWRL [12, 13] implementations like Pellet [26] allow to handle DL-safe rules [15], that is, rules where each variable appears in at least one non-DL-Atom. We discussed potential modeling of equation axioms as SWRL rules already in Example 1: as mentioned there, rules for each variant of each equation axiom must be added to enable query answering for DL^{E}_{RDFS}. Taking this approach, experiments with Pellet showed that queries over certain data-coherent ABoxes were answered correctly (despite – to our reading – rules like (1)+(2) are not DL-safe in the strict sense), but we still experienced termination problems for e.g. the data and query mentioned in Example 1, since strictly speaking, the data for `:Vienna` is not data-coherent (due to rounding errors). Due to the finite nature of our rewriting, our approach always terminates and is

thus robust even for such – strictly speaking – incoherent data. Sect. 5 will give more details.

Jena[10] provides rule-based inference on top of TDB in a proprietary rule language with built-ins, with SPARQL querying on top. Similar to SWRL, we can encode all variants of equation axioms. Jena allows to execute rules in backward and forward mode, where backward execution does not terminate due to its recursive nature (including empty ABoxes). Forward execution suffers from similar non-termination problems as mentioned above for incoherent data as in Example 1, whereas forward execution for data-coherent ABoxes terminates. Jena offers a hybrid rule based reasoning where pure RDFS inferencing is executed in a backward-chaining manner, but still can be combined with forward rules; this approach was incomplete in our experiments, because property inclusion axioms did not "trigger" the forward rules modeling equation axioms correctly.

5 A Practical Use Case and Experiments

For a prototypical application to compare and compute base indicators of cities – as its needed for studies like Siemens' Green City Index[11] – we collected open data about cities from several sources (DBPedia, Eurostat, ...) from several years. When aggregating these sources into a joint RDF dataset, different kinds of problems such as incoherences, incomplete data, incomparable units along the lines of the extract in Example 1 occurred. Most indicators (such as demography, economy, or climate data) comprise numeric values, where functional dependencies modeled as equation axioms are exploitable to arrive at more complete data from the sparse raw values.

For an initial experiment to test the feasibility of the query answering approach presented in this paper, we assembled a dataset containing ABox 254,081 triples for a total of 3162 city contexts (i.e., when we speak of a "city" sloppily, we actually mean one particular city in a particular year) along with the following (attribute-acyclic) TBox:

e1 $:\mathsf{tempHighC} = (:\mathsf{tempHighF} - 32) \cdot 5 \div 9$
e2 $:\mathsf{populationRateMale} = :\mathsf{populationMale} \div :\mathsf{population}$
e3 $:\mathsf{populationRateFemale} = :\mathsf{populationFemale} \div :\mathsf{population}$
e4 $:\mathsf{area\_km2} = :\mathsf{area\_m2} \div 1000000$
e5 $:\mathsf{area\_km2} = :\mathsf{area\_mile2} \div 2.589988110336$
e6 $:\mathsf{populationDensity} = :\mathsf{population} \div :\mathsf{area\_km2}$

$:\mathsf{City} \sqsubseteq :\mathsf{Location} \quad \mathsf{foaf}:\mathsf{name} \sqsubseteq \mathsf{rdfs}:\mathsf{label} \quad \mathsf{dbpedia}:\mathsf{name} \sqsubseteq \mathsf{rdfs}:\mathsf{label}$

We use the following queries for our experiments:

Q1. Return the population density of all cities:

```
SELECT ?C ?P
WHERE { ?C rdf:type :City . ?C :populationDensity ?P . }
```

Q2. Select cities with a maximum annual temperature above 90°F.

[10] http://jena.apache.org/documentation/inference/index.html
[11] http://www.siemens.com/entry/cc/en/greencityindex.htm

```
SELECT ?C
WHERE { ?C rdf:type :City . ?C rdfs:label ?L .
        ?C :tempHighF ?P . FILTER(?F > 90) }
```

Q3. Select locations with a label that starts with "W" and a population over 1 million:

```
SELECT ?C
WHERE { ?C rdf:type :Location . ?C rdfs:label ?L .
        ?C :population ?P .
        FILTER(?P > 1000000 && STRSTARTS(?L,"W")) }
```

Q4. Select places with a higher female than male population rate.

```
SELECT ?C
WHERE { ?C :populationRateFemale ?F .
        ?C :populationRateMale ?M . FILTER( ?F > ?M ) }
```

Experimental results are summarized in Table 3. For the reasons given in Sect. 4, we compare our approach only to Jena Rules. Experiments were run on the dataset using Jena and ARQ 2.9.2 (without a persistent RDF Store). For Jena Rules, first we encoded the essential RDFS rules plus all variants of equation axioms in a straightforward manner as forward rules, leading to the expected non-termination problems with incoherent data. To avoid this, we created a coherent sample of our dataset (253,114 triples) by removing triples leading to possible incoherences, however still reaching a timeout of 10min for all 4 queries. As an alternative approach, we used Jena's negation-as-failure built-in `noValue` which returns sound but incomplete results, in that it fires a rule only if no value exists for a certain attribute (on the inferences so far or in the data); similar to our approach, this returns complete results for data-coherent datasets and always terminates. As an example of encoding the variants of an axiom in Jena Rules, we show the encoding of equation e6 (which is identical to the naive encoding except the `noValue` predicates). Possible divisions by 0, which we do not need to care about in our SPARQL rewriting, since BIND just filters them out as errors, are caught by `notEqual(`$Quotient$`,0)` predicates.

```
[  (?city :area ?ar) (?city :population ?p)
   notEqual(?ar, 0) quotient(?p, ?ar, ?pd)
   noValue(?city, :populationDensity)
   -> (?city :populationDensity ?d)]
[  (?city :area ?ar) (?city :populationDensity ?pd)
   product(?ar, ?pd, ?p) noValue(?city, :population)
   -> (?city :population ?p)]
[  (?city :populationDensity ?pd) (?city :population ?p)
   notEqual(?pd, 0) quotient(?p, ?pd, ?ar) noValue(?city, :area)
   -> (?city :area ?ar)]
```

Overall, while this experiment was mainly meant as a feasibility study of our query-rewriting approach, the results as shown in Table 3 are promising: we clearly outperform the only rule-based approach we could compare to. However, looking further into alternative implementation strategies and optimizations remains on our agenda.

As a final remark, we observed during our experiments that single Web sources tend to be coherent in the values they report for a single city, thus data-incoherences, i.e. ambiguous results in our queries for one city typically stem from the combination of different sources considered for computing values through equations. As a part of future

work, we aim to further investigate this, building up on our earlier results for combining inferences in SPARQL with conveying provenance information in the results, cf. [27].

6 Further Related Work and Possible Future Directions

OWL ontologies for measurements and units such as QUDT [20], OM [21] provide means to describe units and – to a certain extent – model conversion between these units, though without the concrete machinery to execute these conversions in terms of arbitrary SPARQL queries. Our approach is orthogonal to these efforts in that (a) it provides not only a modeling tool for unit conversions, but integrates attribute equations as axioms in the ontology language, and (b) allows for a wider range of use cases, beyond conversions between pairs of units only. It would be interesting to investigate whether ontologies like QUDT and OM can be mapped to the framework of DL^{E}_{RDFS} or extensions thereof.

Moreover, in the realm of DL-Lite query rewriting, following the PerfectRef algorithm [3] which we base on, there have been a number of extensions and alternative query rewriting techniques proposed [7, 14, 17, 22, 23] which could likewise serve as a basis for extensions by attribute equations. Another obvious direction for further research is the extension to more expressive ontology languages than DL^{E}_{RDFS}. Whereas we have deliberately kept expressivity to a minimum in this paper, apart from further DL-Lite fragments we are particularly also interested in lightweight extensions of RDFS such as OWL LD [6] which we aim to consider for future work.

Apart from query answering, this work opens up research in other reasoning tasks such as query containment of SPARQL queries over DL^{E}_{RDFS}. While containment and equivalence in SPARQL are a topic of active research [4,16,25] we note that containment could in our setting depends not only on the BGPs, but also on FILTERs. E.g., intuitively query Q4 in our setting would be equivalent (assuming `:population` > 0) to

```
SELECT ?C WHERE { ?C :populationFemale ?F .
      ?C :populationMale ?M . FILTER( ?F > ?M ) }
```

While we leave closer investigation for future work, we note another possible connection to related work [24] on efficient query answering under FILTER expression also based in constraint-based techniques.

Lastly, we would like to point out that our approach could be viewed as rather related to Constraint-handling-rules [5] than to mainstream semantic Web rules approaches such as SWRL, etc.; we aim to further look into this.

Table 3. Query response times in seconds

	Coherent Sample of our Dataset			Full Dataset		
#	Our System	Jena naive	Jena noValue	Our System	Jena naive	Jena noValue
Q1	6.5	>600	30.7	7.3	–	30.1
Q2	5.8	>600	32.7	5.7	–	31.3
Q3	7.8	>600	32.5	8.2	–	29.0
Q4	6.9	>600	34.3	7.9	–	32.4

7 Conclusions

We have presented a novel approach to model mathematical equations as axioms in an ontology, along with a practical algorithm for query answering using SPARQL over such enriched ontologies. To the best of our knowledge, this is the first framework that combines ontological reasoning in RDFS, inferencing about functional dependencies among attributes formulated as generic equations, and query answering for SPARQL. Experimental results compared to rule-based reasoning are encouraging. Given the increasing amount of published numerical data in RDF on the emerging Web of data, we strongly believe that this topic deserves increased attention within the Semantic Web reasoning community.

Acknowledgements. Stefan Bischof has been partially funded by the Vienna Science and Technology Fund (WWTF) through project ICT12-015.

References

1. Arenas, M., Botoeva, E., Calvanese, D., Ryzhikov, V., Sherkhonov, E.: Representability in DL-Lite$_R$ knowledge base exchange. In: 25th Int'l DL Workshop, vol. 846, pp. 4–14 (2012)
2. de Bruijn, J., Heymans, S.: Logical foundations of (e)RDF(S): Complexity and reasoning. In: Aberer, K., et al. (eds.) ISWC/ASWC 2007. LNCS, vol. 4825, pp. 86–99. Springer, Heidelberg (2007)
3. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The DL-Lite family. Journal of Automated Reasoning 39(3), 385–429 (2007)
4. Chekol, M.W., Euzenat, J., Genevès, P., Layaïda, N.: Sparql query containment under shi axioms. In: 26th AAAI Conf. (2012)
5. Frühwirth, T.W.: Constraint handling rules: the story so far. In: 8th PPDP, pp. 13–14 (2006)
6. Glimm, B., Hogan, A., Krötzsch, M., Polleres, A.: OWL: Yet to arrive on the web of data? In: WWW 2012 Workshop on Linked Data on the Web (2012)
7. Gottlob, G., Schwentick, T.: Rewriting ontological queries into small nonrecursive datalog programs. In: 13th Int'l KR Conf. (2012)
8. Haarslev, V., Möller, R.: RACER system description. In: Goré, R.P., Leitsch, A., Nipkow, T. (eds.) IJCAR 2001. LNCS (LNAI), vol. 2083, pp. 701–705. Springer, Heidelberg (2001)
9. Haarslev, V., Möller, R.: Description logic systems with concrete domains: Applications for the semantic web. In: 10th Int'l KRDB Workshop (2003)
10. Harris, S., Seaborne, A.: SPARQL 1.1 query language. W3C proposed rec., W3C (2012)
11. Hayes, P.: RDF semantics. W3C rec., W3C (2004)
12. Horrocks, I., Patel-Schneider, P.F.: A proposal for an owl rules language. In: 13th Int'l Conf. on World Wide Web (WWW 2004), pp. 723–731. ACM (2004)
13. Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M.: SWRL: A semantic web rule language combining OWL and RuleML. W3C member subm., W3C (2004)
14. Kontchakov, R., Lutz, C., Toman, D., Wolter, F., Zakharyaschev, M.: The combined approach to ontology-based data access. In: 22nd IJCAI, pp. 2656–2661 (2011)
15. Motik, B., Sattler, U., Studer, R.: Query answering for OWL-DL with rules. Journal of Web Semantics (JWS) 3(1), 41–60 (2005)
16. Pérez, J., Arenas, M., Gutierrez, C.: Semantics and complexity of SPARQL. ACM Transactions on Database Systems 34(3) (2009)

17. Pérez-Urbina, H., Motik, B., Horrocks, I.: Tractable query answering and rewriting under description logic constraints. Journal of Applied Logic 8(2), 186–209 (2010)
18. Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., Rosati, R.: Linking data to ontologies. In: Spaccapietra, S. (ed.) Journal on Data Semantics X. LNCS, vol. 4900, pp. 133–173. Springer, Heidelberg (2008)
19. Prud'hommeaux, E., Seaborne, A. (eds.): SPARQL Query Language for RDF. W3C rec., W3C (2008)
20. Ralph Hodgson, P.J.K.: Qudt - quantities, units, dimensions and data types in owl and xml (2011), `http://www.qudt.org/`
21. Rijgersberg, H., van Assem, M., Top, J.: Ontology of units of measure and related concepts. Semantic Web Journal (SWJ) 4(1), 3–13 (2013)
22. Rosati, R.: Prexto: Query rewriting under extensional constraints in $DL-lite$. In: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 360–374. Springer, Heidelberg (2012)
23. Rosati, R., Almatelli, A.: Improving query answering over dl-lite ontologies. In: 12th Int'l KR Conf. (2010)
24. le Clément de Saint-Marcq, V., Deville, Y., Solnon, C., Champin, P.-A.: Castor: A constraint-based SPARQL engine with active filter processing. In: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 391–405. Springer, Heidelberg (2012)
25. Schmidt, M., Meier, M., Lausen, G.: Foundations of sparql query optimization. In: ICDT (2010)
26. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoner. Journal of Web Semantics (JWS) 5(2), 51–53 (2007)
27. Zimmermann, A., Lopes, N., Polleres, A., Straccia, U.: A general framework for representing, reasoning and querying with annotated semantic web data. Journal of Web Semantics (JWS) 12, 72–95 (2012)

A Comparison of Knowledge Extraction Tools for the Semantic Web

Aldo Gangemi[1,2]

[1] LIPN, Université Paris13-CNRS-SorbonneCité, France
[2] STLab, ISTC-CNR, Rome, Italy

Abstract. In the last years, basic NLP tasks: NER, WSD, relation extraction, etc. have been configured for Semantic Web tasks including ontology learning, linked data population, entity resolution, NL querying to linked data, etc. Some assessment of the state of art of existing Knowledge Extraction (KE) tools when applied to the Semantic Web is then desirable. In this paper we describe a landscape analysis of several tools, either conceived specifically for KE on the Semantic Web, or adaptable to it, or even acting as aggregators of extracted data from other tools. Our aim is to assess the currently available capabilities against a rich palette of ontology design constructs, focusing specifically on the actual semantic reusability of KE output.

1 Introduction

We present a landscape analysis of the current tools for Knowledge Extraction from text (KE), when applied on the Semantic Web (SW).

Knowledge Extraction from text has become a key semantic technology, and has become key to the Semantic Web as well (see. e.g. [31]). Indeed, interest in ontology learning is not new (see e.g. [23], which dates back to 2001, and [10]), and an advanced tool like Text2Onto [11] was set up already in 2005.

However, interest in KE was initially limited in the SW community, which preferred to concentrate on manual design of ontologies as a seal of quality. Things started changing after the linked data bootstrapping provided by DBpedia [22], and the consequent need for substantial population of knowledge bases, schema induction from data, natural language access to structured data, and in general all applications that make joint exploitation of structured and unstructured content. In practice, also Natural Language Processing (NLP) research started using SW resources as background knowledge, and incrementally graph-based methods are entering the toolbox of semantic technologies in the large.

As a result, several tools have appeared providing useful, scalable, application-ready and precise learning of basic semantic data structures, such as tagged named entities, factual relations, topics for document classification, and their integration with SW languages is growing fast. These tools are the bulk of the set considered in this study.

On the other hand, the SW community soon realized that learning just basic semantic data structures fails to achieve complex analytic KE tasks that require e.g. event recognition, event dependency detection, logical relation induction,

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 351–366, 2013.

etc. For example [5] points against the topological sparsity of the results of early ontology learning even at the schema (TBox) level (let alone at the data level), and proves the importance of reusing ontology patterns for improving the topological connectedness of learnt ontologies.

Very recently, more tools are appearing that attempt a deeper KE, typically by hybridizing statistical (trained models) and rule-based methods, and taking advantage of existing knowledge from Linked Open Data as well as of smart heuristics that cling to all sorts of features and structures that become incrementally available on the Web. These tools are also considered in this study.

This study does not intend to be complete in terms of tools tested, parameters singled out for testing, or sample size used in testing. On the contrary, as a *landscape analysis*, it aims to indicate the problems encountered, and some directions and solutions, in order to prepare the ground for a substantial benchmark and a reference evaluation procedure for KE tools on the SW (KE2SW tools).

In Section 2 we make a short recap of the efforts in abridging linguistic and formal semantics, which is the central problem of KE2SW. In Section 3 we survey parameters that can be applied to the comparison between tools for KE2SW: tool performance, structural measures, basic tasks across NLP and SW applications. In Section 4 we describe the text used in the comparison, and the testing principles. In Section 5 we describe the tools. In Section 6 we present the measures obtained from running the tools on the test text, and discuss them.

2 Knowledge Extraction and the Semantic Web

Traditionally, NLP tasks are distinguished into basic (e.g. named entity recognition), and applied (e.g. question answering). When we try to reuse NLP algorithms for the SW, we can also distinguish between basic (e.g. class induction) and application tasks (NL querying of linked data). In this landscape analysis, we map NLP basic tasks to SW ones, and compare different tools with respect to possible functionalities that accomplish those tasks.

The semantics provided by NLP resources is quite different from that assumed for ontologies in knowledge representation and the SW in particular. Moreover, with the exception of formal deep parsing, e.g. based on Discourse Representation Theory (DRT) [21], or Markov Logic [13], the (formal) semantics of NLP data is fairly shallow, being limited to intensional relations between (multi-)words, senses, or synsets, informal identity relation in entity resolution techniques, sense tagging from typically small sets of tags (e.g. WordNets "super senses"), lightweight concept taxonomies, etc.

The actual exploitation and enrichment of ontologies partly relies on the ability to reuse NLP results after appropriate conversion. Such ability is exemplified in some academic and industrial applications that label these techniques as "semantic technology". The current situation of semantic technology can be summarized as in Figure 1, which depicts the relations between formal and linguistic knowledge: linguistic knowledge uses formal background knowledge, but can enable access to formal knowledge (and enrich it) as well. The union of formal and

formalized linguistic knowledge can be further extended by means of automated inferences.

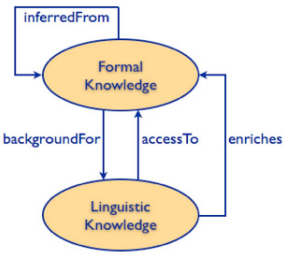


Fig. 1. Hybridization of formal and linguistic knowledge in semantic technologies

Despite recent uprise in adoption of NLP techniques for SW and conversely of SW knowledge for NLP, there is still a large gap between the data structures of lexical and NLP data, and the formal semantics largely adopted for ontologies in the Semantic Web. Current proposals of schemas and formats for abridging NLP and SW, e.g. LMF [15], SKOS-XL [25], LIR [32] Lemon [24],[1] FISE[2], NIF [19], with implementations like Apache Stanbol[3] and NERD[4] are helpful, but they address primarily the "porting" or "lifting" of NLP results or lexical resources to the SW, while the problem of formally reusing NLP results in the SW is mostly left to the choice of specific applications or users. It is therefore interesting to assess the current situation at the tool level, in order to look at the possible best practices that are emerging, as well as to stress them a little bit in order to figure out what can be done in practice, even when there is no direct abridging between the two worlds.

3 Parameters

As we stated in the introduction, this study has no pretense of completeness over all the tools that can be used for KE on the SW: we have tested some of them with a setting that is an attempt to clarify the actual functionalities available when we make KE for a SW application, and we have to figure out the formal semantics of the extracted structures. In other words, the major contribution of the study is a clarification of *what* we do when we use KE for the SW, with an explicit intention to map linguistic semantics into formal semantics. A more complete survey is planned for a journal version of this study.

We firstly distinguished among measures addressing system-level features (time performance, export capabilities, standard compliance), structural measures of the produced ontologies (axioms, density, presence of linguistic annotations, textual grounding, etc.), and measures of achievements with reference to *basic tasks*. Only the third type of measurements has been carried out in this study. The measures precision p, recall r and accuracy a have been applied (when possible) to a subset of the following parameters related to *basic tasks*,

[1] The W3C Ontology-Lexicon Community Group (`http://www.w3.org/community/wiki/Main_Page`) is active on drafting a standard out of Lemon.

[2] `http://stanbol.apache.org/docs/trunk/components/enhancer/enhancementstructure.html#fisetextannotation`

[3] `http://dev.iks-project.eu:8081/enhancer`

[4] `http://nerd.eurecom.fr`

with correspondence between NLP and SW terminology, which of course reflects the different notions of semantics usually assumed in the two fields of expertise:[5]

1. topic extraction (recognition of specific topics, e.g. individuals from the range of the property `dc:subject`), see also [3]
2. named entity recognition (individual induction) [27]
3. named entity resolution (identity resolution for individuals) [4]
4. named entity coreference (coreference of individuals) [29]
5. terminology extraction (induction of constants pertinent to a domain, typically for classes or properties) [18]
   (a) class induction
   (b) property induction
6. sense tagging (≈ class membership induction) [8]
7. sense disambiguation (≈ identity resolution for classes) [28]
8. taxonomy induction (≈ subclass relation induction) [33]
9. (non-taxonomic, non-role, binary) relation extraction (property assertion –fact– induction) [9,2]
10. semantic role labeling (≈ property induction for events and n-ary relations) [26]
11. event detection (≈ n-ary relationship induction) [20]
12. frame detection (≈ n-ary relation –type– induction) [12]

There are other basic tasks that have not been tested, because some are mostly unknown to NLP, some only have approximate counterparts in knowledge representation for the SW, some have been noticed during the study, but are not well attested in either literature. These include at least: schema-level logical structure extraction [35,6,13]: class equivalence, union of classes, class covering, class disjointness, disjoint partition, restriction induction (that in NLP is part of e.g. automatic formalization of glosses); as well as data-level logical structure extraction [6,13]: entity linking (identity/difference between individuals), individual conjunction (complex object), individual disjunction (collection), fact negation (negative property assertion), factual entailment (≈ dependency relation between events or reified relationships) [1], etc.

In order to give a more formal basis to the correspondences provided in the previous list, we have reconstructed some of the current translation practices from NLP to formal semantics, and reported them in Table 1. By no means these are definitive recommendations for translation, due to the variety of requirements and domains of application, which can motivate different choices. For the tasks that have been tested, when RDF or OWL representation of extraction results is not provided by the tools, we have applied Table 1 as a set of default assumptions for translation.[6]

[5] Many tasks have quite a large literature, and we can hardly summarize it here; reference work is cited for some of them.

[6] With conjunction or disjunction of individuals, an ontology can be used to represent e.g. *collections* and their members, or *complex entities* and their parts.

Table 1. Translation table, used when default assumptions are to be applied on the results of a tool. The output of basic tasks not listed here is trivially translated according to model-theoretic semantics (e.g. "union of classes").

Topic	`<Document> dc:subject <Topic>`
Named entity	`owl:NamedIndividual`
Entity resolution (NE)	`owl:sameAs`
Entity coreference	`owl:sameAs`
Term	`owl:Class || owl:ObjectProperty || owl:DatatypeProperty`
Sense tag	`owl:NamedIndividual rdf:type owl:Class`
Sense disambiguation (classes)	`owl:equivalentClass`
Taxonomy (subclasses)	`owl:subClassOf`
Extracted (binary) relation	`owl:ObjectProperty || owl:DatatypeProperty`
Semantic role	`owl:ObjectProperty || owl:DatatypeProperty`
Event	`<Event> rdf:type <Event.type> . <Event> <semrole`$_i$`> <Entity`$_j$`>`
Frame	`<Event.type> owl:subClassOf <Frame>`
Restriction	`owl:Restriction`
Linked entities	`owl:sameAs || owl:differentFrom`
Conjunct of individuals	`owl:NamedIndividual`
Disjunction of individuals	`owl:NamedIndividual`
Factual entailment	`<Event`$_1$`> <dependency> <Event`$_2$`>`

4 The Sample Text

The sample used in this study has been taken from an online article of The New York Times[7] entitled "Syrian Rebels Tied to Al Qaeda Play Key Role in War", and its size has been cut to 1491 characters in order to adapt it to the smallest maximum size of texts accepted by the tested tools (Section 5).[8] The text is cited here (minor typographic editing has been performed for character encoding compatibility across the tools):

The lone Syrian rebel group with an explicit stamp of approval from Al Qaeda has become one of the uprising most effective fighting forces, posing a stark challenge to the United States and other countries that want to support the rebels but not Islamic extremists. Money flows to the group, the Nusra Front, from like-minded donors abroad. Its fighters, a small minority of the rebels, have the boldness and skill to storm fortified positions and lead other battalions to capture military bases and oil fields. As their successes mount, they gather more weapons and attract more fighters. The group is a direct offshoot of Al Qaeda in Iraq, Iraqi officials and former Iraqi insurgents say, which has contributed veteran fighters and weapons. "This is just a simple way of returning the favor to our Syrian brothers that fought with us on the lands of Iraq," said a veteran of Al Qaeda in Iraq, who said he helped lead the Nusra Front's efforts in Syria. The United States, sensing that time may be running out for Syria president Bashar al-Assad, hopes to isolate the group to prevent it from inheriting Syria or fighting on after

[7] `http://www.nytimes.com/2012/12/09/world/middleeast/syrian-rebels-tied-to-al-qaeda-play-key-role-in-war.html`

[8] One text only may seem a small sample even for a landscape analysis, but in practice we had to measure 14 tools across 15 dimensions, with a total amount of 1069 extracted constructs, 727 of which are included in the merged ontology, and 524 in the reference ontology.

Mr. Assad's fall to pursue its goal of an Islamic state. As the United States pushes the Syrian opposition to organize a viable alternative government, it plans to blacklist the Nusra Front as a terrorist organization, making it illegal for Americans to have financial dealings with the group and prompting similar sanctions from Europe.

We have produced one ontology from the output of each tool from the list in Section 5, translating it when necessary according to the default assumptions as in Table 1, or editing it when RDF or OWL parsing was difficult.

As explained in Section 3, we want to assess some information measures on the produced ontologies, and we need some reference knowledge space for that. We have chosen the simplest way to create such a knowledge space: a reference ontology. But how to produce it without introducing subjective biases or arbitrary design decisions?

For this study we have decided not to produce a "gold standard" ontology from a top-down, intellectual ontology design interpreting the text. This choice is due to a lack of requirements: ontology design and semantic technologies are highly dependent on application tasks and expert requirements: it would be too subjective or even unfair to produce an ontology based on an average or ideal task/requirement set.

A possible solution is to choose specific application requirements, and to design the ontology based on them, e.g. "find all events involving a terroristic organization". Another solution is to "merge" all the results of the tested tools, so that each tool is *comparatively* evaluated within the semantic tool space. Of course, the merged ontology needs to be cleaned up of all errors and noise coming from specific tools, in order to produce a reference ontology. This solution is inspired by the typical testing used in information retrieval with incomplete information [7], where supervised relevant results from different methods are merged in order to provide a baseline.

The second solution seemed more attractive to us because if makes us free from the problem of choosing a task that does not look like biasing the test towards a certain tool. It is also interesting as an indicator of how far "merging tools" like Apache Stanbol or NERD can be pushed when integrating multiple KE outputs[9].

The produced ontologies, including the merged and the reference ones, are available online.[10] An analysis of the results based on the measures listed in Section 3 is reported in Section 6.

5 Tools

The tools considered share certain characteristics that make them a low hanging fruit for our landscape analysis. They are available as easily installable downloadable code, web applications, or APIs, and at least in public demo form. They are also tools for Open Domain information extraction, which means that they

[9] In the planned extended survey, we will consider also other experimental settings, including explicit requirements, user behavior, etc.

[10] `http://stlab.istc.cnr.it/documents/testing/ke2swontologies.zip`

are not dependent on training to a specific domain[11]. Their licensing has not been investigated for this study, because we are interested in assessing the state of art functionalities, rather than their legal exploitability in either commercial or academic projects. We have not confined our study to tools that can produce SW output (typically RDF or OWL), because it is usual practice to reuse KE output in SW tools. Therefore, in cases where RDF or OWL is not produced by the tool, we have applied default assumptions on how to convert the output (see Section 3). Finally, certain tools can be configured (in terms of confidence or algorithm to be used) in order to optimize their performance: in this study, we have sticked to default configurations, even if this choice might have penalized some tools (in particular Apache Stanbol).

The following tools have been selected:

- AIDA[12] is a framework and online tool for named entity recognition and resolution. Given a natural-language text or a Web table, it maps mentions of ambiguous names onto canonical entities (e.g., individual people or places) registered in the YAGO2 knowledge base[13], used also to provide sense tagging. AIDA can be configured for the algorithm to be applied (prior probability, key phrase similarity, coherence). It is available as a demo web application or as a Java RMI web service [36].
- AlchemyAPI[14] uses machine learning and natural language parsing technology for analyzing web or text-based content for named entity extraction, sense tagging, as well as for relationships and topics. It does not provide a direct RDF encoding. It is available as a demo web application or as a REST service, also for mobile SDKs.
- Apache Stanbol[15] is an Open Source HTTP service meant to help Content Management System developers to semi-automatically enhance unstructured content with semantic annotations to be able to link documents with related entities and topics. Current enhancers include RDF encoding of results from multilingual named entity recognition and resolution, sense tagging with reference to DBpedia and GeoNames, text span grounding, confidence, and related images. It is available as a demo web application, as a REST service, or downloadable.
- DBpedia Spotlight[16] is a tool for automatically annotating mentions of DBpedia resources in text. It is available as a demo web application, as a REST service, or downloadable.
- CiceroLite[17] (formerly known as Extractiv), performs named entity recognition for English, Arabic, Chinese, and a number of European-language texts. It also performs sense tagging, relation extraction, and semantic role labeling. It is available as a demo web application, and as a REST service.
- FOX[18] is a merger and orchestrator of KE tools, focusing on results that include named entity recognition and resolution, sense tagging, term extraction, and

[11] This is not totally true for PoolParty Knowledge Extractor, but its dependency is harmless for the sake of this study.

[12] `http://www.mpi-inf.mpg.de/yago-naga/aida/`

[13] `http://www.mpi-inf.mpg.de/yago-naga/yago`

[14] `http://www.alchemyapi.com/api/demo.html`

[15] `http://dev.iks-project.eu:8081/enhancer`

[16] `http://dbpedia-spotlight.github.com/demo`

[17] `http://demo.languagecomputer.com/cicerolite`

[18] `http://aksw.org/Projects/FOX.html`

relation extraction. It provides an ontology that generalizes over the sense tags provided by the merged tools. FOX also uses NIF [19] to generalize over textual grounding methods;[19]. It is available as a demo web application.

- FRED[20] is a tool for automatically producing RDF/OWL ontologies and linked data from text. The method is based on deep semantic parsing as implemented Boxer [6], Discourse Representation Theory [21], Linguistic Frames [30], and Ontology Design Patterns [16]. Results are enriched with NER from the Semiosearch Wikifier (see below). It is available as a demo web application, as a REST service, or downloadable. The current output of FRED is either graphic or in Turtle encoding: the second is an "intermediate" specification, which is typically refactored in order to comply to the type of text analyzed: encyclopedic definitions, factual information, etc. [34]
- NERD[21] [17] is a merger of KE tools (at the time of writing: AlchemyAPI, DBpedia Spotlight, Extractiv, Lupedia, OpenCalais, Saplo, SemiTags, Wikimeta, Yahoo! Content Analysis, and Zemanta), currently focusing on results that include named entity recognition and resolution, and sense tagging. It provides a simple ontology that generalizes over the sense tags provided by the merged tools. NERD also uses NIF [19]. It is available as a demo web application, and as a web service, with APIs for Java and Python.
- Open Calais[22] is a KE tool that extracts named entities with sense tags, facts and events. It is available as a web application and as a web service. It has been used via the web application for homogeneity with the other tools. We have also tried the Open Calais TopBraid Composer[23] plugin, which produces an RDF file automatically. The RDF schemata used by Open Calais have a mixed semantics, and have to be refactored in order to be used as a formal output that is relevant to the domain addressed by the text.
- PoolParty Knowledge Discoverer[24] is a text mining and entity extraction tool based on knowledge models, thesauri and linked data. Content, categories, images and tags are recommended automatically when controlled vocabularies are used as a base knowledge model. In other words, Knowledge Discoverer is dependent on a reference knowledge base typically derived from some controlled vocabularies, e.g. a thesaurus. Configuring one controlled vocabulary instead of another makes results completely different. for our test, we have checked it with two configurations: "all kind of topics", and "economy". It is available as a demo web application.
- ReVerb[25] is a program that automatically identifies and extracts binary relationships from English sentences. ReVerb is designed for web-scale information extraction, where the target relations cannot be specified in advance. ReVerb runs on a model trained out of the big dataset of Open Information Extraction web triples. ReVerb takes raw text as input, and outputs (argument1, relation phrase, argument2) triples. It can be downloaded and there is a related web application[26], not used for this study because it does not accept bulk text [14].

[19] http://ontowiki.net/Projects/FOX/files?get=fox_evaluation.pdf
[20] http://wit.istc.cnr.it/stlab-tools/fred
[21] http://nerd.eurecom.fr
[22] http://viewer.opencalais.com/
[23] http://www.topquadrant.com/products/TB_Composer.html
[24] http://poolparty.biz/demozone/general
[25] http://reverb.cs.washington.edu
[26] http://openie.cs.washington.edu/

- Semiosearch Wikifier[27] resolves arbitrary named entities or terms (i.e. either individuals or concepts) on DBpedia entities by integrating several components: a named entity recognizer (currently Alchemy[28]), a semiotically informed index of Wikipedia pages (text is selected from page sections and metadata according to explicit formal queries), as well as matching and heuristic strategies. It is available as a demo web application.
- Wikimeta[29] is a tool for multilingual named entity recognition and resolution, and sense tagging. It links texts data to concepts of the Linked Open Data network through various sources like DBpedia, Geonames, CIA World Factbook or directly to Wikipedia or the web when there is no available resource. It is available as a demo web application and as a REST service.
- Zemanta[30] provides enriched content for articles, images and websites to bloggers. It matches text with publicly available content and displays it in the creation tool as it is being written. Behind its interaction capabilities, it does named entity recognition and resolution, as well as content linking. It is available as a demo web application and as an API for content management systems.

6 Results and Discussion

We firstly include (Table 2) a table including all the tools with their featured tasks. We have considered only a subset of the basic tasks (1 to 12), from the list given in Section 3. Some measures of these 12 tasks are not included in the paper for space reasons, but are available online[31]. Tool-level and structural measures have not been addressed in this landscape analysis. We have made an assessment of the precision, recall, F-measure, and accuracy of the tools distinguishing them by basic tasks. Measures have been calculated on the merged ontology for each one of the 12 tasks, so that the merged output is used as the upper limit for the measurement. Only eight measures are included in this paper.

Topic extraction tools produce output including broad topics (Alchemy and Open Calais), topics resolved into Wikipedia categories (PoolParty), subject tags (Alchemy), and social tags (Open Calais). We have decided to treat them all as topics, since a real distinction is very hard to make at the theoretical level, while the methods to extract them (e.g. from social tag spaces or Wikipedia) are relevant for the specific task, but do not impact much at the level of produced knowledge, unless there is a resolution performed with respect to e.g. Linked Open Data (this is true only for PoolParty Knowledge Discoverer). Table 3 contains the results, and show very different performances. 64 topics have been extracted and merged by the three tools, with an overall precision (manually evaluated after merging) of .72.

Named entity recognition in this KE2SW study was assessed only for named entities that are typically represented as individuals in an ontology, while the

[27] http://wit.istc.cnr.it/stlab-tools/wikifier
[28] http://www.alchemyapi.com/api/demo.html
[29] http://www.wikimeta.com/wapi/semtag.pl
[30] http://www.zemanta.com/demo/
[31] http://stlab.istc.cnr.it/stlab/KnowledgeExtractionToolEval

Table 2. Summary of featured basic tasks (as obtained from testing)

Tool	Topics	NER	NE-RS	TE	TE-RS	Senses	Tax	Rel	Roles	Events	Frames
AIDA	–	+	+	–	–	+	–	–	–	–	–
Alchemy	+	+	–	+	–	+	–	+	–	–	–
Apache Stanbol	–	+	+	–	–	+	–	–	–	–	–
CiceroLite	–	+	+	+	+	+	–	+	+	+	+
DB Spotlight	–	+	+	–	–	+	–	–	–	–	–
FOX	+	+	+	+	+	+	–	–	–	–	–
FRED	–	+	+	+	+	+	+	+	+	+	+
NERD	–	+	+	–	–	+	–	–	–	–	–
Open Calais	+	+	–	–	–	+	–	–	–	+	–
PoolParty KD	+	–	–	–	–	–	–	–	–	–	–
ReVerb	–	–	–	–	–	–	–	+	–	–	–
Semiosearch	–	–	+	–	+	–	–	–	–	–	–
Wikimeta	–	+	–	+	+	+	–	–	–	–	–
Zemanta	–	+	–	–	–	–	–	–	–	–	–

named entities that are typically appropriate to class or property names are assessed in the terminology extraction and resolution measures (not presented here). After merging and cleaning, 58 named entities remained for evaluation. Table 5 contains the results for this task, showing here a quite consistent behavior across tools. Out of the 58 named entities (individuals) extracted and merged, the overall precision (manually evaluated after merging) is .25. Alchemy, AIDA, and Zemanta stand out on all measures.

Several issues have been encountered when merging and cleaning the results from the different tools. In some cases, named entities have been given directly in terms of *resolved* entities: we have decided to evaluate them as correct or wrong based on the validity of the resolution, even if there is no specific indication of the phrase that has been recognized. In some cases, terms have been recognized instead of named entities: when these are actually referential usages of terms (e.g. "the rebels") they have been accepted as individuals, otherwise they counted as errors. Finally, we had to decide if we need to count tokens (multiple references to the same entity in text) or just types. After a detailed scrutiny, the effect of tokens on precision and recall seemed negligible (two failed recognitions added by tokens across all tools), so we decided to skip tokens for this study.

Table 6 contains the results for the named entity resolution task. Out of the 19 named entities (individuals) that have been resolved, the overall precision (manually evaluated after merging) is .55. AIDA stands out in terms of precision and accuracy, while Wikimeta is high on recall. Most resolutions are made with respect to DBpedia entities.

Table 4 contains the results for the sense tagging task. 19 named entities have been tagged, and the type triples have been merged, with an overall precision (manually evaluated after merging) of .74. Overall, the tools performed quite well on this task (with Wikimeta standing out on recall and accuracy), confirming

Table 3. Comparison of topic extraction tools

Topic Ex Tool	*p*	*r*	F_1	*a*
Alchemy	.74	.50	.60	.52
OpenCalais	1.00	.28	.44	.48
PoolParty KE	.50	.22	.30	.28

Table 4. Comparison of sense tagging tools

Sense Tagging Tool	*p*	*r*	*F*1	*a*
AIDA	1.00	.57	.73	.64
Alchemy	1.00	.57	.73	.64
Apache Stanbol	1.00	.43	.60	.50
CiceroLite	.64	.64	.64	.54
DBpedia Spotlight	.83	.36	.50	.42
FOX	1.00	.50	.67	.57
FRED+SST	.75	.43	.55	.48
NERD	.90	.64	.75	.69
OpenCalais	1.00	.50	.67	.57
Wikimeta	.85	.79	.81	.80
Zemanta	1.00	.21	.35	.27

Table 5. Comparison of named entity recognition tools

NER Tool	*p*	*r*	*F*1	*a*
AIDA	1.00	.57	.73	.89
Alchemy	1.00	.57	.73	.89
Apache Stanbol	.55	.43	.48	.77
CiceroLite	.79	.79	.79	.89
DBpedia Spotlight	.75	.21	.33	.79
FOX	.88	.50	.64	.86
FRED	.73	.57	.64	.84
NERD	.73	.79	.76	.88
Open Calais	.70	.50	.58	.82
Wikimeta	.71	.71	.71	.86
Zemanta	.92	.79	.85	.93

Table 6. Comparison of named entity resolution tools

NE Resolution Tool	*p*	*r*	*F*1	*a*
AIDA	1.00	.64	.78	.80
Apache Stanbol	.33	.36	.35	.25
CiceroLite	1.00	.55	.71	.75
DBpedia Spotlight	.75	.27	.40	.55
FOX	.88	.64	.74	.75
FRED+Semiosearch	.80	.36	.50	.60
NERD	1.00	.27	.43	.60
Semiosearch	.67	.55	.60	.60
Wikimeta	.71	.91	.80	.75

Table 7. Comparison of terminology extraction tools

Term Extraction Tool	*p*	*r*	*F*1	*a*
Alchemy	.76	.16	.26	.20
CiceroLite	1.00	.17	.29	.21
FOX	.90	.27	.42	.33
FRED	.93	.89	.91	.90
Wikimeta	1.00	.03	.06	.04

Table 8. Comparison of terminology resolution tools

Term Resolution Tool	*p*	*r*	*F*1	*a*
CiceroLite	1.00	.05	.10	.07
FOX	.71	.63	.67	.65
FRED+Semiosearch	1.00	.05	.10	.07
Semiosearch	.41	.47	.44	.46
Wikimeta	.33	.05	.09	.07

the good results from literature when using DBpedia and other linked data as background knowledge.

Table 7 contains the results of the terminology extraction task. 109 terms have been extracted and merged by five tools, with an overall precision (manually

Table 9. Comparison of relation extraction tools

RelEx Tool	*p*	*r*	*F*1	*a*
Alchemy	.69	.25	.37	.30
CiceroLite	.90	.20	.33	.25
FRED	.84	.82	.83	.82
ReVerb	.67	.23	.34	.27

Table 10. Comparison of event detection tools

Event Detection Tool	*p*	*r*	*F*1	*a*
CiceroLite	1.00	.14	.24	.18
FRED	.73	.93	.82	.87
Open Calais	.50	.03	.06	.04

evaluated after merging) of .94, and with FRED standing out on all measures. Table 8 contains the results of "terminology resolution", which is typically the output of a Word Sense Disambiguation (WSD) algorithm; however, the tested tools do not include any WSD components, therefore disambiguation is just the result of performing NE resolution on terms that refer to classes rather than to individuals. Indeed, only 35 out of 109 terms (.32%) have been resolved, with an overall precision of .54. FOX stands out in this task, with an accuracy of .65.

Table 9 contains the results for the relation extraction task. The variety of relations found is here very high, since the techniques and the assumptions on the relation pattern to discover are very different. In particular, FRED is based on neo-Davidsonian event-reification semantics, for example it represents the sentence: *they gather more weapons* as an event `gather_1` with the semantic role: `agent` and `theme`, played by the entities `thing_1` and `weapon_1`. On the contrary, Alchemy and ReVerb follow a strict binary style, e.g. they extract a relationship `gather(they,more weapons)`. CiceroLite has an intermediate approach, trying to guess the arity of the relation, and here it has a binary: `gather(they,weapons)`.

In the previous example, it seems quite efficient to go with the binary style, because the relation/event is used with two explicit arguments. However, things change when there are more than two arguments. For example, with the sentence: *it plans to blacklist the Nusra Front as a terrorist organization*, binary-style tools do not go very far. There are important cohesion aspects here that are hardly caught by means of simple triple patterns: *it* is an anaphora for `United States`, `blacklist` is used with three explicit arguments, and `plan` is used with two, but one of them is the sub-sentence governed by `blacklist`. Here are the representations given by the four tools in this case:

- (ReVerb): no extraction
- (Alchemy): `plans to blacklist(it,the Nusra Front as a terrorist organization)`
- (CiceroLite): `plans to blacklist(it,front,(AS) a terrorist organization)`
- (FRED): `experiencer(plan_3,United_States) ; theme(plan_3,blacklist_3) ; agent(blacklist_3,United_States) ; patient(blacklist_3,NusraFront) ; as(blacklist_3,organization_3) ; TerroristOrganization(organization_3)`

For this task, we have decided to exclude event reification, which is instead tested as a separate task. However, this choice does not penalize FRED, because besides semantic roles, it infers "semantic-web-style" binary relations. For example, from

the phrase: *its fighters*, where *its* is an anaphora to Al-Qaeda, FRED extracts: `fighterOf(fighter_1,AlQaeda)`.

When merging the results, we have then considered only non-role binary relations from the four tools, generating 62 relations, with an overall precision (manually evaluated after merging) of .71. The results for this task seem then very promising, and deserve further investigation on how much integration can be done among the different perspectives. For example, a stronger merging could be done by mapping reified events from FRED or CiceroLite to purely binary relations from Alchemy or ReVerb. This may be done in OWL2 by exploiting punning constructs.

Table 10 contains the results for the event detection task. Only 3 tools contain such functionality: CiceroLite, FRED, and Open Calais. As we commented in the previous test, in order to perform event detection, a tool needs also to perform semantic role labeling. FRED and Open Calais also apply some typing of events and values filling the roles, so that they can also be considered "frame detection" tools [34]. For example, Open Calais provides the following frame on top of a detected event from the sentence: *the United States pushes the Syrian opposition to organize a viable alternative government*: `DiplomaticRelations(diplomaticentity: United States ; diplomaticaction: opposition ; diplomaticentity: viable alternative government).`

FRED is the only tool here that provides RDF output (at least from the web applications that we have tested), and resolves event frames onto reference lexicons (VerbNet and FrameNet). After merging the results, we have generated 40 events, with an overall precision (manually evaluated after merging) of .73. The difference in recall is meaningful in this task (as well as in the previous one): FRED uses a categorial parser to extract syntactic structures that are formalized as events (i.e. it provides *deep parsing*, while the other tools apparently use a purely statistical approach with *shallow parsing*, which is known to reach a much lower recall on this task. FRED stands out also in precision, which seems to confirm that deep parsing approach positively correlates with good results on relation and event extraction.

The measures on semantic role labeling and frame detection (only available on FRED and CiceroLite) are not shown here for space reasons[32], but they contain a detailed analysis of the elements extracted for the task: semantic roles, correctness of roles, correctness of fillers, correctness of frames, and coreference resolution. If we simply sum all elements (297 in total), FRED performed better, with an accuracy of .82.

7 Conclusions

We have presented the results of a landscape analysis in the area of knowledge extraction for the Semantic Web (KE2SW). We have investigated the feasibility of a comparison among KE tools when used for SW tasks. We have proved that this is feasible, but we need to create formally correct correspondences between NLP basic tasks, and SW population basic tasks. Design activities to obtain

[32] They are available at `http://stlab.istc.cnr.it/stlab/SRLFE`

semantic homogeneity across tool outputs are required. In addition, evaluation and measures differ across different tasks, and a lot of tool-specific issues emerge when comparing the outputs. This study results to be a first step in the creation of adequate benchmarks for KE applied to SW, and proves the importance of integrating measurement of different tasks in the perspective of providing useful analytic data out of text. Future work includes an actual experiment on a larger dataset, also exploiting integration functionalities provided by platforms like NERD, FOX and Stanbol.

A practical conclusion of this study is that tools for KE provide good results for all the tested basic tasks, and there is room for applications that integrate NLP results for the Semantic Web. Firstly, the measures for merged ontologies result to be good enough, and we imagine optimization methods to filter out the contribution coming from worst tools for a certain task. Secondly, with appropriate semantic recipes (transformation patterns), the production of merged ontologies can be automatized. Merging and orchestrating applications like Stanbol Enhancers, NERD and FOX with standards like NIF, are on the right track, and refactoring components like Stanbol Rules[33] make it possible to customize the output in appropriate ways for reasoning over the Web the Data.

References

1. Androutsopoulos, I., Malakasiotis, P.: A survey of paraphrasing and textual entailment methods. CoRR, abs/0912.3747 (2009)
2. Banko, M., Etzioni, O.: The tradeoffs between open and traditional relation extraction. In: Annual Meeting of the ACL (2008)
3. Berry, M.W., Castellanos, M.: Survey of Text Mining II: Clustering, Classification and Retrieval. Springer (2008)
4. Bhattacharya, I., Getoor, L.: Collective entity resolution in relational data. ACM Transactions on Knowledge Discovery from Data (ACM-TKDD) (2007)
5. Blomqvist, E.: Ontocase-automatic ontology enrichment based on ontology design patterns. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 65–80. Springer, Heidelberg (2009)
6. Bos, J.: Wide-Coverage Semantic Analysis with Boxer. In: Bos, J., Delmonte, R. (eds.) Semantics in Text Processing, pp. 277–286. College Publications (2008)
7. Buckley, C., Voorhees, E.M.: Retrieval evaluation with incomplete information. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2004, pp. 25–32. ACM, New York (2004)
8. Ciaramita, M., Altun, Y.: Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In: Proceedings of EMNLP 2006, Sydney, Australia (2006)
9. Ciaramita, M., Gangemi, A., Ratsch, E., Saric, J., Rojas, I.: Unsupervised learning of semantic relations between concepts of a molecular biology ontology. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence, IJCAI 2005 (2005)

[33] `http://stanbol.apache.org/docs/trunk/components/rules/`

10. Cimiano, P.: Ontology Learning and Population from Text: Algorithms, Evaluation and Applications. Springer (2006)
11. Cimiano, P., Völker, J.: Text2onto - a framework for ontology learning and data-driven change discovery (2005)
12. Coppola, B., Gangemi, A., Gliozzo, A., Picca, D., Presutti, V.: Frame detection over the semantic web. In: Aroyo, L., et al. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 126–142. Springer, Heidelberg (2009)
13. Davis, J., Domingos, P.: Deep transfer: A markov logic approach. AI Magazine 32(1), 51–53 (2011)
14. Fader, A., Soderland, S., Etzioni, O.: Identifying relations for open information extraction. In: Proc. of the Conference of Empirical Methods in Natural Language Processing, EMNLP 2011, Edinburgh, Scotland, UK (2011)
15. Francopoulo, G., George, M., Calzolari, N., Monachini, M., Bel, N., Pet, M., Soria, C.: Lexical markup framework (LMF). In: Proc. of the International Conference on Language Resources and Evaluation, LREC, Genoa, Italy. ACL (2006)
16. Gangemi, A., Presutti, V.: Ontology Design Patterns. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies, 2nd edn. Springer (2009)
17. Rizzo, G., Troncy, R., Hellmann, S., Bruemmer, M.: NERD meets NIF: Lifting NLP extraction results to the linked data cloud. In: LDOW, 5th Wks. on Linked Data on the Web, Lyon, France (April 2012)
18. Hartmann, S., Szarvas, G., Gurevych, I.: Mining multiword terms from wikipedia. In: Pazienza, M.T., Stellato, A. (eds.) Semi-Automatic Ontology Development: Processes and Resources, pp. 226–258. IGI Global, Hershey (2012)
19. Hellmann, S., Lehmann, J., Auer, S.: Linked-data aware URI schemes for referencing text fragments. In: ten Teije, A., Völker, J., Handschuh, S., Stuckenschmidt, H., d'Acquin, M., Nikolov, A., Aussenac-Gilles, N., Hernandez, N. (eds.) EKAW 2012. LNCS, vol. 7603, pp. 175–184. Springer, Heidelberg (2012)
20. Hogenboom, F., Frasincar, F., Kaymak, U., de Jong, F.: An overview of event extraction from text. In: Proceedings of Derive 2011 Workshop, Bonn (2011)
21. Kamp, H.: A theory of truth and semantic representation. In: Groenendijk, J.A.G., Janssen, T.M.V., Stokhof, M.B.J. (eds.) Formal Methods in the Study of Language, vol. 1, pp. 277–322. Mathematisch Centrum (1981)
22. Lehmann, J., Bizer, C., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia - A Crystallization Point for the Web of Data. Journal of Web Semantics 7(3), 154–165 (2009)
23. Maedche, A., Staab, S.: Ontology learning for the semantic web. IEEE Intelligent Systems 16, 72–79 (2001)
24. McCrae, J., Spohr, D., Cimiano, P.: Linking lexical resources and ontologies on the semantic web with lemon. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) ESWC 2011, Part I. LNCS, vol. 6643, pp. 245–259. Springer, Heidelberg (2011)
25. Miles, A., Bechhofer, S.: Skos simple knowledge organization system extension for labels (skos-xl). W3C Recommendation (2009), `http://www.w3.org/TR/skos-reference/skos-xl.html`
26. Moschitti, A., Pighin, D., Basili, R.: Tree kernels for semantic role labeling. Computational Linguistics 34(2), 193–224 (2008)
27. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Journal of Linguisticae Investigationes 30(1) (2007)
28. Navigli, R.: Word sense disambiguation: A survey. ACM Comput. Surv. 41(2) (2009)

29. Ng, V.: Supervised noun phrase coreference research: The first fifteen years. In: ACL (2010)
30. Nuzzolese, A.G., Gangemi, A., Presutti, V.: Gathering Lexical Linked Data and Knowledge Patterns from FrameNet. In: Proc. of the 6th International Conference on Knowledge Capture (K-CAP), Banff, Alberta, Canada, pp. 41–48 (2011)
31. Pazienza, M.T., Stellato, A.: Semi-Automatic Ontology Development: Processes and Resources. IGI Global, Hershey (2012)
32. Peters, W., Montiel-Ponsoda, E., Aguado de Cea, G., Gomez-Perez, A.: Localizing ontologies in owl. In: Proceedings of OntoLex Workshop (2007), http://olp.dfki.de/OntoLex07/
33. Ponzetto, S.P., Strube, M.: Taxonomy induction based on a collaboratively built knowledge repository. Artificial Intelligence 175 (2011)
34. Presutti, V., Draicchio, F., Gangemi, A.: Knowledge extraction based on discourse representation theory and linguistic frames. In: ten Teije, A., Völker, J., Handschuh, S., Stuckenschmidt, H., d'Acquin, M., Nikolov, A., Aussenac-Gilles, N., Hernandez, N. (eds.) EKAW 2012. LNCS, vol. 7603, pp. 114–129. Springer, Heidelberg (2012)
35. Völker, J., Rudolph, S.: Lexico-logical acquisition of owl dl axioms – an integrated approach to ontology refinement (2008)
36. Yosef, M.A., Hoffart, J., Bordino, I., Spaniol, M., Weikum, G.: Aida: An online tool for accurate disambiguation of named entities in text and tables. In: Proceedings of the 37th International Conference on Very Large Databases, VLDB 2011, Seattle, WA, US (2011)

Constructing a Focused Taxonomy from a Document Collection

Olena Medelyan[1], Steve Manion[1], Jeen Broekstra[1], Anna Divoli[1], Anna-Lan Huang[2], and Ian H. Witten[2]

[1] Pingar Research, Auckland, New Zealand
{alyona.medelyan,steve.manion,anna.divoli}@pingar.com, jeen@rivuli-development.com
[2] University of Waikato, Hamilton, New Zealand
{ahuang,ihw}@cs.waikato.ac.nz

Abstract. We describe a new method for constructing custom taxonomies from document collections. It involves identifying relevant concepts and entities in text; linking them to knowledge sources like Wikipedia, DBpedia, Freebase, and any supplied taxonomies from related domains; disambiguating conflicting concept mappings; and selecting semantic relations that best group them hierarchically. An RDF model supports interoperability of these steps, and also provides a flexible way of including existing NLP tools and further knowledge sources. From 2000 news articles we construct a custom taxonomy with 10,000 concepts and 12,700 relations, similar in structure to manually created counterparts. Evaluation by 15 human judges shows the precision to be 89% and 90% for concepts and relations respectively; recall was 75% with respect to a manually generated taxonomy for the same domain.

1 Introduction

Domain-specific taxonomies constitute a valuable resource for knowledge-based enterprises: they support searching, browsing, organizing information, and numerous other activities. However, few commercial enterprises possess taxonomies specialized to their line of business. Creating taxonomies manually is laborious, expensive, and unsustainable in dynamic environments (e.g. news). Effective automatic methods would be highly valued.

Automated taxonomy induction has been well researched. Some approaches derive taxonomies from the text itself [1], some from Wikipedia [2], while others combine text, Wikipedia and possibly WordNet to either extend these sources with new terms and relations [3] or carve a taxonomy tailored to a particular collection [4,5]. Our research falls into the last category, but extends it by defining a framework through which any combination of knowledge sources can drive the creation of document-focused taxonomies.

We regard taxonomy construction as a process with five clearly defined stages. The first, initialization, converts documents to text. The second extracts concepts and named entities from text using existing NLP tools. The third connects

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 367–381, 2013.

named entities to Linked Data sources like Freebase and DBpedia. The fourth identifies conflicting concept mappings and resolves them with an algorithm that disambiguates concepts that have matching labels but different URIs. The fifth connects the concepts into a single taxonomy by carefully selecting semantic relations from the original knowledge sources, choosing only relations that create meaningful hierarchies given the concept distribution in the input documents. These five stages interoperate seamlessly thanks to an RDF model, and the output is a taxonomy expressed in SKOS, a standard RDF format.

The method itself is domain independent—indeed the resulting taxonomy may span multiple domains covered by the document collection and the input knowledge sources. We have generated and made available several such taxonomies from publicly available datasets in five different domains.[1] This paper includes an in-depth evaluation of a taxonomy generated from news articles. Fifteen human judges rated the precision of concepts at 89% and relations at 90%; recall was 75% with respect to a manually built taxonomy for the same domain. Many of the apparently missing concepts are present with different—and arguably more precise—labels.

Our contribution is threefold: (a) an RDF model that allows document-focused taxonomies to be constructed from any combination of knowledge sources; (b) a flexible disambiguation technique for resolving conflicting mappings and finding equivalent concepts from different sources; and (c) a set of heuristics for merging semantic relations from different sources into a single hierarchy. Our evaluation shows that current state-of-the-art concept and entity extraction tools, paired with heuristics for disambiguating and consolidating them, produce taxonomies that are demonstrably comparable to those created by experts.

2 Related Work

Automatic taxonomy induction from text has been studied extensively. Early corpus-based methods extract taxonomic terms and hierarchical relations that focus on the intrinsic characteristics of a given corpus; external knowledge is rarely consulted. For example, hierarchical relations can be extracted based on term distribution statistics [6] or using lexico-syntactic patterns [7,1]. These methods are usually unsupervised, with no prior knowledge about the corpus. However, they typically assume only a single sense per word in the corpus, and produce taxonomies based on words rather than word senses.

Research has been conducted on leveraging knowledge bases to facilitate taxonomy induction from both closed- and open-domain text collections. Some researchers derive structured taxonomies from semi-structured knowledge bases [2,8] or from unstructured content on the Web at large [9]. Others expand knowledge bases with previously unknown terms and relations discovered from large corpora—for example, Matuszek et al. enrich the Cyc knowledge base with information extracted from the Web [10], while Snow et al. expand WordNet with new synsets by

[1] `http://bit.ly/f-step`

using statistical classifiers built from lexical information extracted from news articles [3]. Still others interlink documents and knowledge bases: they match phrases in the former with concepts in the latter [11,12] and identify taxonomic relations between them [4,5]. These studies do address the issue of sense ambiguity: polysemous phrases are resolved to their intended senses while synonyms are mapped to the same concept. However, they typically only consult a single source and users do not intervene in the taxonomy construction process.

The Castanet project [4] and Dakka and Ipeirotis's research [5] relate closely to our work. They both derive hierarchical metadata structures from text collections and both consult external sources—WordNet in the former case and Wikipedia, WordNet and the Web in the latter—to find important concepts in documents. Castanet identifies taxonomic relations based on WordNet's `is-a` relations, whereas Dakka and Ipeirotis use subsumption rules [6]. The latter only select those taxonomic concepts for final groupings that occur frequently in the documents in non-related contexts. In contrast to our work, both studies represent the extracted information as hierarchical faceted metadata: the outcome is no longer a single taxonomy but is instead split into separate facets. Although Dakka and Ipeirotis consult multiple sources, they do not check which concepts are the same and which are different. In contrast, we explicitly address the problem of sense disambiguation and consolidation with multiple sources.

Our work also intersects with research on relation extraction and ontology induction from text, the closest being [13], which also links phrases in text to Wikipedia, DBpedia and WordNet URIs, extracts relations, and represents them as RDF. However, their input is a single short piece of text, whereas we analyze an entire document collection as a whole, and focus on organizing the information hierarchically.

3 Architecture of the Taxonomy Generator

The primary input to our taxonomy generator is a collection of documents and, optionally, a taxonomy for a related domain (e.g., the Agrovoc thesaurus or the Gene ontology). Our system automatically consults external knowledge sources, and links concepts extracted from the documents to terminology in these sources. By default we use Freebase, DBpedia and Wikipedia, but domain-specific linked data sources like Geonames, BBC Music, or the Genbank Entrez Nucleotide database can also be consulted.[2] Finally, a small taxonomy with preferred root nodes can be supplied to guide the upper levels of the generated taxonomy.

3.1 Defining Taxonomies in SKOS

The result of each step of the taxonomy generation process is stored as an RDF data structure, using the Simple Knowledge Organization System vocabulary. SKOS is designed for sharing and linking thesauri, taxonomies, classification

[2] Suitable linked data sources can be found at `http://thedatahub.org/group/lodcloud`

schemes and subject heading systems via the Web.[3] An SKOS model consists of a hierarchical collection of *concepts*, defined as "units of thought"—abstract entities representing ideas, objects or events. A concept is modeled as an instance of the class `skos:Concept`. An `skos:prefLabel` attribute records its preferred name and `skos:altLabel` attributes record optional synonyms. Concepts are linked via semantic relations such as `skos:broader` (to indicate that one concept is broader in meaning than another) and its inverse `skos:narrower`. These relations allow concepts to be structured into a taxonomic hierarchy.

Our goal is to produce a new knowledge organization system (a taxonomy) based on heterogeneous sources, including concepts extracted from text as well as concepts in existing sources, and SKOS is a natural modeling format. Also, many existing public knowledge systems are available online as SKOS data,[4] and reusing these sources ensures that any taxonomy we generate is immediately linked via concept mappings to third-party data sources on the Web.

3.2 Information Model

We have built a set of loosely coupled components that perform the individual processing steps. Each component's results are stored as RDF data in a central repository using the OpenRDF Sesame framework [14].

Figure 1 shows the information model. The central class is `pw:Ngram`, which represents the notion of an extracted string of N words. The model records every position of the ngram in the input text, and each occurrence of the same ngram in the same document is a single instance of the `pw:Ngram` class.

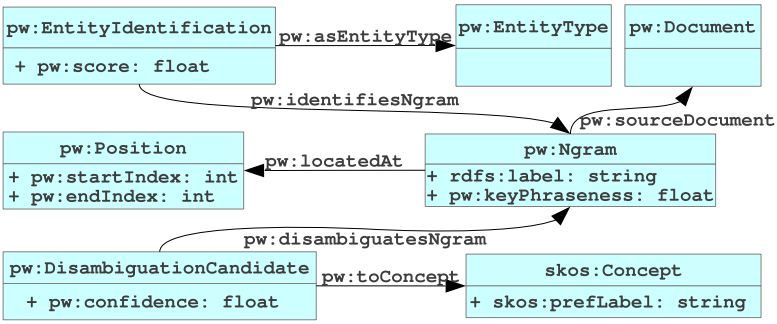


Fig. 1. Shared RDF model for ngram and entity information

The `pw:EntityType` class supports entity typing of ngrams. It has a fixed number of instances representing types such as people, organizations, locations, events, etc. In order to be able to record the relation between an ngram and its type, as well as an identification score reported by the extraction tool, the relation is modeled as an an object, of type `pw:EntityIdentification`.

[3] See `http://www.w3.org/2004/02/skos`
[4] See a.o. `http://www.w3.org/2001/sw/wiki/SKOS/Datasets`

`pw:DisambiguationCandidate` is introduced to allow ngrams to be annotated with corresponding concepts from external sources. This class records the relation (and the system's confidence in it) between an extracted ngram and an external source. These external sources are modeled as instances of `skos:Concept`. They are the building blocks of the taxonomy we generate.

Using a shared RDF model to hold extracted data ensures that components can interoperate and reuse each other's results. This is a significant advantage: it facilitates the use of different language processing tools in a single system by mapping their outputs to a common vocabulary. Moreover, users can add other Linked Data sources, and insert and remove processing steps, as they see fit. It can also be used for text annotation.[5]

In addition, the use of an RDF repository allows one to formulate SPARQL[6] queries over the aggregated data. Using these, data from different components can be analyzed quickly and efficiently at each processing step.

4 Generating the Taxonomy

Figure 2 shows the processing steps in our system, called F-STEP (Focused SKOS Taxonomy Extraction Process). Existing tools are used to extract entities and concepts from document text (steps 2a and 2b respectively in the Figure). Purpose-built components annotate entities with information contained in Linked Data sources (step 3), disambiguate concepts that are mapped to the same ngram (step 4), and consolidate concepts into a hierarchy (step 5).

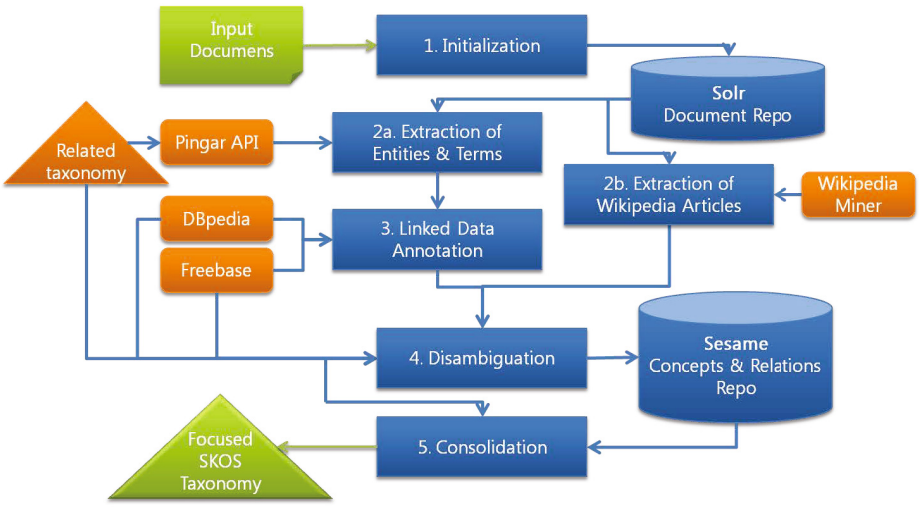


Fig. 2. Automated workflow for turning input documents into a focused taxonomy

[5] A possible alternative is the recently-defined NLP2RDF format `http://nlp2rdf.org`.
[6] See `http://www.w3.org/TR/sparql11-query/`

4.1 Initialization

Taxonomies organize knowledge that is scattered across documents. To federate inputs stored on file systems, servers, databases and document management systems, we use Apache Tika to extract text content from various file formats and Solr for scaleable indexing.[7] Solr stores multiple document collections in parallel, each document being referenced via a URL, which allows concepts to be linked back to the documents containing them in our RDF model.

4.2 Extracting Named Entities and Concepts

Extraction step 2a in Figure 2 uses a text analytics API[8] to identify names of people, organizations and locations, and to identify relevant terms in an existing taxonomy if one is supplied. Step 2b uses the Wikipedia Miner toolkit [15] to relate documents to relevant concepts in Wikipedia.

Named Entities. Names of people, organizations, and locations are concepts that can usefully be included in a taxonomy; existing systems extract such entities with an accuracy of 70%–80% [16]. We extract named entities from the input documents using the text analytics API and convert its response to RDF. Named entities are represented by a `pw:EntityIdentification` relation between the original ngram and an entity type. The entities are passed to the annotation step to disambiguate any matches to Linked Data concepts.

Concepts from Related Taxonomies. As mentioned in Section 3, the input can include one or more taxonomies from related domains. The same text analytics API records any concepts in a related taxonomy that appear in the input documents, maps them to SKOS, and links to the source document ngram via a `pw:DisambiguationCandidate` relation.

Concepts from Wikipedia. Each Wikipedia article is regarded as a "concept." Articles describe a single concept, and for (almost) any concept there exists a Wikipedia article. We use the Wikipedia Miner toolkit to annotate ngrams in the text with corresponding Wikipedia articles. This toolkit allows the number of annotations to be controlled, and disambiguates ngrams to their correct meaning—for example, the word *kiwi* may refer to a.o. a bird, a fruit, a person from NZ, or the NZ national rugby league team, all of which have distinct Wikipedia entries. The approach is described in detail in [15].

The user determines what kind of concepts will be included in the taxonomy. For example, if no related taxonomies are available, only named entities and Wikipedia articles returned by the Wikification process will be included in the final taxonomy.

[7] See `http://tika.apache.org/` and `http://lucene.apache.org/solr/`
[8] See `http://apidemo.pingar.com`

4.3 Annotating with Linked Data

Once entities such as people, places, and organisations have been extracted, the annotation step queries Freebase [17] and DBpedia [18] for corresponding concepts (Figure 2, step 3). The queries are based on the entity's type and label, which is the only structured information available at this stage. Other Linked Data sources can be consulted in this step, either by querying via a SPARQL endpoint,[9] which is how we consult DBpedia, or by accessing the Linked Data source directly over the HTTP protocol.

We define mappings of our three entity types to Linked Data concept classes. For example, in the case of Freebase, our entity type "Person" (`pw:person`) is mapped to `http://rdf.freebase.com/ns/people/person`, and for each extracted *person* entity Freebase is queried for lexically matching concepts of the mapped type. Several candidate concepts may be selected for each entity (the number is given as a configuration parameter). These matches are added as disambiguation candidates to every ngram that corresponds to the original entity.

4.4 Disambiguation

The preceding processing steps use various techniques to determine relevant concepts in documents. A direct consequence is that a given ngram may be mapped to more than one concept: a taxonomy term, a Wikipedia article, a Freebase or a DBpedia concept. Although the Wikipedia Miner incorporates its own built-in disambiguation component, this merely ensures that at most one Wikipedia concept corresponds to each ngram. A second disambiguation step (Figure 2, step 4) determines whether concepts from *different* sources share the same meaning and whether their meaning is contextually relevant.

The disambiguation is performed for each document, one ngram at a time. If an ngram has a single concept mapping, it is considered unambiguous and this concept is added to the final taxonomy. If an ngram has multiple mappings, the conflicting concepts are inspected first. Here, we compare the context of the ngram with the contexts of each concept, as it is defined in its original source. The context of the ngram is as a set of labels of concepts that co-occur in the same document, whereas the context of each concept is a set of labels derived from its associated concepts, computed in a way that depends on the concept's origin. In SKOS taxonomies, associated concepts are determined via `skos:broader`, `skos:narrower`, and `skos:related` relations. For each associated concept we collect the `skos:prefLabel` and one or more `skos:altLabel`s. In Wikipedia, these labels are sourced from the article's redirects, its categories, the articles its abstract links to, and other linked articles whose semantic relatedness [15] exceeds a certain threshold (we used 0.3, which returns 27 linked articles on average). In the case of Freebase and DBpedia, we utilize the fact that many Freebase concepts have mappings to DBpedia, which in turn are (practically all) mapped to Wikipedia articles. We locate the corresponding Wikipedia article and use the above method to determine the concepts.

[9] A SPARQL endpoint is a web service that implements the W3C SPARQL protocol.

Once all related labels have been collected we calculate the distance between every pair of labels. To account for lexical variation between the labels, we use the Dice coefficient between the sets of bigrams that represent the labels. We then compute a final similarity score by averaging the distance over the top n scoring pairs. n is chosen as the size of the smaller set, because if the concepts the sets represent are truly identical, every label in the smaller set should have at least one reasonably similar partner in the other set; larger values of n tend to dilute the similarity score when one of the concepts has many weakly associated concept labels, which is often the case for Wikipedia concepts.

Given this similarity metric, disambiguation proceeds as follows. First, we choose the concept with the greatest similarity to the ngram's context to be the canonical concept. (This assumes that there is at least one correct concept among the conflicting ones.) Second, we compare the similarity of every other candidate concept to the canonical one and, depending on its similarity score s, list it as an `skos:exactMatch` (if $s > 0.9$), an `skos:closeMatch` (if $0.9 \geq s \geq 0.7$), or discard it (if $s < 0.7$). The thresholds were determined empirically.

As an example of disambiguation, the ngram *oceans* matches three concepts: *Ocean*, *Oceanography* (both Wikipedia articles), and *Marine areas* (a taxonomy concept). The first is chosen as the canonical concept because its similarity with the target document is greatest. *Marine areas* is added as `skos:closeMatch`, because its similarity with *Ocean* is 0.87. However, *Oceanography*'s similarity falls below 0.7, so it is discarded. As a another example, the ngram *logged* is matched to both *Logs* (a taxonomy concept) and *Deforestation* (a Wikipedia article). *Logs* is semantically connected to another taxonomy concept, which is why it was not discarded by the text analytics API, but it is discarded by the disambiguation step because it is not sufficiently closely related to other concepts that occur in the same document.

4.5 Consolidation

The final step is to unite all unambiguous and disambiguated concepts found in documents into a single taxonomy. Each concept lists several URIs under `skos:exactMatch` and (possibly) `skos:closeMatch` that define it in other sources: the input taxonomy, Wikipedia, Freebase and DBpedia. These sources already organize concepts into hierarchies, but they differ in structure. The challenge is to consolidate these hierarchies into a single taxonomy.

Sources of Relations. Taxonomies from related domains, as optional inputs, already define the relations we seek: `skos:broader` and `skos:narrower`. However, they may cover certain areas in more or less detail than what we need, which implies that some levels should be flattened while others are expanded. Because *broader* and *narrower* are transitive relations, flattening is straightforward. For expansion, concepts from other sources are needed.

Wikipedia places its articles into categories. For example, the article on George Washington belongs to 30 categories; some useful, e.g. *Presidents of the US*

and *US Army generals*, and others that are unlikely to be relevant in a taxonomy, e.g. *1732 births*. Some articles have corresponding categories (e.g., there is a category "George Washington"), which lead to further broader categories. Furthermore, names may indicate multiple relations (e.g. *Politicians of English descent* indicates that *George Washington* is both a *Politician* and *of English descent*). Wikipedia categories tend to be fine-grained, and we discard information to create broader concepts. We remove years (*1980s TV series* becomes *TV series*), country and language identifiers (*American sitcoms* becomes *Sitcoms*; *Italian-language comedy films* becomes *Comedy films*), and verb and prepositional phrases that modify a head noun (*Educational institutions established in the 1850s* becomes *Educational institutions*; *Musicians by country* becomes *Musicians*). The entire Wikipedia category structure is available on DBpedia in SKOS format, which makes it easy to navigate. We query the SPARQL DBpedia endpoint to determine categories for a given Wikipedia article.

Other potential sources are Freebase, where categories are defined by users, and DBpedia, which extracts relations from Wikipedia infoboxes. We plan to use this information in future when consolidating taxonomies.

Consolidation Rules. F-STEP consolidates the taxonomy that has been generated so far using a series of rules. First, direct relations are added between concepts. For each concept with a SKOS taxonomy URI, if its broader and narrower concepts match other input concepts, we connect these concepts, e.g. *Air transport* `skos:narrower` *Fear of flying*. If a concept has a Wikipedia URI and its immediate Wikipedia categories match an existing concept, we connect these concepts, e.g. *Green tea* `skos:narrower` *Pu-erh tea*.

Following the intuition that some concepts do not appear in the documents, but may useful for grouping others that do, we iteratively add such concepts. For each concept with a SKOS taxonomy URI, we use a transitive SPARQL query to check whether it can be connected by new intermediate concepts to other concepts. If a new concept is found, it is added to the taxonomy and its relations are populated for all further concepts. For example, this rule connects concepts like *Music* and *Punk rock* via a new concept *Music genres*, whereupon a further relation is added between *Music genres* and *Punk rock*.

Next, the Wikipedia categories are examined to identify those of interest. The document collection itself is used to quantify the degree of interest: categories whose various children co-occur in many documents tend to be more relevant. Specifically, a category's "quality" is computed by iterating over its children and checking how many documents contain them. If this score, normalized by the total number of comparisons made, exceeds a given threshold, the category is added to the output taxonomy. This helps eliminate categories that combine too many concepts (e.g. *Living people* in a news article) or that do not group co-occurring concepts, and singles out useful categories instead (e.g. *Seven Summits* might connect *Mont Blanc*, *Puncak Jaya*, *Aconcagua*, and *Mount Everest*). Next, we retrieve broader categories for these newly added categories and check whether their names match existing concepts, allowing us to add new relations.

One could continue up the Wikipedia category tree, but the resulting categories are less satisfactory. For example, *Music* belongs to *Sound*, which in turn belongs to *Hearing*, but the relation between *Music* and *Hearing* is associative rather than hierarchical. In fact, unlike conventional SKOS taxonomies, the Wikipedia category structure is not, in general, transitive.

Parentheses following some Wikipedia article names indicate possible groupings for a concept, e.g. *Madonna_(entertainer)* is placed under *Entertainers*, if such a concept exists. We also match each category name's last word against existing concept names, but choose only the most frequent concepts to reduce errors introduced by this crude technique.

We group all named entities that are found in Freebase using the Freebase categories, and all those found in DBpedia using the corresponding Wikipedia categories. The remainder are grouped by their type, e.g. *John Doe* under *Person*.

These techniques tend to produce forests of small subtrees, because general concepts rarely appear in documents. We check whether useful general terms can be found in a related taxonomy, and also examine the small upper-level taxonomy that a user may provide, as mentioned in Section 1. For example, a media website may divide news into *Business*, *Technology*, *Sport* and *Entertainment*, with more specific areas underneath, e.g. *Celebrities*, *Film*, *Music*—a two-level taxonomy of broad categories. For each input concept we retrieve its broadest concept—the one below the root—and add it, skipping intermediate levels. This rule adds relations like *Cooperation* `skos:broader` *Business and industry*.

Pruning Heuristics. Prunning can make a taxonomy more usable, and eliminate redundancies. First, following [4], who extract a taxonomy from WordNet, we elide parent–child links for single children. If a concept has a single child that itself has one or more children, we remove the child and point its children directly to its parent.

Second, we eliminate multiple inheritance that repeats information in the same taxonomy subtree, which originates from redundancy in the Wikipedia category structure. We identify cases where either relations or concepts can be removed without compromising the tree's informativeness. Figure 3 shows examples. In (a) the two-parent concept *Manchester United FC* is reduced to a single parent by removing a node that does not otherwise contribute to the structure. In (b) the two-parent concept *Tax* is reduced to a single parent by removing a small redundant subtree. In (c) a common parent of the two-parent concepts *The Notorious B.I.G.* and *Tupac Shakur* is pruned.

5 Evaluation and Discussion

Domain-specific taxonomies (and ontologies) are typically evaluated by (a) comparing them to manually-built taxonomies, (b) evaluating the accuracy of their concepts and relations, and (c) soliciting feedback from experts in the field. This section evaluates our system's ability to generate a taxonomy from a news

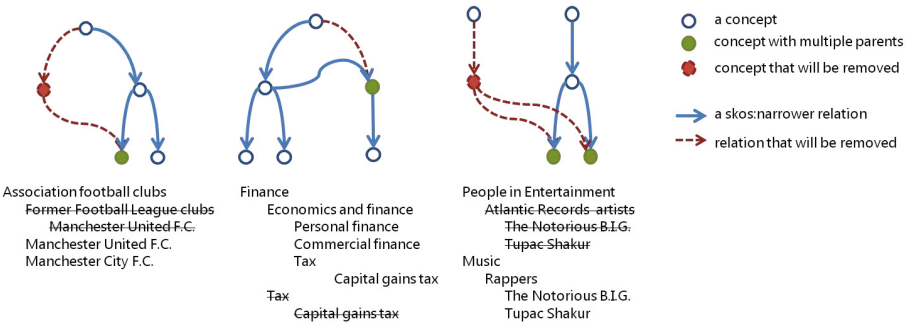


Fig. 3. Pruning concepts and relations to deal with multiple inheritance

collection. We give an overview of the dataset used, compare the dimensions of the taxonomy generated with other taxonomies, assess its coverage by comparing it with a hand-built taxonomy for the domain, and determine the accuracy of both its concepts and its relations with respect to human judgement.

5.1 The Domain

Fairfax Media is a large media organization that publishes hundreds of news articles daily. Currently, these are stored in a database, organized and retrieved according to manually assigned metadata. Manual assignment is time-consuming and error-prone, and automatically generated metadata, organized hierarchically for rapid access to news on a particular topic or in a general field, would be of great benefit.

We collected 2000 news articles (4.3MB of uncompressed text) from December 2011, averaging around 300 words each. We used the UK Integrated Public Service Sector vocabulary (`http://doc.esd.org.uk/IPSV/2.00.html`) as an input taxonomy. A taxonomy was extracted using the method described in Section 4 and can be viewed at `http://bit.ly/f-step`. It contains 10,150 concepts and 12,700 relations and is comparable in size to a manually-constructed taxonomy for news, the New York Times taxonomy (`data.nytimes.com`), which lists 10,400 *People*, *Organizations*, *Locations* and *Descriptors*. The average depth of the tree is 2.6, with some branches being 10 levels deep. Each concept appears in an average of 2 news articles. The most frequent, *New Zealand*, appears as metadata for 387 articles; the most topical, *Christmas*, is associated with 127 articles. About 400 concepts were added during the consolidation phase to group other concepts, and do not appear as metadata.

5.2 Coverage Comparison

To investigate the coverage of the automatically-generated taxonomy, we compared it with one comprising 458 concepts that Fairfax librarians had constructed

manually to cover all existing and future news articles. Interestingly, this taxonomy was never completed, most likely because of the labor involved. Omissions tend to be narrower concepts like individual sports, movie genres, music events, names of celebrities, and geographic locations. In order to evaluate our new taxonomy in terms of recall, we checked which of the 458 manually assigned concepts have labels that match labels in the new taxonomy (considering both preferred or alternative labels in both cases). There were a total of 271 such "true positives," yielding a recall of 59%. However, not all the manually assigned concepts are actually mentioned in the document set used to generate our taxonomy, and are therefore, by definition, irrelevant to it. We used Solr to seek concepts for which at least one preferred or alternative label appears in the document set, which reduced the original 458 concepts to 298 that are actually mentioned in the documents. Re-calculating the recall yields a figure of 75% (224 out of 298).

Inspection shows that some of the missing concepts are present but with different labels—instead of *Drunk*, the automatically generated taxonomy includes *Drinking alcohol* and *Alcohol use and abuse*. Others are present in a more specific form—instead of *Ethics* it lists *Ethical advertising* and *Development ethics*. Nevertheless, some important concepts are missing—for example, *Immigration*, *Laptop* and *Hospitality*.

5.3 Accuracy of Concepts

Fifteen human judges were used to evaluate the precision of the concepts present in the taxonomy generated from the documents. Each judge was presented with the text of a document and the taxonomy concepts associated with it, and asked to provide yes/no decisions on whether the document refers to each term. Five documents were chosen and given to all judges; a further 300 documents were distributed equally between the judges.

Looking first at the five common documents, the system extracted 5 to 30 concepts from each, with an average of 16. Three judges gave low scores, agreeing with only 74%, 86% and 90% of the concepts respectively, averaged over the five documents. The remaining 12 each agreed with virtually all—more than 97%—of the concepts identified by the system. The overall precision for automatic identification of concepts, averaged over all 15 judges, was 95.2%.

Before these figures were calculated the data was massaged slightly to remove an anomaly. It turned out that the system identified for each article the name of the newspaper in which it was published (e.g. *Taranaki Daily News*), but the human judges disagreed with one another on whether that should be counted as a valid concept for the article. A decision was taken to exclude the name of the newspaper from the first line of the article.

Turning now to the 300 documents that were examined by one judge each, the system identified a total of 3,347 concepts. Of these, 383 were judged incorrect, yielding an overall precision of 88.6%. (In 15 cases the judge was unwilling to give a yes/no answer; these were counted as incorrect.) Table 1 shows the source of the errors. Note that any given concept may originate in more than one source, which explains the discrepancy in the total of the Errors column (393, not 383).

Table 1. Sources of error in concept identification

Type	Number	Errors	Rate
People	1145	37	3.2%
Organizations	496	51	10.3%
Locations	988	114	11.5%
Wikipedia named entities	832	71	8.5%
Wikipedia other entities	99	16	16.4%
Taxonomy	868	229	26.4%
DBPedia	868	81	8.1%
Freebase	135	12	8.9%
Overall	3447	393	11.4%

The most accurate concepts are ones that describe people. The most error-prone ones emanate from the input taxonomy, 26% of which are incorrect. This taxonomy describes rather general concepts, which introduces more ambiguity than the other sources.

5.4 Accuracy of Relations

The same fifteen judges were used to evaluate the precision of the hierarchical relations present in the taxonomy. Each judge received 100 concept pairs and was asked for a yes/no decision as to whether that relation makes sense—i.e., whether the first concept really is narrower than the second. A total of 750 relations were examined, each adjudicated by two different judges.

The overall precision figure was 90%—that is, of the 1500 decisions, judges expressed disagreement in 150 cases. The interannotator agreement, calculated as the number of relationships that both judges agreed on expressed as a proportion of all relationships, was 87%.

An examination of where the two judges made different decisions revealed that some were too strict, or simply wrong (for example, *Acid* $\sqsubset$ *base chemistry*, *Leeds* $\sqsubset$ *North Yorkshire*, *History of Israel* $\sqsubset$ *Israel*, where $\sqsubset$ means "has parent"). Indeed, it appears that, according to some judges, polio is not an infectious disease and Sweden is not in Scandinavia! It is interesting to analyze the clear errors, discarding cases where the judges conflicted. Of the 25 situations where both judges agreed that the system was incorrect, ten pairs were related but not in a strict hierarchical sense (e.g., *Babies* $\not\sqsubset$ *school children*), four were due to an overly simplistic technique that we use to identify the head of a phrase (e.g. *Daily Mail* $\not\sqsubset$ *Mail*), two could have (and should have) been avoided (e.g. *League* $\not\sqsubset$ *League*), and nine were clearly incorrect and correspond to bugs that deserve further investigation (e.g. *Carter Observatory* $\not\sqsubset$ *City*).

6 Conclusions

This paper has presented a new approach to analyzing documents and generating taxonomies focused on their content. It combines existing tools with new

techniques for disambiguating concepts originating from various sources and consolidating them into a single hierarchy. A highlight of the scheme is that it can be easily extended. The use of RDF technology and modeling makes coupling and reconfiguring the individual components easy and flexible. The result, an SKOS taxonomy that is linked to both the documents and Linked Data sources, is a powerful knowledge organization structure that can serve many tasks: browsing documents, fueling facetted search refinements, question answering, finding similar documents, or simply analyzing one's document collection.

The evaluation has shown that in one particular scenario in the domain of news, the taxonomy that is generated is comparable to manually built exemplars in the dimensions of the hierarchical structure and in its coverage of the relevant concepts. Recall of 75% was achieved with respect to a manually generated taxonomy for the same domain, and inspection showed that some of the apparently missing concepts are present but with different—and arguably more precise—labels. With respect to multiple human judgements on five documents, the accuracy of concepts exceeded 95%; the figure decreased to 89% on a larger dataset of 300 documents. The accuracy of relations was measured at 90% with respect to human judgement, but this is diluted by human error. Analysis of cases where two judges agreed that the system was incorrect revealed that at least half were anomalies that could easily be rectified in a future version. Finally, although we still plan to perform an evaluation in an application context, initial feedback from professionals in the news domain is promising. Some professionals expect to tweak the taxonomy manually by renaming some top concepts, removing some irrelevant relations, or even re-grouping parts of the hierarchy, and we have designed a user interface that supports this.

Compared to the effort required to come up with a taxonomy manually, a cardinal advantage of the automated system is speed. Given 10,000 news articles, corresponding to one week's output of Fairfax Media, a fully-fledged taxonomy is generated in hours. Another advantage is that the taxonomy focuses on what actually appears in the documents. Only relevant concepts and relations are included, and relations are created based on salience in the documents (e.g. occurrence counts) rather than background knowledge. Finally, because Wikipedia and Freebase are updated daily by human editors, the taxonomy that is produced is current, which is important for ever-changing domains such as news.

Finally, the approach is applicable to any domain. Every knowledge-based organization deals with mountains of documents. Taxonomies are considered a very useful document management tool, but uptake has been been slow due to the effort involved in building and maintaining them. The scheme described in this paper reduces that cost significantly.

Acknowledgements. This work was co-funded by New Zealand's Ministry of Science and Innovation. We also thank David Milne and Shane Stuart from the University of Waikato and Reuben Schwarz from Fairfax Media NZ.

References

1. Caraballo, S.: Automatic construction of a hypernym-labeled noun hierarchy from text. In: Proc. of the 37th Annual Meeting of the ACL, pp. 120–126. ACL (1999)
2. Ponzetto, S., Strube, M.: Deriving a large scale taxonomy from wikipedia. In: Proc. of the 22nd National Conference on Artificial Intelligence, pp. 1440–1445. AAAI Press (2007)
3. Snow, R., Jurafsky, D., Ng, A.: Semantic taxonomy induction from heterogenous evidence. In: Proc. of the 21st Intl. Conf. on Computational Linguistics, pp. 801–808. ACL (2006)
4. Stoica, E., Hearst, M.A.: Automating creation of hierarchical faceted metadata structures. In: Procs. of the HLT/NAACL Conference (2007)
5. Dakka, W., Ipeirotis, P.: Automatic extraction of useful facet hierarchies from text databases. In: Proc. of the 24th IEEE Intl. Conf. on Data Engineering, pp. 466–475. IEEE (2008)
6. Sanderson, M., Croft, B.: Deriving concept hierarchies from text. In: Proc. of the 22nd Annual Intl. Conf. on R&D in Information Retrieval, pp. 206–213. ACM (1999)
7. Hearst, M.: Automatic acquisition of hyponyms from large text corpora. In: Proc. of the 14th Conference on Computational Linguistics, pp. 539–545. ACL (1992)
8. Suchanek, F., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proc. of the 16th Intl. Conference on World Wide Web, pp. 697–706. ACM (2007)
9. Wu, W., Li, H., Wang, H., Zhu, K.: Probase: A probabilistic taxonomy for text understanding. In: Proc. of the 2012 ACM Intl. Conf. on Management of Data, pp. 481–492. ACM (2012)
10. Matuszek, C., Witbrock, M., Kahlert, R., Cabral, J., Schneider, D., Shah, P., Lenat, D.: Searching for common sense: Populating cyc from the web. In: Proc. of the 20th Nat. Conf. on Artificial Intelligence, pp. 1430–1435. AAAI Press (2005)
11. Milne, D., Witten, I.: Learning to link with wikipedia. In: Proc. of the 17th Conference on Information and Knowledge Management, pp. 509–518. ACM (2008)
12. Mendes, P., Jakob, M., García-Silva, A., Bizer, C.: Dbpedia spotlight: shedding light on the web of documents. In: Proc. of the 7th Intl. Conf. on Semantic Systems, pp. 1–8. ACM (2011)
13. Augenstein, I., Padó, S., Rudolph, S.: LODifier: Generating Linked Data from Unstructured Text. In: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 210–224. Springer, Heidelberg (2012)
14. Broekstra, J., Kampman, A., van Harmelen, F.: Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In: Horrocks, I., Hendler, J. (eds.) ISWC 2002. LNCS, vol. 2342, pp. 54–68. Springer, Heidelberg (2002)
15. Milne, D., Witten, I.H.: An open-source toolkit for mining Wikipedia. Artificial Intelligence (2012)
16. Marrero, M., Sanchez-Cuadrado, S., Lara, J., Andreadakis, G.: Evaluation of Named Entity Extraction Systems. In: Proc. of the Conference on Intelligent Text Processing and Computational Linguistics, CICLing, pp. 47–58 (2009)
17. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proc. of Intl. Conf. on Management of Data, SIGMOD 2008, pp. 1247–1250. ACM, New York (2008)
18. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A nucleus for a web of open data. In: Aberer, K., et al. (eds.) ISWC/ASWC 2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007)

Semantic Multimedia Information Retrieval Based on Contextual Descriptions

Nadine Steinmetz and Harald Sack

Hasso Plattner Institute for Software Systems Engineering, Potsdam, Germany
{nadine.steinmetz,harald.sack}@hpi.uni-potsdam.de

Abstract. Semantic analysis and annotation of textual information with appropriate semantic entities is an essential task to enable content based search on the annotated data. For video resources textual information is rare at first sight. But in recent years the development of technologies for automatic extraction of textual information from audio visual content has advanced. Additionally, video portals allow videos to be annotated with tags and comments by authors as well as users. All this information taken together forms video metadata which is manyfold in various ways. By making use of the characteristics of the different metadata types context can be determined to enable sound and reliable semantic analysis and to support accuracy of understanding the video's content. This paper proposes a description model of video metadata for semantic analysis taking into account various contextual factors.

Keywords: context model, semantic analysis, video analysis, metadata analysis, named entity recognition.

1 Introduction

Context is an important factor that is mandatory for general understanding. Depending on the context, information might entail different meaning and thus, lead to different decisions. Context can be considered as the sum of available information items that put together enable unambiguous determination of the meaning of information.

For information retrieval and esp. for semantic and explorative search that take into account content-related information, it is of high importance to decide upon the various possible meaning of information. For semantic analysis, besides authoritative (textual) information supplied by experts also automatically extracted metadata or user-provided annotation contribute essential additional information about the content. However, metadata from different sources involve different characteristics and reliability.

Furthermore, due to the rich expressiveness of natural language textual information entails the problem of ambiguity. Thus, the word sense disambiguation of document metadata deserves special attention. The context needed for disambiguating ambiguous terms within a document is provided by all the surrounding information such as further metadata or textual content related to the same document or document fragment under consideration.

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 382–396, 2013.

The different characteristics of metadata items influence their confidence and relevance when applied as context items for the disambiguation process. So far, in computer science context is primarily discussed in the sense of *user context*. User context describes the situation of an interacting user. Here, context is used to solve a specific request in a personalized way, as e. g., in an ubiquitous computing scenario.

In this paper we present a context model that describes characteristics of metadata items. These metadata items may serve as context items for other metadata items according to their characteristics. Our context model includes a derived confidence value representing information about the anticipated ambiguity and correctness of the metadata item. This confidence value is applied to rank metadata items for a given context. Context determining metadata items, henceforth refered to as *context items*, support the subsequent semantic analysis process. As an application we apply our context model to support understanding of video metadata from various sources and improve the accuracy of semantic analysis.

The paper is organized as follows: Section 2 recapitulates related work in the field on context awareness and context definitions. The context model and a description of the identified contextual factors are presented in Section 3. In Section 4 the application of the context model within the semantic analysis process is presented. The proposed context model has been evaluated on the basis of an annotated dataset of video metadata. The evaluation results including the dataset are described in Section 5. Section 6 summarizes the achievements of this work and gives an outlook on future work.

2 Related Work

Recently, context and context-aware computing has received increasingly attention [10]. But the discussions about the influence and importance of context date far back into the past throughout various scientific fields of computer science. Mainly these discussions address the context a person is enclosed by. Therefore, characteristics of context are defined to solve personalization problems in e-commerce and ubiquitous computing, to identify life stages of a person for data mining, or to improve online marketing and management [1]. This context can be considered as user context. Although the received opinion agrees on the difficulties of defining context in general and finding a universal definition, the different disciplines identify certain characteristics for their field of interest. Lenat [7] states that for artificial intelligence context has been ignored or treated as black box for a long time. For the large knowledge base Cyc[1] he defined twelve dimensions of context to "specify the proper context in which an assertion (or question) should be stated". Bazire et. al collected 150 different definitions of context from different disciplines to identify the main components of context [2]. They concluded their study by determining all definitions to the parameters constraint, influence, behavior, nature, structure, and system. In ubiquitous

[1] `http://cyc.com/cyc/opencyc`

computing context is broadly used for two purposes: as retrieval cue and to tailor the behavior and the response type of the system [6]. Dourish has identified two different views on context: a representational and an interactional view and suggests the latter to be the more challenging for the field of interactive systems.

In 1931 Dewey wrote "We grasp the meaning of what is said in our language not because appreciation of context is unnecessary but because context is inescapably present." [5]. Although, this sentence addresses context in the field of psychology it is also valid for the characteristics of metadata as context items. Context is defined by the characteristics of the items included in it.

We utilize the characteristics of context items for semantic analysis, in particular for Named Entity Recognition (NER). In Natural Language Processing (NLP) the term NER refers to a method to find entities of specific types (persons, places, companies etc.) in a text. Similar to Word Sense Disambiguation (WSD) approaches we consider NER as the method to find specific entities with a unique meaning ("Berlin" as the German capital and not the town in Connecticut, U.S.) Mihalcea et. al published one of the first NER approaches using Wikipedia[2] URIs to identify specific entities [12]. This paper presents a combined approach of an analytical method comparing Wikipedia articles with contextual paragraphs and a machine-learning approach for the disambiguation process. Another machine-learning approach is presented in [3]. This approach uses different specific kernels in linear combination to disambiguate terms in a given text. The kernels are trained with surrounding words of an entity link within the paragraphs of the Wikipedia article. DBpedia Spotlight is an established NER application that applies an analytical approach for the disambiguation process. The context information of the text to be annotated is represented by a vector. Every entity candidate of a term[3] found in the text is represented as a vector composed of all terms that co-occurred within the same paragraphs of the Wikipedia articles where this entity is linked [11]. Recently, Damljanovic et. al presented an approach of combining the classic NER tagging (in terms of NLP) and entity disambiguation [4]. The terms the NER tagging tool identified as one of the expected categories (person, place, or organization) are assigned to DBpedia[4] classes. Entity candidates for this term are retrieved within the instances of the assigned ontology class.

All these NER approaches aim at the analysis of text documents. Context definitions are limited to merely structural characteristics such as word, sentence, paragraph, or full document [14]. We extend this context definition by determining further specific characteristics of the metadata items pertaining to a context.

3 Context and Contextual Factors

Documents are created within a specific user context determining the purpose the document was created for. This context can also be considered as pragmatics.

[2] `http://www.wikipedia.org`

[3] The authors use the expression "surface form" for a word or a word group representing an entity. Subsequently we use "term" synonymously to this definition.

[4] `http://dbpedia.org/About`

The metadata provided for the document as well as data automatically extracted from the document form a different context. This context determines the meaning of the given information. Therefore, we define:

Definition 1. A **context** is represented by a finite set C of context items. Each **context item** $ci \in C$ is a tuple $(term, uri, cd, c)$, where:

- $term$ denotes the value (string text) of the context item,
- uri denotes the list of (semantic) entities assigned to the $term$,
- cd denotes the contextual description $cd \in CD$ of the context item ci,
- $c \in [0...1]$ denotes the confidence value that is calculated according to cd.

Thereby we state that a context consists of context items. The context items derive from the metadata a document is provided with. The metadata items of a context belong to certain domains and thereby define the meaning of the textual information. In that way metadata items become context items.[5] Most of the metadata and automatically extracted information is provided in the form of natural language text. As already mentioned in the introduction natural language is expressive but entails the problem of ambiguity. To enable semantic annotation of documents and the documents' metadata the ambiguity of the textual information has to be removed.This is where the context comes into play. The characteristics of a context are determined by the items pertaining to it. But these context items originate from different sources, have different reliabilities and should therefore be weighted according to their significance within a context. We have defined a contextual description depicting the characteristics of these context items.

Definition 2. A **contextual description** $cd \in CD$ is a tuple (tt, st, sd, cl), where:

- $tt \in Tt$, where Tt is a finite set of text types,
- $st \in St$, where St is a finite set of source types,
- $sd \subseteq Sd$, where Sd is the set of available sources for the video,
- $cl \in Cl$, where Cl is a finite set of ontology classes,
- CD denotes the set of all contextual descriptions.

For our proposed use case, the semantic analysis of video metadata, we have restricted text types, sources, and ontology classes to the following sets:

- the set of text types Tt is determined to natural language text, keywords, and tags.
- the set of ontology classes Cl is determined to place, organization, and person.
- the set of source types St is determined to authoritative and non-authoritative sources, Automatic Speech Recognition (ASR), and Optical Character Recognition (OCR).

[5] Subsequently, we use the terms metadata item and context item synonymously.

Useful sources for automatically extracted textual information for video data are OCR and ASR algorithms. Usually few authoritative metadata is available as e. g., a title, speaker or primary persons, publisher etc. Additionally, some video resources are provided with textual, time-related tags by non-authorative sources[6]. Therefore we have restricted the set of available source types for video metadata St to these four sources.

Metadata from ASR and OCR sources, as well as the title and description from the authoritative metadata can be considered as natural language text. Information about the speaker or the publisher are usually given as keywords. Tags form a third text type as they are mostly given as a group of single words and only subsets of the group belong together (c.f. [8] for tag processing). Tt is therefore restricted to these three text types.

To determine appropriate entities for a given textual information it helps to know the prospective ontology class the entity belongs to. Some of the provided authoritative metadata can directly be assigned to ontology classes, as e. g., the metadata item for *speaker* can directly be assigned to the ontology class *Person*. For natural language processing Conditional Random Field (CRF) classifiers[7] are used to find entities of such ontology classes in fluent text. By using a 3-class model the ontology classes *Person*, *Place*, and *Organization* can be found in a text. Therefore the set Cl is restricted to these three ontology classes.

3.1 Detailed Contextual Description and Confidence Calculation

According to the contextual description the confidence of the context item is calculated. For each of the four contextual factors (tt, st, sd, and cl) a double precision value v is calculated, where $0 < v \leq 1.0$.

Source Reliability. The term *reliability* is referring to a prospective error rate concerning the source type st. Document metadata can either be created by human or computer agents. Human agents can be the author, who created the document or any user, who annotated the document with additional information. Computer agents are analysis algorithms, which extract (mostly) textual information from a multimedia document, such as OCR and ASR. All these agents provide information with different degrees of reliability. Where human agents in general can be considered more reliable than computer agents because of knowledge and experience, authoritative human agents are considered more reliable than non-authoritative human agents. According to this simple presumption the agents' reliability is ranked. The value v_{st} is set highest for authoritative ($v_{st} = 1.0$) and slightly lower for non-authoritative (human) sources ($v_{st} = 0.9$). As reliability values for computer agents we simply adopt the achieved

[6] Video portals like Yovisto (www.yovisto.com) allow the videos to be tagged by any user to make time-related references to the video.

[7] As used in the Stanford Named Entity Recognizer - `http://nlp.stanford.edu/software/CRF-NER.shtml`

evaluation results on precision for the considered analysis engines. Unfortunately, most video OCR evaluations base on single frame processing, which embellishes the results. Precision for video OCR on videos with equally text and non-text frames is still very low. According to [15], the error rate for news videos is up to 65%. Therefore we assume a worst case precision of 35% ($v_{st} = 0.35$) for context items with an OCR analysis as source agent. Word error rates for ASR analysis engines range between 10% and 50% (respectively an accuracy rate between 50% and 90%)[13]. We assume the worst case and determine the reliability value for context items from ASR results to $v_{st} = 0.5$.

Source Diversity. Source diversity specifies how many of the available annotation sources agree on the same metadata item. The diversity ranges from a single source to all available sources. The more sources agree on the value of a context item the more reliable the item is considered. Depending on the available sources (Sd) and the set of sources that agree on the same item i (s_i), the value for the source diversity v_{sd} is calculated as follows:

$$v_{sd} = \frac{|s_i|}{|Sd|}$$

Example: The text "computer" is automatically extracted by OCR analysis from a video frame. The title of the video is "The birth of the computer". For this video the only sources of textual information are the authoritative metadata and the extracted texts by OCR. In this case $v_{sd} = \frac{2}{2} = 1.0$ for context items having $term = computer$ as the term "computer" is confirmed by both available sources.

Text Type. According to the source of the metadata item the general type of the context item's values differ. Authoritative information of a document as e. g. the creator, production location, or keywords have key terms as values. These key terms usually in total depict an entity. Further authoritative textual information, such as the title or a descriptive text are given as running text in natural language. A third text type are typed literals, as e. g., "print run = 1.000 copies". It is assumed that the ambiguity of metadata items with text type 'typed literal' is lowest. Therefore the according confidence value is highest with $v_{tt} = 1.0$. But usually this text type is not representative for video metadata. The ambiguity of running text depends on the precision of the NLP algorithm used to extract key terms. We are using the Stanford POS tagger[8] to identify word types in text. This tagger has an accuracy rate of 56% per sentence[9], which leads to $v_{tt} = 0.56$. By using this rate as reliability value for running text we have a measure independent from text length. POS tagging is not needed for context items that are given as key terms. But still, to allow an uncertainty we determine the reliability of key terms slightly lower than for typed literals as $v_{tt} = 0.9$.

[8] `http://nlp.stanford.edu/software/tagger.shtml`

Class Cardinality. The contextual factor of class cardinality corresponds to the number of instances the assigned ontology class contains. In general a descriptive text does not refer to a specific ontology class, if a CRF classifier does not find any classes in the text. The entities found in such a text can be of any type. In that case the context items found in this natural language text are assigned to the most general class, $\top$ class of the ontology[9] and the class cardinality is highest. According to the ontology class cl assigned to the metadata item and its known cardinality the value v_{cl} is calculated proportional to the overall number of all known entities ($|\top|$), where $\top$ denotes the most general class containing all individuals of the knowledge base, and $|\top|$ denotes the number of all instances pertaining to this class. A high class cardinality entails a high ambiguity. Therefore, the value v_{cl} is inverted to reflect a reverse proportionality regarding the amount of the value and the ambiguity:

$$v_{cl} = 1 - \frac{|cl|}{|\top|}$$

Example: A context item of a video might be identified as Person (by uploading author or by an automatic NER tagging tool). Using the DBpedia Version 3.8.0 as knowledge base, the class "Person" contains 763,644 instances. owl:Thing as top class of the DBpedia ontology holds 2,350,907 instances. Accordingly, the confidence value $v_{cl} = 1 - \frac{763{,}644}{2{,}350{,}907} = 0.67$ for a context item assigned to the DBpedia ontology class "Person".

The number of entity candidates of a term can also be a measure for the prospective ambiguity of the term. However, evaluations showed better results for the approach on class cardinality. Details on the evaluation results are described in Section 5.

After calculating each confidence value for the four constituents of the contextual description the total confidence value for a context item calculates as follows:

$$c = \frac{v_c + v_{sd} + v_{sr} + v_{tt}}{4}$$

3.2 Exemplary Confidence Calculation for Context Items

Let an example video have the following authoritative metadata information:

- Title: "The birth of the computer."
- Speaker: "George Dyson"
- Publisher: "TED"

Additionally, "computer" and "alamogordo" were extracted from the video via OCR analysis.

Speaker and publisher information are considered as keywords. The title and the OCR texts are considered as natural language text. Speaker is assigned to

[9] Which means, all entities of the knowledge base have to be considered and the amount cannot be restricted to a certain class.

Table 1. Example values for contextual factors and the according confidence

term	*tt*	v_{tt}	*cl*	v_{cl}	*st*	v_st	*sd*	v_{sd}	*c*
TED	keyword	0.9	Organiz.	0.96	auth.	1.0	auth.	0.5	0.84
George Dyson	keyword	0.9	Person	0.85	auth.	1.0	auth.	0.5	0.81
computer	nat. language	0.56	$\top$	0.0	auth.	1.0	auth., OCR	1.0	0.64
birth	nat. language	0.56	$\top$	0.0	auth.	1.0	auth.	0.5	0.52
computer	nat. language	0.56	$\top$	0.0	OCR	0.35	auth., OCR	1.0	0.48
alamogordo	nat. language	0.56	$\top$	0.0	OCR	0.35	OCR	0.5	0.35

the DBpedia ontology class "Person" and publisher is assigned to the DBpedia ontology class "Organization". The NER tagger did not find any class types in the title or the OCR information. After NLP pre-processing six context items are generated from the given metadata. The contextual factors and the calculated confidence value of the six context items are shown in Table 1.

3.3 Context Items Views

As shown in Figure 1, the identified contextual factors and dimensions influence different superordinate characteristics and can be aggregated in two different views: the confidence view and the relevance view on context items. The confidence view aggregates the characteristics of a context item described above. But context items also have characteristics regarding their context relevance within the video.

Confidence View. The **correctness** of a context item is influenced by the source diversity as well as by the source reliability. The more sources agree on an item and the higher the reliability of the item's source is, the higher is the reliability that this item is correct. The **ambiguity** of a context item is influenced by the text type and the class cardinality assigned to the item. Natural language text needs NLP technologies to identify important key terms. Due to the possible number of potential errors the ambiguity of natural language text is considered higher than for simple restricted key terms. For key terms no further processing is needed. Also, the lower the amount of instances of the assigned ontology class the lower is the item's potential ambiguity.

Both, ambiguity and correctness influence the confidence of a context item. With the term confidence we aim at the trust level we assign to the item for further analysis steps. A high correctness rate and a low ambiguity rate entail a high confidence for the context item. The confidence view is used to order context items according to their correctness and ambiguity. The higher the confidence the higher is the probability that the context item is analyzed correctly and the accurate entity is assigned to the item.

Relevance View. The spatial, temporal, and social dimension specify the relevance of a context item in relation to other context items of the document.

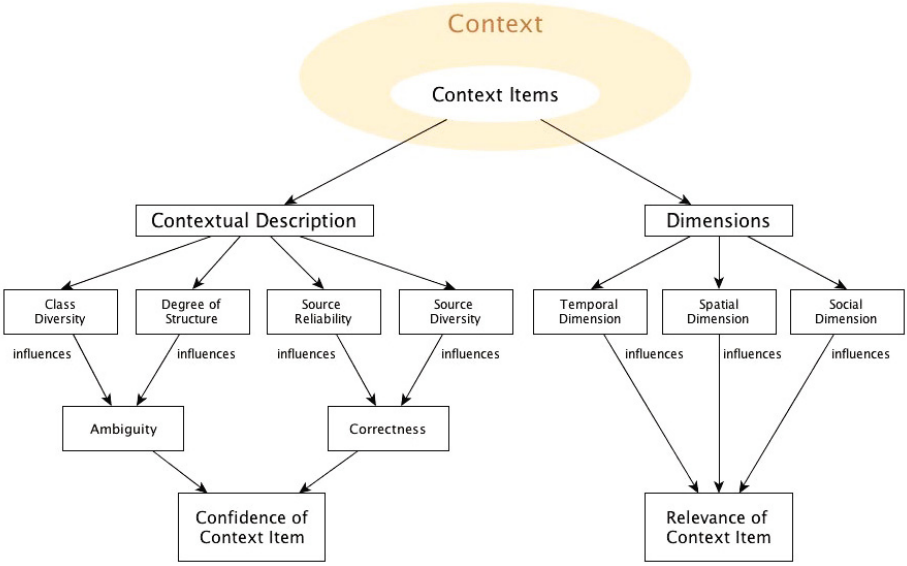


Fig. 1. Contextual factors of context items

With the help of the dimensions the divergence of the context items w.r.t. the document's content can be identified. By creating a context for a semantic analysis of context items the relevance view is important to aggregate the amount of all context items related to a document to smaller groups of stronger content-related coherence. In this way the semantic analysis of context items can be performed within more accurate and therefore also more meaningful contexts.

Metadata items of time referenced documents, such as video or audio files can be assigned to document fragments or the full document. The **temporal dimension** reflects the reference period of the item. The values of this dimension have a range between the smallest unit of the document (e. g., a frame for a video) and the full document. The **spatial dimension** assigns the metadata item to a specific region respectively to the entire document. E. g., for a video document the values starts with a single pixel within a frame over "geometrically determined region within a frame" to the full frame. The **social dimension** plays a special role within the characteristics of context items. It takes into account information about social relationships of the user who created the metadata item as well as the user, who accesses the document. Therefore, this dimension is dependent of the user and covers a personal perspective.

4 Using Context for Semantic Analysis

We apply the proposed context description model to the semantic analysis of video metadata and the annotation of textual information with semantic entities.

Subsequently, we will refer to our semantic analyzing engine as *conTagger*. The semantic analysis of metadata items of a video consists of three main steps:

- Collecting metadata items and defining contextual description
- Calculating the confidence value and sorting the list of metadata items according to their confidence
- Disambiguating every item using dynamically created context

4.1 Semantic Analysis Based on Ranked Context Items

The conTagger disambiguates context items based on term-entity co-occurrence as well as on the Wikipedia link graph. According to the degree of integration of the context item to be disambiguated within the context (which is a list of context items) the highest ranked entity candidate is chosen[10]. A context item initially includes a list of entity candidates for the item's textual term - if the term is ambiguous. After the disambiguation process the entity candidates are replaced by the resulting entity. The resulting entity features a disambiguation score. This disambiguation score has a range of [0.0 ... 1.0] and represents a trust value of the disambiguation process. The higher the value the higher the probability of a correct disambiguation. If an already disambiguated context item is added to influence the disambiguation of another item, the assigned entity is used as a fix point for the context creation. Otherwise the entire list of all entity candidates of the context item is used for the disambiguation process. Non-ambiguous context items initially contain only one entity featuring a disambiguation score of 1.0. Subject to these conditions the following hypothesis is put forward:

Hypothesis 1. The disambiguation results of context items are improved, if context items with higher confidences are disambiguated first.

4.2 Dynamically Creating Context for Disambiguation

The context of a context item determines the meaning of ambiguous textual information of the item to a single entity. The more specific the context the higher the probability of a correct disambiguation. Usually documents of any type are structured according to content-related segments. The more segments are aggregated as context the more general the contextual information is considered. Ambiguous textual information is hard to be disambiguated using a rather general context, because a general context probably contains more heterogenous information. Thus, the document should be segmented into fragments of coherent content to be able to create more accurate contexts. Considering this presumption the following hypothesis is put forward:

Hypothesis 2. The context of context items within a document should be restricted to segments of coherent content.

[10] For more information on the disambiguation process, please cf. [8].

Table 2. Evaluation of Hypothesis 2

	ASR		OCR		Tags	
	Recall	Precision	Recall	Precision	Recall	Precision
conTagger, Segment-Based	**55.0**	**61.0**	**56.0**	**24.0**	**71.0**	**69.5**
conTagger, Video-Based	53.0	46.0	51.0	21.0	69.0	68.0

Following hypotheses 1 and 2 the context for the disambiguation of each item is created dynamically. Only context items of the same segment and with a defined minimum confidence value are added to the context and thereby influence the disambiguation process. The context items from authoritative metadata are added as context for the disambiguation of all time-related context items – but also only if their confidence value exceeds a certain threshold. This threshold can be set dynamically for each context item type and is discussed in Section 5. The same applies for the disambiguation score of a disambiguated context item. For the dynamic context creation the score has to exceed a defined threshold. This threshold is also discussed in Section 5. By using thresholds for confidence value and disambiguation score the precision of the disambiguation process is aimed to be high without decreasing the recall. Using the dynamically created context each context item is disambiguated and the highest ranked entity is assigned as determining entity for the textual information instead of the list of entity candidates. Analysis and evaluation results for hypotheses 1 and 2 are discussed in the following section.

5 Evaluation

To evaluate NER algorithms a ground truth consisting of a text and a list of correct entities assigned to terms in this text is needed. Few datasets of simple texts and according entities are available in order to compare different NER algorithms. The creation of such a dataset is costly and time-consuming. Mendes et. al published such a dataset for the evaluation of the NER tool *DBpedia Spotlight* [11].

For the evaluation of *conTagger* a dataset consisting of different types of video metadata including the correct entities assigned to all the available textual information is needed. As far as we know, no dataset of that structure and for that purpose is available. Therefore, we have created a dataset of annotated video metadata in order to be able to evaluate our approach.

The evaluation dataset consists of metadata from five videos. The videos are live recordings of TED[11] conference talks covering the topics physics, biology, psychology, sociology, and history science. The metadata for each video consist of authoritative metadata (including title, speaker, providing organization, subject, keywords, descriptive text, and a Wikipedia text corresponding to the speaker), user-generated tags, and automatically extracted text from OCR and ASR.

[11] `http://www.ted.com`

The videos have been partitioned into content-related video segments via automatic scene cut detection. The time-related metadata (tags, ASR, and OCR) is assigned to the related video segments. Overall the dataset consists of 822 metadata items, where an item can be a key term or fluent text consisting of up to almost 1000 words[12].

Table 3. Evaluation results (R – Recall, P – Precision, and F_1-Measure) of the *conTagger* compared to simple segment-based NER, DBpedia Spotlight and the Wiki Machine

	conTagger			*Simple NER*			*Wiki Machine*			*Spotlight*		
	R	P	F_1	R	P	F_1	R	P	F_1	R	P	F_1
Authorative	**60.0**	54.5	57.0	52.0	46.0	49.0	59.5	**56.5**	**58.0**	50.0	44.0	47.0
Tags	**71.0**	**69.5**	**70.0**	61.0	60.0	60.5	44.0	62.0	51.5	60.0	59.0	59.5
ASR	55.0	**61.0**	**58.0**	56.5	38.0	45.5	**61.5**	50.0	55.0	56.0	34.0	42.5
OCR	**56.0**	**24.0**	**34.0**	44.0	17.5	25.0	24.5	18.0	21.0	47.0	18.0	26.0
Segments	54.0	**58.0**	**56.0**	57.0	39.0	46.5	57.0	49.0	52.5	**59.0**	35.0	43.5
Video	56.0	**48.0**	**52.0**	57.0	30.0	39.5	**58.0**	43.0	49.5	54.0	31.0	39.5

For evaluating Hypothesis 2 the context items were disambiguated using the entire video as context as well as for only context items of the same segments for comparison. Evaluation results are shown in Table 2. Recall values state how many of the entities of the ground truth are found by the respective analysis approach. Precision states how many of the extracted entities are present in the ground truth. As anticipated, the disambiguation results are improved using content based segments as context. Especially results for ASR metadata items differ in recall and precision for both variants. This probably follows from the fact that in our dataset there are much more ASR metadata items and because speech usually comprehends wider spread content in terms of context information. However, recall and precision are not significantly different, which results from the homogeneous character of the single videos of our dataset and their video segments.

To evaluate the *conTagger* regarding Hypothesis 1 we have compared the evaluation results to our own simple segment-based NER, NER by DBpedia Spotlight[11] and NER by the Wiki Machine[3]. For the analysis of the video metadata using DBpedia Spotlight, all metadata items assigned to a video segment – constituting a context – have been processed together via the Spotlight Webservice. The Wiki Machine results are achieved by disambiguating each metadata item on its own.

The evaluation results according to the different sources as well as video and segment-based evaluation are depicted in Table 3. The results are aggregated according to different sources and different relevance views. For the different sources the recall and precision values are calculated per video and averaged

[12] For downloading the dataset and the ground truth please cf. the readme file at `http://tinyurl.com/cztyayu`

over all five videos. For segments the recall and precision values are calculated for every segment over all sources and averaged over all segments. The evaluation results for videos are calculated respectively. Most notably, the *conTagger* achieves significantly good results on the metadata items with lower confidence, as OCR and ASR results. The overall evaluation of annotated entities per segment and video confirms the very good results. *ConTagger* achieves very good precision and F_1-measure results compared to the other NER approaches. As described in Section 3.1 the ambiguity of a context item can be defined by the number of entity candidates. We evaluated the disambiguation process using the inverted normalized number of entity candidates instead of the class cardinality measure. Better evaluation results were achieved by using the class cardinality. F_1-measures for all source types were lower at an average of 5% when using the ambiguity measure based on entity candidates. Obviously a low number of entity candidates does not necessarily mean that the correct entity is amongst the few candidates. Therefore, the ambiguity measure is set according to class cardinality of an assigned class.

For the dynamic context creation we have processed exhaustive test runs to determine the best suited thresholds for the confidence value and the disambiguation score when adding items to the context for a disambiguation process. The values for both parameters range between 0 and 1. Therefore, the context creation and subsequent disambiguation process has been performed with all combinations of confidence and disambiguation score values increasing the parameters in steps of 0.05, resulting in 441 runs. Subsequently, the parameters settings achieving the best recall and precision values aggregated over different source types have been identified. The best recall and precision results for metadata items from OCR and ASR analysis (featuring lowest confidence values) are achieved by creating the context from context items with a minimum confidence value of $c = 0.7$. Authoritative metadata items (featuring highest confidence values) are disambiguated using context items with highest confidence values in any case, because no time-referenced items must be used. Therefore the identified minimum threshold for the dynamic context creation is comparatively low with $c = 0.25$. The minimum threshold for the disambiguation of time-referenced tags is determined mid-range. This means that some of the other time-referenced metadata items (from OCR or ASR analysis) are used as context items, but not all of them as the lowest calculated confidence value for time-referenced metadata was calculated with $c = 0.285$. Apparently the disambiguation score is not as important as the confidence value for the context items used as influential items for a disambiguation process.

These evaluation results support our premise that the characteristics and the use of contextual factors of different metadata items support the semantic analysis process.

6 Ongoing and Future Work

Major contributions of our work include the definition of contextual information of video metadata for the purpose of NER and calculating a confidence value.

This value is used to bring metadata items in a specific order and to use them as context items for the disambiguation process. Based on this information and the temporal, spatial and social dimension metadata items influence the results of semantic analysis as context items. Current NER approaches miss to identify specific characteristics of document metadata. We have presented an extensible context description model that determines the important facts of document metadata items in a context. The characteristics of the context items are exemplary applied for semantic analysis on video metadata. Moreover, the context model is also applicable to any document type where metadata is harvested from different sources.

Ongoing and future work concentrates on the further refinement of a context. The social dimension plays an important role from the users' perspective of metadata. Metadata endorsed by friends or colleagues can be helpful for the user as additional descriptive information. This dimension also can be used to represent the pragmatics of a user when editing or creating metadata. In this way the context might change over time.

Future work includes the consideration of the influence of this additional context dimension on the context model and its application. A context can also further be refined by sample low level adjustments as white and black lists. Whitelisting can either be achieved statically by applying a specific knowledge base that only "knows" relevant entities and reduces the ambiguity of terms or by logical constraining rules. E. g., a document produced in 1960 does most likely only reference persons born before this date. So only a constrained number of entities qualifies for the analysis of this document. While persons naturally have a time reference, other real world entities may be hard to classify. Ongoing work includes the definition of a time-related scope for various entity types. Blacklisting on the other hand disqualifies particular entities for the analysis process. This can also either be achieved manually or automatically. Automatic blacklisting can be achieved by adding the previously deselected entity candidates (those that were not selected by disambiguation) of a disambiguated context item to a context restriction. With every disambiguated context item this "negative context" grows and a dynamic blacklist is achieved. Entity candidates related to this negative context will receive a penalty.

With the presented work we point out the importance of contextual factors of metadata. The proposed context model enables the characterization of metadata items from different sources and of various structure. By using the example of video metadata we were able to show how to support the (automatic) comprehension of a document's content with the help of its metadata.

References

1. Adomavicius, G., Tuzhilin, A.: Context-aware recommender systems. In: Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys 2008, pp. 335–336. ACM, New York (2008)

2. Bazire, M., Brézillon, P.: Understanding context before using it. In: Dey, A.K., Kokinov, B., Leake, D.B., Turner, R. (eds.) CONTEXT 2005. LNCS (LNAI), vol. 3554, pp. 29–40. Springer, Heidelberg (2005)
3. Bryl, V., Giuliano, C., Serafini, L., Tymoshenko, K.: Supporting natural language processing with background knowledge: Coreference resolution case. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010, Part I. LNCS, vol. 6496, pp. 80–95. Springer, Heidelberg (2010)
4. Damljanovic, D., Bontcheva, K.: Named entity disambiguation using linked data. In: 9th Extended Semantic Web Conference (ESWC 2012) (May 2012)
5. Dewey, J.: Context and thought, vol. 12. University of California publications in philosophy (1931)
6. Dourish, P.: What we talk about when we talk about context. Personal Ubiquitous Comput. 8(1), 19–30 (2004)
7. Lenat, D.: The dimensions of context-space. Technical report, Cycorp (1998)
8. Ludwig, N., Sack, H.: Named entity recognition for user-generated tags. In: Proceedings of the 2011 22nd International Workshop on Database and Expert Systems Applications, DEXA 2011, pp. 177–181. IEEE Computer Society, Washington, DC (2011)
9. Manning, C.D.: Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In: Gelbukh, A.F. (ed.) CICLing 2011, Part I. LNCS, vol. 6608, pp. 171–189. Springer, Heidelberg (2011)
10. Mehra, P.: Context-aware computing: Beyond search and location-based services. IEEE Internet Computing 16, 12–16 (2012)
11. Mendes, P.N., Jacob, M., García-Silva, A., Bizer, C.: Dbpedia spotlight: Shedding light on the web of documents. In: Proc. of 7th Int. Conf. on Semantic Systems, Graz, Austria, September 7-9 (2011)
12. Mihalcea, R., Csomai, A.: Wikify!: Linking documents to encyclopedic knowledge. In: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, CIKM 2007, pp. 233–242. ACM, New York (2007)
13. Mostefa Djamel, C.K., Olivier, H.: Evaluation of Automatic Speech Recognition and Speech Language Translation within TC-STAR: Results from the first evaluation campaign. In: Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006), Genoa, Italy. ELRA (May 2006)
14. Sen, P.: Collective context-aware topic models for entity disambiguation. In: Proceedings of the 21st International Conference on World Wide Web, WWW 2012, pp. 729–738. ACM, New York (2012)
15. Wactlar, H.D., Hauptmann, A.G., Christel, M.G., Houghton, R.A., Olligschlaeger, A.M.: Complementary video and audio analysis for broadcast news archives. Commun. ACM 43(2), 42–47 (2000)

Automatic Expansion of DBpedia Exploiting Wikipedia Cross-Language Information

Alessio Palmero Aprosio[1,2], Claudio Giuliano[1], and Alberto Lavelli[1]

[1] Fondazione Bruno Kessler, Via Sommarive 18, I-38123 Trento
{aprosio,giuliano,lavelli}@fbk.eu
[2] Università degli Studi di Milano, Via Comelico 39/41, I-20135 Milano

Abstract. DBpedia is a project aiming to represent Wikipedia content in RDF triples. It plays a central role in the Semantic Web, due to the large and growing number of resources linked to it. Nowadays, only 1.7M Wikipedia pages are deeply classified in the DBpedia ontology, although the English Wikipedia contains almost 4M pages, showing a clear problem of coverage. In other languages (like French and Spanish) this coverage is even lower. The objective of this paper is to define a methodology to increase the coverage of DBpedia in different languages. The major problems that we have to solve concern the high number of classes involved in the DBpedia ontology and the lack of coverage for some classes in certain languages. In order to deal with these problems, we first extend the population of the classes for the different languages by connecting the corresponding Wikipedia pages through cross-language links. Then, we train a supervised classifier using this extended set as training data. We evaluated our system using a manually annotated test set, demonstrating that our approach can add more than 1M new entities to DBpedia with high precision (90%) and recall (50%). The resulting resource is available through a SPARQL endpoint and a downloadable package.

1 Introduction

The need of structured information from the Web has led to the release of several large-scale knowledge bases (KB) in the last years. Most of them have been populated using Wikipedia as primary data source. The online encyclopedia represents a practical choice, as it is freely available, big enough to cover a large part of human knowledge, and populated by about 100,000 active contributors, therefore the information it contains represents a good approximation of what people need and wish to know. Some relevant examples include FreeBase,[1] DBpedia,[2] and Yago,[3] created using various techniques that range from crowd sourcing to handcrafted rules.

[1] http://www.freebase.com/
[2] http://dbpedia.org/About
[3] http://www.mpi-inf.mpg.de/yago-naga/yago/

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 397–411, 2013.

We are particularly interested in DBpedia as it plays a central role in the development of the Semantic Web. The large and growing number of resources linked to it makes DBpedia one of the central interlinking hubs of the emerging Web of Data. First, the DBpedia project develops and mantains an ontology, available for download in OWL format. Then, this ontology is populated using a rule-based semi-automatic approach that relies on Wikipedia *infoboxes*, a set of *subject-attribute-value* triples that represents a summary of some unifying aspect that the Wikipedia articles share. For example, biographical articles typically have a specific infobox (`Persondata` in the English Wikipedia) containing information such as *name*, *date of birth*, *nationality*, *activity*, etc. Specifically, the DBpedia project releases an extraction framework used to extract the structured information contained in the infoboxes and to convert it in triples. Moreover, crowd sourcing is used to map infoboxes and infobox attributes to the classes and properties of the DBpedia ontology, respectively. Finally, if an infobox is mapped to a DBpedia class, all Wikipedia articles containing such infobox are added to the class. As the number of required mappings is extremely large, the whole process follows an approach based on the frequency of the infoboxes and infobox attributes. Most frequent items are mapped first. This guarantees a good coverage because infoboxes are distributed according the Zipf's law. Therefore, despite the number of mappings is small, a large number of articles have been added to the ontology. At the time of starting the experiments, there are 360 mappings available for the English DBpedia, covering around 1.7M entities, against almost 4M articles in Wikipedia. The remaining pages are automatically mapped to the trivial top-level class `owl:Thing`. Hereafter, when we speak about coverage, we will always refer to classes different from `owl:Thing`. The Italian chapter has only 50 mappings, but covering more than 600K pages (out of around 1M articles in the corresponding Wikipedia), because some infoboxes cover highy populated classes, like `Person` and `Place`. The French and Spanish chapters, differently, contain around 15K pages each, with 70 and 100 mappings respectively. Finally, the resulting KB is made available as Linked Data,[4] and via DBpedia's main SPARQL endpoint.[5]

Unfortunately, there is a lot of variability in the names used for infoboxes and infobox attributes. Thus, it often happens that two or more infoboxes might be mapped to the same class, but none of them is included in DBpedia because their individual frequency is too small. Moreover, the DBpedia ontology often has classes that do not have a corresponding Wikipedia infobox. For example, the class `Actor` does not have a generic infobox in the English Wikipedia. However, Wikipedia provides some very specific infoboxes mapped to subclasses of `Actor`, such as `Chinese-language_singer_and_actor`. In this way, Bruce Lee is present in the database as an `Actor`, while other very famous actors like Clint Eastwood and Brad Pitt are not, clearly an undesirable result. Finally, some articles do not have an infobox, even if Wikipedia provides one for the purpose. This may

[4] `http://wiki.dbpedia.org/Downloads`
[5] `http://dbpedia.org/sparql`

happen because the user who writes that article does not know how to specify it, or simply does not know that infoboxes exist.

At the early stages of the project, the construction of DBpedia was solely based on the English Wikipedia. More recently, other contributors around the world have joined the project to create localized and interconnected versions of the resource. The goal is to populate the same ontology used in the English project, using articles from editions of Wikipedia in different languages. At the time of writing, there are 16 different localized versions of DBpedia. The inclusion of more languages has widened the problem of coverage. As each edition of Wikipedia is managed by different groups of volunteers with different guidelines, the DBpedia leading idea to semi-automatically populate the ontology by mapping infoboxes to classes does not work properly in some cases. For example, in the Italian DBpedia, the `Cyclist` category is empty, simply because the Italian edition of Wikipedia has a more generic `Sportivo` (sportsman) infobox, evidently considered adequate by the Italian contributors. This is convenient because one can assign a lot of pages to a class with only a single mapping, but cannot identify a more specific class. Besides the `Cyclist` class, also `Actor`, `Writer` and `Scientist` are empty in the Italian DBpedia, for the same reason. Other languages have similar problems: there are no entities for `Politician` in French and German, for `Plant` in Spanish, and so on.

In this paper, we address the problem of populating the DBpedia ontology, that has 359 classes. We propose an automatic two-stage approach that exploits Wikipedia cross-language links to extend the DBpedia coverage in different languages. First, the cross-language links are used to add Wikipedia articles not present in the DBpedia for one language but present in others. In the above example, those cyclists in the Italian Wikipedia having a cross-language link to an English article already present in the English DBpedia can be automatically added to the Italian DBpedia. Thanks to this first step, we increased the DBpedia coverage on Wikipedia articles by around 60% on the six languages considered in our experiments (English, Italian, German, French, Spanish, and Portuguese). The relative error of cross-lingual links in Wikipedia is very small, so we asses that the precision of the first phase is almost 100% [11].

Second, we further boost the coverage by training a supervised kernel-based classifier using both the articles already present in DBpedia and the ones extracted in the first stage, and then classify those articles for which cross-language links do not exist. Experiments have been performed on a dataset of 400 articles manually annotated by the authors. Starting from 5.6M total entities extracted from Wikipedia in the six languages, around 2.2M are added using the first step. We show that our approach further increases the coverage of the DBpedia with high accuracy. Our algorithm can be tuned to have different tradeoffs between precision and recall. The resulting resource contains a total of nearly 4M entities, 1.7M of them not included in the original DBpedia for the six languages considered for the experiment.

2 Entity Representation

The goal of our research is to assign novel entities to DBpedia classes requiring no additional human supervision. Specifically, we consider those entities not already present in DBpedia for which there exists at least a Wikipedia article, no matter in which language. The ontology population task is cast as a machine-learning classification problem, where entities already present in DBpedia (again, no matter in which language the corresponding Wikipedia articles are available) are used to train a state-of-the-art classifier that assigns novel entities to the most specific class in the DBpedia ontology.

Our approach exploits the Wikipedia cross-language links to represent each entity with features extracted from the corresponding articles in different languages. This novel contribution is supported by the observation that different Wikipedia communities tend to structure the articles in a slightly different way. As already reported in Section 1, English and Italian Wikipedia have an infobox for biographies (`PersonData` and `Bio`, respectively), while Spanish and French do not. DBpedia offers the triple set of cross-language links, but the information stored in one language is not automatically trasferred on other ones.

Formally, we proceed as follows to automatically derive the set of entities $\mathcal{E}$, also used to build the training set. Let $\mathcal{L}$ be the set of languages available in Wikipedia, we first build a matrix E where the i-th row represents an entity $e_i \in \mathcal{E}$ and j-th column refers to the corresponding language $l_j \in \mathcal{L}$. The cross-language links are used to automatically align on the same row all Wikipedia articles that describe the same entity. The element $E_{i,j}$ of this matrix is *null* if a Wikipedia article describing the entity e_i does not exist in l_j. An instance in our machine learning problem is therefore represented as a row vector e_i where each j-th element is a Wikipedia article in language l_j. Figure 1 shows a portion of the entity matrix.

en	de	it	...
Xolile Yawa	Xolile Yawa	*null*	...
The Locket	*null*	Il segreto del medaglione	...
Barack Obama	Barack Obama	Barack Obama	...
null	*null*	Giorgio Dendi	...
...	...	...	...

Fig. 1. A portion of the entity matrix

3 Kernels for Entity Classification

The strategy adopted by kernel methods [20,19] consists of splitting the learning problem in two parts. They first embed the input data in a suitable feature space, and then use a linear algorithm (e.g., the perceptron) to discover nonlinear patterns in the input space. Typically, the mapping is performed implicitly by a so-called *kernel function*. The kernel function is a similarity measure between

the input data that depends exclusively on the specific data type and domain. A typical similarity function is the inner product between feature vectors. Characterizing the similarity of the inputs plays a crucial role in determining the success or failure of the learning algorithm, and it is one of the central questions in the field of machine learning.

Formally, the kernel is a function $k : X \times X \to \mathbb{R}$ that takes as input two data objects (e.g., vectors, texts, parse trees) and outputs a real number characterizing their similarity, with the property that the function is symmetric and positive semi-definite. That is, for all x_1, $x_2 \in X$, it satisfies

$$k(x_1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle,$$

where ϕ is an explicit mapping from X to an (inner product) feature space $\mathcal{F}$.

In the remainder of this section, we define and combine different kernel functions that calculate the pairwise similarity between entities using their corresponding Wikipedia articles as source of information. They are the only domain specific elements of our classification system, while the learning algorithm is a general purpose component. Many classifiers can be used with kernels, we use k-nearest neighbor (k-nn).

3.1 Bag-of-Features Kernels

The simplest method to calculate the similarity between two entities is to compute the inner product of their vector representation in the vector space model (VSM). Formally, we define a space of dimensionality N in which each dimension is associated with one feature, and the entity e is represented in the language $l_j \in \mathcal{L}$ by a row vector

$$\phi_j(e_i) = (w(f_1, E_{i,j}), w(f_2, E_{i,j}), \ldots, w(f_N, E_{i,j})), \tag{1}$$

where the function $w(f_k, E_{i,j})$ records whether a particular feature f_k is active in the Wikipedia article $E_{i,j}$. Using this representation we define the *bag-of-features kernel* between entities as

$$K_F(e_1, e_2) = \sum_{j=1}^{|\mathcal{L}|} \langle \phi_j(e_1), \phi_j(e_2) \rangle, \tag{2}$$

Notice that this kernel computes the similarity between e_1 and e_2 as the sum of their similarities in those languages for which Wikipedia articles exist. Based on this general formulation, we define 4 basic kernel functions as follows.

Bag-of-Templates Kernel. To define the similarity between pairs of entities, we count how many occurrences of templates their corresponding Wikipedia articles in a specific language share. Templates are commonly used for boilerplate messages, standard warnings or notices, infoboxes, navigational boxes and similar purposes. In our experiments, we take into consideration solely the infoboxes (Section 4.1 describes the set of heuristics used to extract the infoboxes). The *Bag-of-templates kernel* (K_T) is defined as in Equation (2), where the function $w(f_k, E_{i,j})$ in Equation (1) is a binary function that records whether a particular infobox f_k is used in the Wikipedia article $E_{i,j}$.

Bag-of-Categories Kernel. Wikipedia categories are intended to group together articles on similar subjects and have proven useful in text classification [22], ontology learning [15], and ontology population [21]. The *bag-of-categories kernel* (K_C) is defined as in Equation (2) where the function $w(f_k, E_{i,j})$ in Equation (1) is a binary function that records whether a particular category f_k is used in the Wikipedia article $E_{i,j}$.

Bag-of-Sections Kernel. Wikipedia articles are structured in several sections that might provide relevant cues for classification. For example, biographical articles typically include sections like *Early life*, *Career*, and *Personal life*; while articles referring to cities usually include sections like *Places of interest*, *Demographic evolution*, and *Administration*. The *bag-of-sections kernel* (K_C) is defined as in Equation (2) where the function $w(f_k, E_{i,j})$ in Equation (1) is a binary function that records whether a particular section name f_k is used in the Wikipedia article $E_{i,j}$.

Bag-of-Words Kernel. The use of infoboxes, categories, and sections ensures highly accurate classification, however it produces extremely sparse feature spaces that compromises the recall. To overcome this problem, we also exploit content words of the text article as additional sources of information. The *bag-of-words kernel* (K_W) is defined as in Equation (2) where the function $w(f_k, E_{i,j})$ in Equation (1) is the standard *term frequency–inverse document frequency* (tf $\times$ idf) of the word f_k in the Wikipedia article $E_{i,j}$.

3.2 Latent Semantic Kernel

Given that the bag-of-words representation does not deal well with lexical variability, in the following we introduce the latent semantic kernels and show how to define an effective semantic VSM using (unlabeled) external knowledge. It has been shown that semantic information is fundamental for improving the accuracy and reducing the amount of training data in many natural language tasks, including fine-grained classification of named entities [4,7], question classification [12], text categorization [9], word sense disambiguation [10].

In the context of kernel methods, semantic information can be integrated considering linear transformations of the type $\tilde{\phi_j}(c_t) = \phi_j(c_t)\mathbf{S}$, where $\mathbf{S}$ is a $N \times k$ matrix [20]. The matrix $\mathbf{S}$ can be rewritten as $\mathbf{S} = \mathbf{WP}$, where $\mathbf{W}$ is a diagonal matrix determining the word weights, while $\mathbf{P}$ is the *word proximity matrix* capturing the semantic relations between words. The proximity matrix $\mathbf{P}$ can be defined by setting non-zero entries between those words whose semantic relation is inferred from an external source of domain knowledge. The *semantic kernel* takes the general form

$$\tilde{k_j}(e_1, e_2) = \phi_j(e_1)\mathbf{SS}'\phi_j(e_2)' = \tilde{\phi_j}(e_1)\tilde{\phi_j}(e_2)'. \tag{3}$$

It follows directly from the explicit construction that Equation (3) defines a valid kernel.

To define the proximity matrix for the latent semantic kernel, we look at co-occurrence information in a (large) corpus. Two words are considered semantically related if they frequently co-occur in the same texts. We use singular valued decomposition (SVD) to automatically derive the proximity matrix $\mathbf{\Pi}$ from a corpus, represented by its term-by-document matrix $\mathbf{D}$, where the $\mathbf{D}_{i,j}$ entry gives the frequency of term p_i in document d_t.[6] SVD decomposes the term-by-document matrix $\mathbf{D}$ into three matrixes $\mathbf{D} = \mathbf{U\Sigma V}'$, where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices (i.e., $\mathbf{U}'\mathbf{U} = \mathbf{I}$ and $\mathbf{V}'\mathbf{V} = \mathbf{I}$) whose columns are the eigenvectors of $\mathbf{DD}'$ and $\mathbf{D}'\mathbf{D}$ respectively, and $\mathbf{\Sigma}$ is the diagonal matrix containing the singular values of $\mathbf{D}$. Under this setting, we define the proximity matrix $\mathbf{\Pi}$ as follows:

$$\mathbf{\Pi} = \mathbf{U}_k\mathbf{\Sigma}_k,$$

where $\mathbf{U}_k$ is the matrix containing the first k columns of $\mathbf{U}$ and k is the dimensionality of the latent semantic space and can be fixed in advance. By using a small number of dimensions, we can define a very compact representation of the proximity matrix and, consequently, reduce the memory requirements while preserving most of the information.

The matrix $\mathbf{\Pi}$ is used to define a linear transformation $\pi_j : \mathbb{R}^N \to \mathbb{R}^k$, that maps the vector $\phi_j(e_t)$, represented in the standard VSM, into the vector $\tilde{\phi_j}(e_t)$ in the latent semantic space. Formally, π_j is defined as follows

$$\pi_j(\phi_j(e_t)) = \phi_j(e_t)(\mathbf{W\Pi}) = \tilde{\phi_j}(e_t), \tag{4}$$

where $\phi_j(e_t)$ is a row vector, $\mathbf{W}$ is a $N \times N$ diagonal matrix determining the word weights such that $\mathbf{W}_{i,i} = \log(\text{idf}(w_i))$, where $\text{idf}(w_i)$ is the *inverse document frequency* of w_i.

Finally, the *latent semantic kernel* is explicitly defined as follows

$$K_L(e_1, e_2) = \sum_{j=1}^{|\mathcal{L}|} \langle \pi_j(\phi_j(e_1)), \pi_j(\phi_j(e_2)) \rangle,$$

where ϕ_j is the mapping defined in Equation (1) and π_j is the linear transformation defined in Equation (4) in language $l_j \in \mathcal{L}$. Note that we have used a series of successive mappings each of which adds some further improvement to the entity representation.

3.3 Composite Kernel

Having defined all the basic kernels, representing different characteristics of entity descriptions, we finally define the composite kernel as

$$K_{\text{COMBO}}(e_1, e_2) = \sum_{n=1} \frac{K_n(e_1, e_2)}{\sqrt{K_n(e_1, e_2) K_n(e_1, e_2)}}, \tag{5}$$

[6] SVD has been first applied to perform latent semantic analysis of terms and latent semantic indexing of documents in large corpora by [3].

Table 1. Versions of DBpedia and Wikipedia used for our tests

	English	Italian	German	French	Spanish	Portuguese
Wikipedia	2012-10-01	2012-09-21	2012-10-09	2012-10-07	2012-09-27	2012-10-06
DBpedia	2012-06-04	2012-10-12	2012-06-04	2012-06-04	2012-06-04	2012-06-04

where K_n is a valid basic kernel. The individual kernels are normalized. This plays an important role in allowing us to integrate information from heterogeneous feature spaces. It follows directly from the explicit construction of the feature space and from closure properties of kernels that the composite kernel is a valid kernel.

4 Experiments

In this section, we evaluate different setups on the task of DBpedia expansion for six languages (English, Italian, German, French, Spanish, and Portuguese). The evaluation only concerns the second stage of our approach, because the first stage has precision almost 100% (see Section 4.1).

4.1 Pre-processing Wikipedia and DBpedia

Our experiments and results refer to the versions of Wikipedia and DBpedia available when this work started in mid October 2012. Table 1 lists the dumps used.

Wikipedia. We parsed the dump files to extract information about each single article and we built the matrix E using cross-language links (see Section 2). We manually check the accuracy of these links on 100 random pages: all of them were correct, so we can assume that the precision of this step is 100%. The matrix E build upon six languages (English, Italian, German, French, Spanish, and Portuguese) contains 5,626,515 entities.

We use a particular strategy for the template extraction, as we only want infoboxes for our classification. As Wikipedia does not provide a simple way to select only such type of templates, we implemented a simple rule-based hand-crafted classifier[7] to filter templates that (i) appear less than 50 times, (ii) appear mostly more than once in a page, (iii) are not written in a key/value form, and (iv) are written on multiple lines. In this way, we filter more than 90% of the templates, obtaining an average of a couple of templates for each page.

DBpedia. Starting from DBpedia dumps, we created a mapping that combines the entities in E with the ontology class(es) they belong to. Using entities instead of Wikipedia pages allows us to automatically extend and improve the DBpedia

[7] Looking at the template name for keywords such as `Infobox` is not a good strategy, as there is plenty of infobox templates that do not follow this rule.

Table 2. Total number of pages in Wikipedia, in DBpedia, and in DBpedia after using Wikipedia cross-language links. Quantities in the last row represent, for each language, the number of pages not included in DBpedia in any language considered

	Matrix E	EN	IT	DE	FR	ES	PT
Wikipedia	5,626,515	3,932,148	924,544	1,325,792	1,251,585	953,848	740,585
DBpedia	-	1,716,555	607,842	205,903	15,463	15,987	226,497
DBpedia CL	2,193,742	1,902,585	652,395	482,747	518,874	419,168	430,603
Not classified	3,432,773	2,029,563	272,149	843,045	732,711	534,680	309,982

coverage. For instance, *Michael Jackson* is classified as a `Person` in the Italian and German DBpedia, an `Artist` in the English DBpedia and a `MusicalArtist` in the Spanish DBpedia. The most specific class is the last one, so the *entity* Michael Jackson becomes `MusicalArtist` in every language. The final mapping contains 2,193,742 entities: comparing this figure with the size of the matrix E, this means that there are around 3,432,773 entities in Wikipedia that are not classified in DBpedia. In our experiments we always refer to this set for the classification part that makes use of kernel methods. Data concerning the enriched DBpedia is shown in Table 2.

4.2 Benchmark

Experiments are carried out on a benchmark extracted from the entity matrix introduced in Section 2. Specifically, the data set contains 400 randomly extracted entities not already present in DBpedia in any language. The data set is split in development set (100 entities) and test set (300 entities). All entities have been annotated with the most specific available class in the version 3.8 of the DBpedia ontology by one of the authors of this paper. 50 more entities have been annotated by three different annotators, resulting in an inter-agreement of 78% (Fleiss' kappa measure, see [5]). An additional `Unknown` class has been introduced to annotate those entities that cannot be assigned to any class in the ontology. When an entity is assigned to a class, it is also implicitly assigned to all its super-classes. For instance, classifying *Michael Jackson* as a `MusicalArtist` we implicitly classify him as `Artist`, `Person` and `Agent`.

The evaluation is performed as proposed by [13] for a similar hierarchical categorization task. In the example above, classifying *Michael Jackson* as an `Athlete`, we obtain a false positive for this wrong classified class, two false negatives for missing classes `MusicalArtist` and `Artist`, and two true positives for `Person` and `Agent`.

4.3 Latent Semantic Models

For each language, we derive the proximity matrix Π (Section 3) from the 200,000 most visited Wikipedia articles. After removing terms that occur less

than 5 times, the resulting dictionaries contain about 300,000 terms. We use the SVDLIBC package[8] to compute the SVD, truncated to 100 dimensions.

4.4 Learning Algorithm

We use a k-nn classifier[9] to classify novel entities into the DBpedia ontology. The optimization of the parameter k is performed on the development set, and $k = 10$ results as the best choice, because it maximizes the F_1 value. Entities are classified by a majority vote of their neighbors. To change the tradeoff between precision and recall, we set the minimum number of votes z ($1 \leqslant z \leqslant k$) a class needs to obtain to be assigned. The algorithm has maximum precision with $z = k$, maximum recall with $z = 1$, and maximum F_1 with $z = 8$.

To train the classifier, we randomly select 100,000 entities from the matrix E included in DBpedia. Each entity is then labelled according to the corresponding DBpedia class.

4.5 Classification Schemas

We compare three alternative training and classification schemas.

Bottom-up. A single training and classification step is performed. k-nn is trained using entities annotated with the most specific classes in DBpedia. In classification, an entity is annotated with the finer-grained class c if c receives $v_c \geqslant z$ votes; `Unknown` otherwise.[10] Note that the algorithm also considers the super-classes of c: if a fine-grained class cannot be assigned with a given confidence level z, it could return a more generic one s ($s \subseteq c$) such that $v_s \geqslant z$. For instance, if $z = 10$ and the 10 votes are divided 5 to `Astronaut` and 5 to `Scientist`, our system answers `Unknown` because none of the classes obtains 10 votes. However, ascending the ontology, we find that the class `Person` receives 10 votes, as both `Astronaut` and `Scientist` belong to it. The system then classifies it as `Person`, instead of `Unknown`. In case this process does not find any class at any level with a sufficient number of votes, the `Unknown` answer is given.

Top-down. Multiple training and classification steps are performed. k-nn is trained using entities annotated with the most generic classes in DBpedia (ontology top-level). In classification, an entity is annotated with a generic class c if it receives $v_c \geqslant z$ votes; `Unknown` otherwise. The procedure is recursively repeated on all subtrees of the ontology using the previous classification to limit the number of classes to consider.

[8] `http://tedlab.mit.edu/~dr/svdlibc/`

[9] During the first experiments, we used our algorithms with two test classes: `Person` and `Place`. In this phase, Support Vector Machine (SVM) produced very good results. When we applied our approach to the entire DBpedia ontology (359 classes), SVM performance dramatically dropped.

[10] Assigning the class `Unknown` is equivalent to abstention.

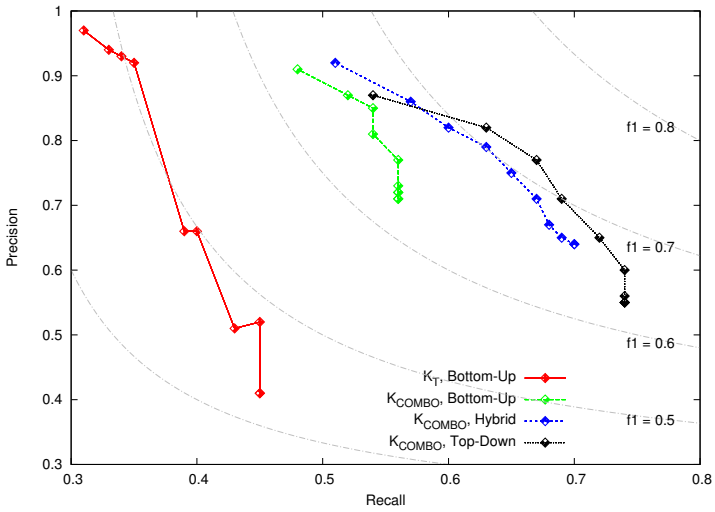


Fig. 2. Precision/recall curve of the system

Table 3. Results of the most frequent class baseline (MF), the basic kernels (see Section 3.1) and the composite kernel K_{COMBO}, using $z = 10$

	MF	K_T	K_C	K_S	K_W	K_L	K_{COMBO}
Precision	0.35	0.97	0.90	0.94	0.81	0.84	0.91
Recall	0.38	0.31	0.40	0.16	0.22	0.41	0.48
F_1	0.31	0.47	0.55	0.27	0.34	0.55	0.63

Hybrid. This variant consists in first training a k-nn as defined in the Bottom-up schema. Then, a set of specialized k-nns are trained for the most populated classes, such as, `Person`, `Organisation`, `Place`, `Work`, etc. In classification, let P be one of these classes, the Bottom-up schema is applied first. Then, if an entity is annotated with the class c such that $c \in P$, then a specialized k-nn is applied.

4.6 Results

First, we investigate the contribution of the kernel combination (Section 3) and then the one of the different training and classification schemas (Section 4.5).

Table 3 reports the results of the most frequent class baseline, the basic kernels (K_T, K_C, K_S, K_W, and K_L), and the composite kernel K_{COMBO}. The experimental results show that the composite kernel K_{COMBO} significantly outperforms the basic kernels. We use approximate randomization [16] to assess the statistical significance between the obtained results (p-value = 0.05).

Figure 2 shows the precision/recall curves obtained by varying the parameter z. We also draw, in grey in the background, the contour lines joining points with the same F_1, so that one can quickly visualize this value. Four different setups are

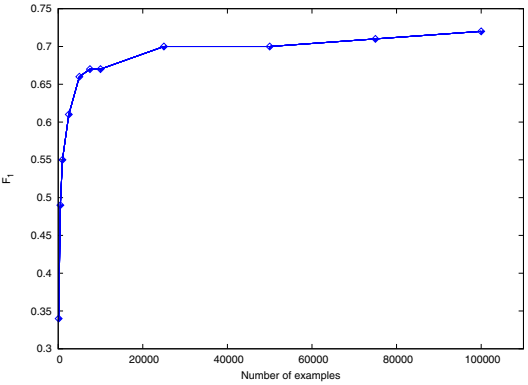


Fig. 3. Learning curve of the system

compared in order to determine the one that produces the best tradeoff between precision and recall.

K_T, Bottom-up uses only the template information (as in the DBpedia framework) and the Bottom-up schema, obtaining the maximum precision of 97% at the expense of low recall of 31% ($z = 10$).

K_{COMBO}, Bottom-up uses all the sources of information and the Bottom-up schema, obtaining a significant improvement in recall (48%) preserving a high precision of 91% ($z = 10$).

K_{COMBO}, Hybrid uses all the sources of information and the Hybrid schema, obtaining a further improvement of precision (92%) and recall (51%).

K_{COMBO}, Top-down uses all the sources of information and the Top-down schema, obtaining the maximum recall (54%), however the precision (87%) is significantly lower than the one obtained in the other experiments.

Figure 3 shows the learning curve of the system in term of F_1 in the configuration that maximizes the F_1 score (in our experiments, this happens in all configurations, when $z = 8$).

Finally, we perform some preliminary error analysis. Errors mostly depend on the following factors: (i) the Wikipedia article is too short; (ii) an appropriate class for the entity does not exist (this often happens with common nouns); (iii) some Wikipedia pages represent lists (for example, `Liste_des_conseillers...`) and our system often classifies them as the objects listed (in the example, `Person`); (iv) nesting of DBpedia classes is not optimal (for example, `Astronaut` and `Scientist` are disjoint classes). The most common factor is (iii), as it is the cause of more than half of the errors in the experiments on the test set.

5 Related Work

The DBpedia project [1], started in 2007, manually creates an ontology starting from Wikipedia infobox templates. Nowadays, the English version covers

around 1.7M Wikipedia pages, although the English Wikipedia contains almost 4M pages. Other languages suffer from an even lower coverage (see Table 2).

Differently, Yago [21], another similar project also started in 2007, aims to extract and map entities from Wikipedia using categories (for fine-grained classes) and WordNet (for upper-level classes). Its coverage is higher, but it is monolingual and its ontology contains thousands of hundreds of classes: it may be difficult to use it in practical applications.

There are also other projects aiming to extract Wikipedia entity types boostrapping information contained in the categories. For example, [17] uses extracted datatypes to train a name entity recogniser, while [15] investigates Wikipedia categories and automatically cleans them.

The tool presented in [6], *Tipalo*, identifies the most appropriate class of a Wikipedia article by interpreting its page abstract using natural language processing tools and resources. In this context, only English Wikipedia is considered, as this classifier cannot be easily adapted to other languages.

Similarly, [18] only considers the English DBpedia and therefore does not take advantages from inter-language links. In addition, there is some manual effort to classify biographies (using tokens from categories), that leads to very good results, but is not automatically portable to other languages; again linguistic tools are used to extract the definition from the first sentence.

The approach presented in [7] classifies people on an excerpt of the WordNet ontology, using kernel functions that implicitly map entities, represented by aggregating all contexts in which they occur, into a latent semantic space derived from Wikipedia. This approach queries online the name of the entity to collect contextual information. We specialize this approach to Wikipedia, that is easily to download and store locally.

[8] proposes an unsupervised approach based on lexical entailment, consisting in assigning an entity to the category whose lexicalization can be replaced with its occurrences in a corpus preserving the meaning.

6 Conclusions and Future Work

We have proposed a two-stage approach that automatically extends the coverage of DBpedia with respect to Wikipedia articles. We have first extended the population of DBpedia using cross-language links, and then used this extended population as training data to classify the remaining pages using a kernel-based supervised method. The experiments have been evaluated on a manually annotated test set containing 400 Wikipedia pages, resulting in high precision and recall, with different tradeoffs of these values depending on the configuration of the algorithm. The resulting resource is available both as a download package and a SPARQL endpoint at `http://www.airpedia.org/`.

DBpedia also maps entity properties, such as `BirthDate` and `birthPlace` for `Person`, `director` for `Film`, and so on. We are currently working to deal with this problem, using natural language processing tools to find the correct relation in the article text. This can be seen as a relation extraction task, and one of the

most reliable approaches to tackle this problem (starting from a large available knowledge base) is *distant supervision* [14]. This paradigm has been successfully used for pattern extraction [23] and question answering [2]. Moreover, we want to deal with entities belonging to more than one class. Some entities in DBpedia are correctly classified in multiple classes. For example, Madonna is a singer (`MusicalArtist`) and an actress (`Actor`).

Finally, we will investigate how to build a new ontology based on Wikipedia categories together with templates, using the results produced by our system.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.G.: DBpedia: A nucleus for a web of open data. In: Aberer, K., et al. (eds.) ISWC/ASWC 2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007)
2. Cabrio, E., Cojan, J., Palmero Aprosio, A., Magnini, B., Lavelli, A., Gandon, F.: QAKiS: An open domain QA system based on relational patterns. In: Glimm, B., Huynh, D. (eds.) International Semantic Web Conference (Posters & Demos). CEUR Workshop Proceedings, vol. 914, CEUR-WS.org (2012)
3. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. Journal of the American Society of Information Science 41(6), 391–407 (1990)
4. Fleischman, M., Hovy, E.: Fine grained classification of named entities. In: Proceedings of the 19th International Conference on Computational Linguistics, Taipei, Taiwan (2002)
5. Fleiss, J.L.: Measuring Nominal Scale Agreement Among Many Raters. Psychological Bulletin 76(5), 378–382 (1971)
6. Gangemi, A., Nuzzolese, A.G., Presutti, V., Draicchio, F., Musetti, A., Ciancarini, P.: Automatic typing of DBpedia entities. In: Cudré-Mauroux, P., Heflin, J., Sirin, E., Tudorache, T., Euzenat, J., Hauswirth, M., Parreira, J.X., Hendler, J., Schreiber, G., Bernstein, A., Blomqvist, E. (eds.) ISWC 2012, Part I. LNCS, vol. 7649, pp. 65–81. Springer, Heidelberg (2012)
7. Giuliano, C.: Fine-grained classification of named entities exploiting latent semantic kernels. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning, CoNLL 2009, pp. 201–209. Association for Computational Linguistics, Stroudsburg (2009)
8. Giuliano, C., Gliozzo, A.: Instance based lexical entailment for ontology population. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 248–256. Association for Computational Linguistics, Prague (2007)
9. Gliozzo, A., Strapparava, C.: Domain kernels for text categorization. In: Ninth Conference on Computational Natural Language Learning (CoNLL 2005), Ann Arbor, Michigan, pp. 56–63 (June 2005)
10. Gliozzo, A.M., Giuliano, C., Strapparava, C.: Domain kernels for word sense disambiguation. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005), Ann Arbor, Michigan, pp. 403–410 (June 2005)
11. Kontokostas, D., Bratsas, C., Auer, S., Hellmann, S., Antoniou, I., Metakides, G.: Internationalization of Linked Data: The case of the Greek DBpedia edition. Web Semantics: Science, Services and Agents on the World Wide Web 15, 51–61 (2012)

12. Li, X., Roth, D.: Learning question classifiers: The role of semantic information. Natural Language Engineering 12(3), 229–249 (2005)
13. Dan Melamed, I., Resnik, P.: Tagger evaluation given hierarchical tag sets. Computers and the Humanities, 79–84 (2000)
14. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, ACL 2009, vol. 2, pp. 1003–1011. Association for Computational Linguistics, Stroudsburg (2009)
15. Nastase, V., Strube, M.: Decoding Wikipedia categories for knowledge acquisition. In: Proceedings of the 23rd National Conference on Artificial Intelligence, AAAI 2008, vol. 2, pp. 1219–1224. AAAI Press (2008)
16. Noreen, E.W.: Computer-Intensive Methods for Testing Hypotheses: An Introduction. Wiley-Interscience (1989)
17. Nothman, J., Curran, J.R., Murphy, T.: Transforming Wikipedia into named entity training data. In: Proceedings of the Australasian Language Technology Workshop, Hobart, Australia (2008)
18. Pohl, A.: Classifying the Wikipedia Articles into the OpenCyc Taxonomy. In: Proceedings of the Web of Linked Entities Workshop in Conjuction with the 11th International Semantic Web Conference (2012)
19. Schölkopf, B., Smola, A.: Learning with Kernels. MIT Press, Cambridge (2002)
20. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press (2004)
21. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: A core of semantic knowledge. In: Proceedings of the 16th International Conference on World Wide Web, WWW 2007, pp. 697–706. ACM, New York (2007)
22. Wang, P., Hu, J., Zeng, H.-J., Chen, Z.: Using wikipedia knowledge to improve text classification. Knowledge and Information Systems 19, 265–281 (2009), doi:10.1007/s10115-008-0152-4
23. Wu, F., Weld, D.S.: Open information extraction using wikipedia. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL 2010, pp. 118–127. Association for Computational Linguistics, Stroudsburg (2010)

A Support Framework for Argumentative Discussions Management in the Web

Elena Cabrio, Serena Villata, and Fabien Gandon

INRIA Sophia Antipolis, France
firstname.lastname@inria.fr

Abstract. On the Web, wiki-like platforms allow users to provide arguments in favor or against issues proposed by other users. The increasing content of these platforms as well as the high number of revisions of the content through pros and cons arguments make it difficult for community managers to understand and manage these discussions. In this paper, we propose an automatic framework to support the management of argumentative discussions in wiki-like platforms. Our framework is composed by (i) a natural language module, which automatically detects the arguments in natural language returning the relations among them, and (ii) an argumentation module, which provides the overall view of the argumentative discussion under the form of a directed graph highlighting the accepted arguments. Experiments on the history of Wikipedia show the feasibility of our approach.

1 Introduction

On the Social Web, wiki-like platforms allow users to publicly publish their own arguments and opinions. Such arguments are not always accepted by other users on the Web, leading to the publication of additional arguments attacking or supporting the previously proposed ones. The most well known example of such kind of platform is Wikipedia[1] where users may change pieces of text written by other users to support, i.e., further specify them, or attack them, i.e., correcting factual errors or highlighting opposite points of view. Managing such kind of "discussions" using the revision history is a tricky task, and it may be affected by a number of drawbacks. First, the dimension of these discussions makes it difficult for both users and community managers to navigate, and more importantly, understand the meaning of the ongoing discussion. Second, the discussions risk to re-start when newcomers propose arguments which have already been proposed and addressed in the same context. Third, these discussions are not provided in a machine-readable format to be queried by community managers to discover insightful meta-information on the discussions themselves, e.g., discover the number of attacks against arguments about a particular politician concerning the economic growth during his government.

In this paper, we answer the following research question: *how to support community managers in managing the discussions on the wiki pages?* This question

[1] http://en.wikipedia.org/wiki/Main_Page

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 412–426, 2013.

breaks down into the following subquestions: (i) how to automatically discover the arguments and the relations among them?, and (ii) how to have the overall view of the ongoing discussion to detect the *winning* arguments? The answer to these sub-questions allows us to answer to further questions: how to detect repeated arguments and avoid loops of changes?, and how to discover further information on the discussion history? Approaches such as the lightweight vocabulary SIOC Argumentation [13] provide means to model argumentative discussions of social media sites, but they are not able to automatically acquire information about the argumentative structures. As underlined by Lange et al. [13], such a kind of automatic annotation needs the introduction of Natural Language Processing (NLP) techniques to automatically detect the arguments in the texts.

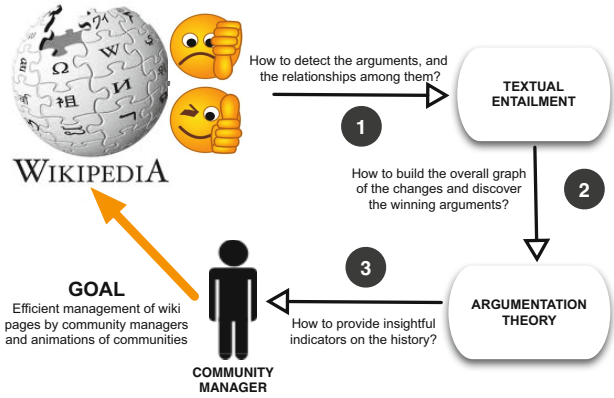


Fig. 1. An overview of the proposed approach to support community managers.

In this work, we propose a combined framework where a natural language module that automatically detects the arguments and their relations (i.e. *support* or *challenge*), is coupled with an argumentation module to have the overall view of the discussion and detect the winning arguments, as visualized in Figure 1.

First, to automatically detect natural language arguments and their relations, we rely on the Textual Entailment (TE) framework, proposed as an applied model to capture major semantic inference needs across applications in the NLP field [8]. Differently from formal approaches to semantic inference, in TE linguistic objects are mapped by means of semantic inferences at a textual level.

Second, we adopt abstract argumentation theory [9] to unify the results of the TE module into a unique argumentation framework able not only to provide the overall view of the discussion, but also to detect the set of *accepted* arguments relying on argumentation semantics. Argumentation theory aims at representing the different opinions of the users in a structured way to support decision making.

Finally, the generated argumentative discussions are described using an extension of the SIOC Argumentation vocabulary[2] thus providing a machine readable version. Such discussions expressed using RDF allow the extraction of a kind of "meta-information" by means of queries, e.g., in SPARQL. These meta-information cannot be easily detected by human users without the support of our automatic framework.

The aim of the proposed framework is twofold: on one side, we want to provide a support to community managers for notification and reporting, e.g., notify the users when their own arguments are attacked, and on the other hand, we support community managers to extract further insightful information from the argumentative discussions. As a case study, we apply and experiment our framework on Wikipedia revision history over a four-year period, focusing in particular on the top five most revised articles.

The paper is organized as follows. Section 2 provides some basic insights on abstract argumentation theory and textual entailment. Section 3 presents our combined framework to support the management of argumentative discussions in wiki-like platforms, and in Section 4 we report on the experimental setting and results. Section 5 presents and compares the related work.

2 Background: Argumentation and NLP

In this section, we provide notions of abstract argumentation theory and of textual entailment, essential to our work.

2.1 Abstract Argumentation Theory

A Dung-style argumentation framework [9] aims at representing conflicts among elements called *arguments* through a binary *attack* relation. It allows to reason about these conflicts in order to detect, starting by a set of arguments and the conflicts among them, which are the so called *accepted arguments*. The accepted arguments are those arguments which are considered as believable by an external evaluator, who has a full knowledge of the argumentation framework.

Definition 1 (Abstract argumentation framework *AF* [9]). *An abstract argumentation framework is a tuple* $\langle A, \rightarrow \rangle$ *where* A *is a finite set of elements called arguments and* $\rightarrow$ *is a binary relation called attack defined on* $A \times A$.

Dung [9] presents several acceptability semantics that produce zero, one, or several sets of accepted arguments. The set of accepted arguments of an argumentation framework consists of a set of arguments that does not contain an argument attacking another argument in the set. Roughly, an argument is *accepted* if all the arguments attacking it are rejected, and it is *rejected* if it has at least an argument attacking it which is accepted. In Figure 2.a, an example of abstract argumentation framework is shown. The arguments are visualized as circles, and

[2] `http://rdfs.org/sioc/argument`

the attack relation is visualized as edges in the graph. Gray arguments are the accepted ones. We have that argument a attacks argument b, and argument b attacks argument c. Using Dung's acceptability semantics [9], the set of accepted arguments of this argumentation framework is $\{a, c\}$.

The need of introducing also a positive relation among the arguments, i.e., a *support* relation, leads to the emergence of the so called *bipolar* argumentation frameworks [6].

Definition 2 (Bipolar argumentation framework BAF **[6]).** *A bipolar argumentation framework is a tuple* $\langle A, \rightarrow, \dashrightarrow \rangle$ *where* A *is a finite set of arguments,* $\rightarrow \subseteq A \times A$*, and* $\dashrightarrow$ *is a binary relation called support defined on* $A \times A$*.*

An example of bipolar argumentation framework is visualized in Figure 2.b where the dashed edge represents the support relation. For more details about acceptability semantics in BAFs, see [6].

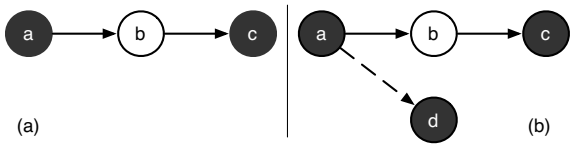


Fig. 2. Example of (a) an abstract argumentation framework, and (b) a BAF

2.2 Textual Entailment

In the NLP field, the notion of Textual entailment refers to a directional relation between two textual fragments, termed *Text (T)* and *Hypothesis (H)*, respectively. The relation holds (i.e. $T \Rightarrow H$) whenever the truth of one text fragment follows from another text, as interpreted by a typical language user. The TE relation is directional, since the meaning of one expression may usually entail the other, while entailment in the other direction is much less certain. Consider the pairs in Example 1 and 2:

Example 1
T: Jackson had three sisters: Rebbie, La Toya, and Janet, and six brothers: Jackie, Tito, Jermaine, Marlon, Brandon (Marlon's twin brother, who died shortly after birth) and Randy.
H: Jackson's siblings are Rebbie, Jackie, Tito, Jermaine, La Toya, Marlon, Randy and Janet.

Example 2 (Continued)
T: It was reported that Jackson had offered to buy the bones of Joseph Merrick (the elephant man) and although untrue, Jackson did not deny the story.
H: Later it was reported that Jackson bought the bones of The Elephant Man.

In Example 1, we can identify an inference relation between T and H (i.e. the meaning of H can be derived from the meaning of T), while in Example 2, T contradicts H. The notion of TE has been proposed [8] as an applied framework to capture major semantic inference needs across applications in NLP (e.g. information extraction, text summarization, and reading comprehension systems). The task of recognizing TE is therefore carried out by automatic systems, mainly implemented using Machine Learning techniques (typically SVM), logical inference, cross-pair similarity measures between T and H, and word alignment.[3] While entailment in its logical definition pertains to the meaning of language expressions, the TE model does not represent meanings explicitly, avoiding any semantic interpretation into a meaning representation level. Instead, in this applied model inferences are performed directly over lexical-syntactic representations of the texts. TE allows to overcome the main limitations showed by formal approaches (where the inference task is carried out by logical theorem provers), i.e. *(i)* the computational costs of dealing with huge amounts of available but noisy data present in the Web; *(ii)* the fact that formal approaches address forms of deductive reasoning, exhibiting a too high level of precision and strictness as compared to human judgments, that allow for uncertainties typical of inductive reasoning. But while methods for automated deduction assume that the arguments in input are already expressed in some formal meaning representation (e.g. first order logic), addressing the inference task at a textual level opens different and new challenges from those encountered in formal deduction. Indeed, more emphasis is put on informal reasoning, lexical semantic knowledge, and variability of linguistic expressions.

3 The Combined Framework

In a recent work, Cabrio and Villata [2] propose to combine natural language techniques and Dung-like abstract argumentation to generate the arguments from natural language text and to evaluate this set of arguments to know which are the accepted ones, with the goal of supporting the participants in natural language debates (i.e. Debatepedia[4]). In particular, they adopt the TE approach, and in their experiments, they represent the TE relation extracted from natural language texts as a *support* relation in bipolar argumentation. In this paper, we start from their observations, and we apply the combined framework proposed in [2] to this new scenario.

Let us consider the argument in Example 3 from the Wikipedia article "United States", and its revised versions in the last four years[5]:

[3] *Dagan et al. (2009)* [8] provides an overview of the recent advances in TE.

[4] `http://bit.ly/Dabatepedia`

[5] Since we are aware that Wikipedia versions are revised daily, we have picked our example from a random dump per year. In Section 4.1, we provide more details about the Wikipedia sample we consider in our experiments.

Example 3
In 2012: The land area of the contiguous United States is 2,959,064 square miles (7,663,941 km2).
In 2011: The land area of the contiguous United States is approximately 1,800 million acres (7,300,000 km2).
In 2010: The land area of the contiguous United States is approximately 1.9 billion acres (770 million hectares).
In 2009: The total land area of the contiguous United States is approximately 1.9 billion acres.

Several revisions have been carried out by different users during this four-year period, both to correct factual data concerning the U.S. surface, or to better specify them (e.g. providing the same value using alternative metric units). Following [2], we propose to take advantage of NLP techniques to automatically detect the relations among the revised versions of the same argument, to verify if the revisions done on the argument by a certain user at a certain point in time support the original argument (i.e. the user has rephrased the sentence to allow an easier comprehension of it, or has added more details), or attack it (i.e. the user has corrected some data, has deleted some details present in the previous version or has changed the semantics of the sentence providing a different viewpoint on the same content). Given the high similarities among the entailment and contradiction notions in TE and the support and attack relation in argumentation theory, we cast the described problem as a TE problem, where the T-H pair is a pair of revised arguments in two successive Wikipedia versions. We consider paraphrases as bidirectional entailment, and therefore to be annotated as a positive TE pair (i.e. support). Moreover, since the label *no entailment* includes both contradictions and pairs containing incomplete informational overlap (i.e. H is more informative than T), we consider both cases as *attacks*, since we want community managers to check the reliability of the corrected or deleted information. To build the T-H pairs required by the TE framework, for each argument we set the revised sentence as T and the original sentence as H, following the chronological sequence, since we want to verify if the more recent version entails or not the previous one, as shown in Example 4.

Example 4 (Continued)
pair id=70.1 entailment=NO
T (Wiki12): The land area of the contiguous United States is 2,959,064 square miles (7,663,941 km2).
H (Wiki11): The land area of the contiguous United States is approximately 1,800 million acres (7,300,000 km2).

pair id=70.2 entailment=NO
T (Wiki11): The land area of the contiguous United States is approximately 1,800 million acres (7,300,000 km2).
H (Wiki10): The land area of the contiguous United States is approximately 1.9 billion acres (770 million hectares).

pair id=70.3 entailment=YES
T (Wiki10): The land area of the contiguous United States is approximately 1.9 billion acres (770 million hectares).
H (Wiki09): The total land area of the contiguous United States is approximately 1.9 billion acres.

On such pairs we apply a TE system, that automatically returns the set of arguments and the relations among them. The argumentation module starts from the couples of arguments provided by the TE module, and builds the complete argumentation framework involving such arguments. It is important to underline a main difference with respect to the approach of Cabrio and Villata [2]: here the argumentation frameworks resulting from the TE module represent a kind of *evolution* of the *same* argument during time in a specific Wikipedia article. From the argumentation point of view, we treat these arguments as separate instances of the same natural language argument giving them different names. Figure 3.a visualizes the argumentation framework of Example 4. This kind of representation of the natural language arguments and their evolution allows community managers to detect whether some arguments have been repeated in such a way that loops in the discussions can be avoided. The argumentation module, thus, is used here with a different aim from the previous approach [2]: it shows the *kind* of changes, i.e., positive and negative, that have been addressed on a particular argument, representing them using a graph-based structure.

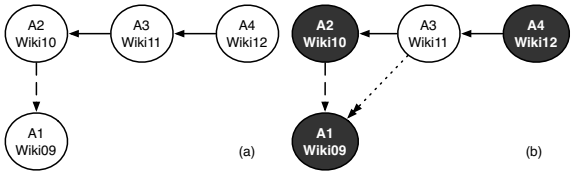


Fig. 3. The bipolar argumentation framework resulting from Example 4

The use of argumentation theory to discover the set of winning, i.e., acceptable, arguments in the framework could seem pointless, since we could assume that winning arguments are only those arguments appearing in the most recent version of the wiki page. However, this is not always the case. The introduction of the support relation in abstract argumentation theory [6] leads to the introduction of a number of *additional attacks* which are due to the presence of an attack and a support involving the same arguments. The additional attacks introduced in the literature are visualized in Figure 4, where dotted double arrows represent the additional attacks. For the formal properties of these attacks and a comparison among them, see Cayrol and Lagasquie-Schiex [6].

The introduction of additional attacks is a key feature of our argumentation module. It allows us to support community managers in detecting further possible attacks or supports among the arguments. In particular, given the arguments and their relations, the argumentation module builds the complete framework

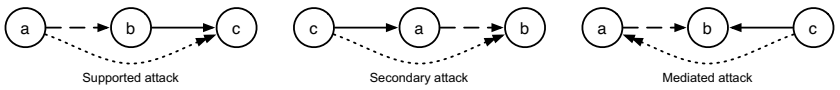


Fig. 4. The additional attacks arising due to the presence of a support relation

adding the additional attacks, and computes the extensions of the bipolar framework. An example of such kind of computation is shown in Figure 3.b where an additional attack is introduced. In this example, the set of accepted arguments would have been the same with or without the additional attack, but there are situations in which additional attacks make a difference. This means that the explicit attacks put forward by the users on a particular argument can then result in *implicit* additional attacks or supports to other arguments in the framework. Consider the arguments of Example 5. The resulting argumentation framework (see Figure 5) shows that argument $A1$ (*Wiki09*) is implicitly supported by argument $A4$ (*Wiki12*) since the attack of $A4$ (*Wiki12*) against $A3$ (*Wiki11*) leads to the introduction of an additional attack against $A2$ (*Wiki10*). The presence of this additional attack reinstates argument $A1$ (*Wiki09*) previously attacked by $A2$ (*Wiki10*). The two accepted arguments at the end are $\{A1, A4\}$.

Example 5
pair id=7.1 entailment=NO
T (Wiki12): In December 2007, the United States entered its longest post-World War II recession, prompting the Bush Administration to enact multiple economic programs intended to preserve the country's financial system.
H (Wiki11): In December 2007, the United States entered the longest post-World War II recession, which included a housing market correction, a subprime mortgage crisis, soaring oil prices, and a declining dollar value.

pair id=7.2 entailment=YES
T (Wiki11): In December 2007, the United States entered the longest post-World War II recession, which included a housing market correction, a subprime mortgage crisis, soaring oil prices, and a declining dollar value.
H (Wiki10): In December 2007, the United States entered its longest post-World War II recession.

pair id=7.3 entailment=NO
T (Wiki10): In December 2007, the United States entered its longest post-World War II recession.
H (Wiki09): In December 2007, the United States entered the second-longest post-World War II recession, and his administration took more direct control of the economy, enacting multiple economic stimulus packages.

Finally, in this paper we further enhance the framework proposed in [2] with a semantic machine readable representation of the argumentative discussions. We do not introduce yet another argumentation vocabulary, but we reuse the SIOC Argumentation module [13], focused on the fine-grained representation of

discussions and argumentations in online communities.[6] The SIOC Argumentation model is grounded on DILIGENT [5] and IBIS[7] models.

We extend the SIOC Argumentation vocabulary with two new properties `sioc_arg:challengesArg` and `sioc_arg:supportsArg` whose range and domain are `sioc_arg:Argument`. These properties represent challenges and supports from arguments to arguments, as required in abstract argumentation theory.[8] This needs to be done since in SIOC Argumentation challenges and supports are addressed from arguments towards `sioc_arg:Statement` only. Figure 6.a shows a sample of the semantic representation of Example 1 and 2 where *contradiction* is represented through `sioc_arg:challengesArg`, and *entailment* is represented through `sioc_arg:supportsArg`.

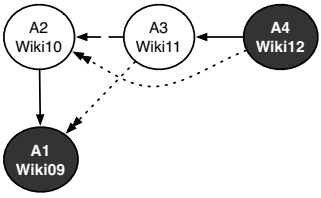


Fig. 5. The bipolar argumentation framework resulting from Example 5

```
EXAMPLE OF CONTRADICTION
<http://example.org/jako/pair1t> rdf:type sioc_arg:Argument ;

     sioc:content "It was reported that Jackson had
           offered to buy the bones of Joseph Merrick
           (the elephant man) and although untrue,
           Jackson did not deny the story." ;

     sioc_arg:challengesArg <http://example.org/jako/pair1h> .

<http://example.org/jako/pair1h> rdf:type sioc_arg:Argument ;

        sioc:content "Later it was reported that Jackson
           bought the bones of The Elephant Man." .

EXAMPLE OF ENTAILMENT
<http://example.org/jako/pair2t> rdf:type sioc_arg:Argument ;

        sioc:content "Jackson had three sisters: Rebbie,
           La Toya, and Janet, and six brothers: Jackie,
           Tito, Jermaine, Marlon, Brandon (Marlon's twin
           brother, who died shortly after birth) and
           Randy." ;

       sioc_arg:supportsArg <http://example.org/jako/pair2h> .

<http://example.org/jako/pair2h> rdf:type sioc_arg:Argument ;

        sioc:content "Jackson's siblings are Rebbie, Jackie,
           Tito, Jermaine, La Toya, Marlon, Randy and
           Janet." .
```

```
PREFIX sioc_arg:<http://rdfs.org/sioc/argument#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl:<http://www.w3.org/2002/07/owl#>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dc:<http://purl.org/dc/elements/1.1/>
PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>
PREFIX sioc:<http://rdfs.org/sioc/ns#>

SELECT ?a1 ?c1 WHERE {
   ?a1 a sioc_arg:Argument .
   ?a2 a sioc_arg:Argument .
   ?a1 sioc_arg:challengesArg ?a2 .
   ?a1 sioc:content ?c1 .
   ?a2 sioc:content ?c2
   FILTER regex(str(?c2),"crisis")
}
```

QUERY RESULT

```
T: "In December 2007, the United States entered its longest post–World
War II recession, prompting the Bush Administration to enact multiple
economic programs intended to preserve the country's financial system."
ATTACKS
H: "In December 2007, the United States entered the longest post–World
War II recession, which included a housing market correction, a subprime
mortgage crisis, soaring oil prices, and a declining dollar value."

T: "Bush entered office with the Dow Jones Industrial Average at 10,587,
and the average peaked in October 2007 at over 14,000."
ATTACKS
H: "The Dow Jones Industrial Average peaked in October 2007 at about
14,000, 30 percent above its level in January 2001, before the subsequent
economic crisis wiped out all the gains and more."
```

Fig. 6. (a) Sample of the discussions in RDF, (b) Example of SPARQL query

The semantic version of the argumentative discussions can further be used by community managers to detect insightful meta-information about the discussions themselves. For instance, given the RDF data set being stored in a datastore with SPARQL endpoint, the community manager can raise a query

[6] For an overview of the argumentation models in the Social Semantic Web, see [15].
[7] `http://purl.org/ibis`
[8] The extended vocabulary can be downloaded at `http://bit.ly/SIOC_Argumentation`

like the one in Figure 6.b. This query retrieves all those arguments which attack another argument having in the content the word "crisis". This simple example shows how the semantic annotation of argumentative discussions may be useful to discover in an automatic way those information which are difficult to be highlighted by a human user.

4 Experimental Setting

As a case study to experiment our framework we select the Wikipedia revision history. Section 4.1 describes the creation of the data set, Section 4.2 the TE system we used, while in Section 4.3 we report on obtained results.

4.1 Data Set

We create a data set to evaluate the use of TE to generate the arguments following the methodology detailed in [1]. We start from two dumps of the English Wikipedia (*Wiki 09* dated 6.03.2009, and *Wiki 10* dated 12.03.2010), and we focus on the five most revised pages[9] at that time (i.e. George W. Bush, United States, Michael Jackson, Britney Spears, and World War II). We then follow their yearly evolution up to now, considering how they have been revised in the next Wikipedia versions (*Wiki 11* dated 9.07.2011, and *Wiki 12* dated 6.12.2012).

After extracting plain text from the above mentioned pages, for both *Wiki 09* and *Wiki 10* each document has been sentence-splitted, and the sentences of the two versions have been automatically aligned to create pairs. Then, to measure the similarity between the sentences in each pair, following [1] we adopted the *Position Independent Word Error Rate (PER)*, i.e. a metric based on the calculation of the number of words which differ between a pair of sentences. For our task we extracted only pairs composed by sentences where major editing was carried out ($0.2 < PER < 0.6$), but still describe the same event.[10] For each pair of extracted sentences, we create the TE pairs setting the revised sentence (from *Wiki 10*) as T and the original sentence (from *Wiki 09*) as H. Starting from such pairs composed by the same revised argument, we checked in the more recent Wikipedia versions (i.e. *Wiki 11* and *Wiki 12*) if such arguments have been further modified. If that was the case, we created another T-H pair based on the same assumptions as before, i.e. setting the revised sentence as the T and the older sentence as the H (see Example 4). Such pairs have then been annotated with respect to the TE relation (i.e. *YES/NO entailment*), following the criteria defined and applied by the organizers of the Recognizing Textual Entailment Challenges (RTE)[11] for the two-way judgment task.

As a result of the first step (i.e. extraction of the revised arguments in *Wiki 09* and *Wiki 10*) we collected 280 T-H pairs, while after applying the procedure on the same arguments in *Wiki 11* and *Wiki 12* the total number of collected pairs is

[9] `http://bit.ly/WikipediaMostRevisedPages`

[10] A different extraction methodology has been proposed in [19].

[11] `http://www.nist.gov/tac/2010/RTE/`

452. To carry out our experiments, we randomly divided such pairs into training set (114 entailment, 114 no entailment pairs), and test set (101 entailment, 123 no entailment pairs). The pairs collected for the test set are provided in their unlabeled form as input to the TE system. To correctly train the TE system we balanced the data set with respect to the percentage of yes/no judgments. In Wikipedia, the actual distribution of attacks and supports among revisions of the same sentence is slightly unbalanced since generally users edit a sentence to add different information or correct it, with respect to a simple reformulation.[12]

To assess the validity of the annotation task and the reliability of the obtained data set, the same annotation task has been independently carried out also by a second annotator, so as to compute inter-annotator agreement. It has been calculated on a sample of 140 argument pairs (randomly extracted). The statistical measure usually used in NLP to calculate the inter-rater agreement for categorical items is Cohen's kappa coefficient [4], that is generally thought to be a more robust measure than simple percent agreement calculation since κ takes into account the agreement occurring by chance. More specifically, Cohen's kappa measures the agreement between two raters who each classifies N items into C mutually exclusive categories. The equation for κ is:

$$\kappa = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)} \tag{1}$$

where Pr(a) is the relative observed agreement among raters, and Pr(e) is the hypothetical probability of chance agreement, using the observed data to calculate the probabilities of each observer randomly saying each category. If the raters are in complete agreement then $\kappa = 1$. If there is no agreement among the raters other than what would be expected by chance (as defined by Pr(e)), $\kappa = 0$. For NLP tasks, the inter-annotator agreement is considered as significant when $\kappa > 0.6$. Applying the formula (1) to our data, the inter-annotator agreement results in $\kappa = 0.7$. As a rule of thumb, this is a satisfactory agreement, therefore we consider these annotated data sets as the *goldstandard*[13], i.e. the reference data set to which the performances of our combined system are compared. As introduced before, the goldstandard pairs have then been further translated into RDF using SIOC Argumentation.[14]

4.2 TE System

To detect which kind of relation underlies each couple of arguments, we use the EDITS system (Edit Distance Textual Entailment Suite) version 3.0, an open-source software package for RTE[15] [12]. EDITS implements a distance-based framework which assumes that the probability of an entailment relation

[12] As introduced before, we set a threshold in our extraction procedure to filter out all the minor revisions, concerning typos or grammatical mistakes corrections.

[13] The dataset is available at `http://bit.ly/WikipediaDatasetXML`

[14] The obtained data set is downloadable at `http://bit.ly/WikipediaDatasetRDF`

[15] `http://edits.fbk.eu/`

between a given T-H pair is inversely proportional to the distance between T and H (i.e. the higher the distance, the lower is the probability of entailment).[16] Within this framework the system implements different approaches to distance computation, i.e. both edit distance and similarity algorithms. Each algorithm returns a normalized distance score (a number between 0 and 1). At a training stage, distance scores calculated over annotated T-H pairs are used to estimate a threshold that best separates positive from negative examples, that is then used at a test stage to assign a judgment and a confidence score to each test pair.

4.3 Evaluation

To evaluate our framework, we carry out a two-step evaluation: first, we assess the performances of EDITS to correctly assign the *entailment* and the *no entailment* relations to the pairs of arguments on the Wikipedia data set. Then, we evaluate how much such performances impact on the application of the argumentation theory module, i.e. how much a wrong assignment of a relation to a pair of arguments is propagated in the argumentation framework. For the first evaluation, we run EDITS on the Wikipedia training set to learn the model, and we test it on the test set. In the configurations of EDITS we experimented, the distance entailment engine applies *cosine similarity* and *word overlap* as the core distance algorithms. In both cases, distance is calculated on lemmas, and a stopword list is defined to have no distance value between stopwords.

Table 1. Systems performances on Wikipedia data set

		Train			**Test**		
EDITS configurations	*rel*	*Precision*	*Recall*	*Accuracy*	*Precision*	*Recall*	*Accuracy*
WordOverlap	*yes*	0.83	0.82	**0.83**	0.83	0.82	**0.78**
	no	0.76	0.73		0.79	0.82	
CosineSimilarity	*yes*	0.58	0.89	0.63	0.52	0.87	0.58
	no	0.77	0.37		0.76	0.34	

Obtained results are reported in Table 1. Due to the specificity of our data set (i.e. it is composed by revisions of arguments), *word overlap* algorithm outperforms *cosine similarity* since there is high similarity between revised and original arguments (in most of the positive examples the two sentences are very close, or there is an almost perfect inclusion of H in T). For the same reason, obtained results are higher than in [2], and than the results obtained on average in RTE challenges. For these runs, we use the system off-the-shelf, applying its basic

[16] In previous RTE challenges, EDITS always ranked among the 5 best participating systems out of an average of 25 systems, and is one of the two RTE systems available as open source `http://aclweb.org/aclwiki/index.php?title=Textual_Entailment_Resource_Pool`

configuration. As future work, we plan to fully exploit EDITS features, integrating background and linguistic knowledge in the form of entailment rules, and to calculate the distance between T and H based on their syntactic structure.

As a second step in our evaluation phase, we consider the impact of EDITS performances (obtained using word overlap, since it provided the best results) on the acceptability of the arguments, i.e. how much a wrong assignment of a relation to a pair of arguments affects the acceptability of the arguments in the argumentation framework. We use admissibility-based semantics [9] to identify the accepted arguments both on the correct argumentation frameworks of each Wikipedia revised argument (where entailment/contradiction relations are correctly assigned, i.e. the goldstandard), and on the frameworks generated assigning the relations resulted from the TE system judgments. The precision of the combined approach we propose in the identification of the accepted arguments is on average 0.90 (i.e. arguments accepted by the combined system and by the goldstandard w.r.t. a certain Wikipedia revised argument), and the recall is 0.92 (i.e. arguments accepted in the goldstandard and retrieved as accepted by the combined system). The F-measure (i.e. the harmonic mean of precision and recall) is 0.91, meaning that the TE system mistakes in relation assignment propagate in the argumentation framework, but results are still satisfying and foster further research in this direction. For this feasibility study, we use four Wikipedia versions, so the resulting AFs are generally composed by four couples of arguments connected by attacks or supports. Reduced AFs are produced when a certain argument is not revised in every Wikipedia version we considered, or when an argument is deleted in more recent versions. Using more revised versions will allow us to generate even more complex argumentation graphs.

5 Related Work

A few works investigate the use of Wikipedia revisions in NLP tasks. In Zanzotto and Pennacchiotti [19], two versions of Wikipedia and semi-supervised machine learning methods are used to extract large TE data sets, while Cabrio et al. [1] propose a methodology for the automatic acquisition of large scale context-rich entailment rules from Wikipedia revisions. [18] focus on using edit histories in Simple English Wikipedia to extract lexical simplifications. Nelken and Yamangil [17] compare different versions of the same document to collect users' editorial choices, for automated text correction and text summarization systems. Max and Wisniewski [14] create a corpus of natural rewritings (e.g. spelling corrections, reformulations) from French Wikipedia revisions. Dutrey et al. [10] analyze part of this corpus to define a typology of local modifications.

Other approaches couple NLP and argumentation. Chasnevar and Maguitman [7] use defeasible argumentation to assist the language usage assessment. Their system provides recommendations on language patterns and defeasible argumentation. No natural language techniques are applied to automatically detect and generate the arguments. Carenini and Moore [3] present a complete computational framework for generating evaluative arguments. The framework,

based on the user's preferences, produces the arguments following the guidelines of argumentation theory to structure and select evaluative arguments. Differently from their work, we do not use natural language generation to produce the arguments, but we use TE to detect the arguments in natural language text. We use the word "generation" with the meaning of generation of the abstract arguments from the text, and not with the meaning of NL generation. Wyner and van Engers [16] present a policy making support tool based on forums. They propose to couple NLP and argumentation to provide the set of well structured statements that underlie a policy. Beside the goals, several points distinguish the two works: *i)* their NLP module guides the user in writing the text using a restricted grammar and vocabulary, while we have no lexicon or grammar restrictions; *ii)* the inserted statements are associated with a mode indicating the relation between the existing and the input statements. We do not ask the user to explicit the relation among the arguments, we infer them using TE; *iii)* no evaluation of their framework is provided. Heras et al. [11] show how to model the opinions on business oriented websites using argumentation schemes. We share the same goal (i.e. providing a formal structure to on-line dialogues for evaluation,), but in our proposal we achieve it using an automatic technique to generate the arguments from natural language texts as well as their relations.

6 Conclusions

In this paper, we presented a framework to support community managers in managing argumentative discussions on wiki-like platforms. In particular, our approach proposes to automatically detect the natural language arguments and the relations among them, i.e., support or challenges, and then to organize the detected arguments in bipolar argumentation frameworks. This kind of representation helps community managers to understand the overall structure of the discussions and which are the winning arguments. Moreover, the generated data set is translated in RDF using an extension of the SIOC Argumentation vocabulary such that the discussions can be queried using SPARQL in order to discover further insightful information. The experimental evaluation shows that in 85% of the cases, the proposed approach correctly detects the accepted arguments.

SIOC[17] allows to connect the arguments to the users who propose them. This is important in online communities because it allows to evaluate the arguments depending on the expertise of their sources. In this paper, we do not represent users neither in the argumentation frameworks nor in the RDF representation of the discussions, and this is left as future work. Moreover, we plan to move from the crisp evaluation of the arguments' acceptability towards a more flexible evaluation where the expertise of the users proposing the arguments plays a role. As future work on the NLP side, we consider experimenting a TE system carrying out a three-way judgment task (i.e. *entailment*, *contradiction* and *unknown*), to allow for a finer-grained classification of non entailment pairs (i.e. to separate when T contradicts H, from when H is more informative than T).

[17] `http://sioc-project.org`

References

1. Cabrio, E., Magnini, B., Ivanova, A.: Extracting context-rich entailment rules from wikipedia revision history. In: The People's Web Meets NLP Workshop (2012)
2. Cabrio, E., Villata, S.: Natural language arguments: A combined approach. In: European Conference on Artificial Intelligence (ECAI), pp. 205–210 (2012)
3. Carenini, G., Moore, J.D.: Generating and evaluating evaluative arguments. Artificial Intelligence 170(11), 925–952 (2006)
4. Carletta, J.: Assessing agreement on classification tasks: the kappa statistic. Computational Linguistics 22(2), 249–254 (1996)
5. Castro, A.G., Norena, A., Betancourt, A., Ragan, M.A.: Cognitive support for an argumentative structure during the ontology development process. In: Intl. Protege Conference (2006)
6. Cayrol, C., Lagasquie-Schiex, M.-C.: Bipolarity in argumentation graphs: Towards a better understanding. In: Benferhat, S., Grant, J. (eds.) SUM 2011. LNCS, vol. 6929, pp. 137–148. Springer, Heidelberg (2011)
7. Chesñevar, C.I., Maguitman, A.: An argumentative approach to assessing natural language usage based on the web corpus. In: European Conference on Artificial Intelligence (ECAI), pp. 581–585 (2004)
8. Dagan, I., Dolan, B., Magnini, B., Roth, D.: Recognizing textual entailment: Rational, evaluation and approaches. JNLE 15(04), i–xvii (2009)
9. Dung, P.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence 77(2), 321–358 (1995)
10. Dutrey, C., Bouamor, H., Bernhard, D., Max, A.: Local modications and paraphrases in wikipedia's revision history. SEPLN Journal 46, 51–58 (2011)
11. Heras, S., Atkinson, K., Botti, V.J., Grasso, F., Julián, V., McBurney, P.: How argumentation can enhance dialogues in social networks. In: Computational Model of Arguments (COMMA), pp. 267–274 (2010)
12. Kouylekov, M., Negri, M.: An open-source package for recognizing textual entailment. In: ACL System Demonstrations, pp. 42–47 (2010)
13. Lange, C., Bojars, U., Groza, T., Breslin, J., Handschuh, S.: Expressing argumentative discussions in social media sites. In: SDoW (2008)
14. Max, A., Wisniewski, G.: Mining naturally-occurring corrections and paraphrases from wikipedia's revision history. In: LREC (2010)
15. Schneider, J., Groza, T., Passant, A.: A review of argumentation for the social semantic web. Semantic Web J. (2011)
16. Wyner, A., van Engers, T.: A framework for enriched, controlled on-line discussion forums for e-government policy-making. In: eGov (2010)
17. Yamangil, E., Nelken, R.: Mining wikipedia revision histories for improving sentence compression. In: Proc. of ACL (Short Papers), pp. 137–140 (2008)
18. Yatskar, M., Pang, B., Danescu-Niculescu-Mizil, C., Lee, L.: For the sake of simplicity: Unsupervised extraction of lexical simplifications from wikipedia. In: HLT-NAACL, pp. 365–368 (2010)
19. Zanzotto, F., Pennacchiotti, M.: Expanding textual entailment corpora from wikipedia using co-training. In: The People's Web Meets NLP Workshop (2010)

A Multilingual Semantic Wiki Based on Attempto Controlled English and Grammatical Framework

Kaarel Kaljurand[1] and Tobias Kuhn[1,2]

[1] Institute of Computational Linguistics, University of Zurich, Switzerland
[2] Chair of Sociology, in particular of Modeling and Simulation, ETH Zurich, Switzerland
{kaljurand,kuhntobias}@gmail.com

Abstract. We describe a semantic wiki system with an underlying controlled natural language grammar implemented in Grammatical Framework (GF). The grammar restricts the wiki content to a well-defined subset of Attempto Controlled English (ACE), and facilitates a precise bidirectional automatic translation between ACE and language fragments of a number of other natural languages, making the wiki content accessible multilingually. Additionally, our approach allows for automatic translation into the Web Ontology Language (OWL), which enables automatic reasoning over the wiki content. The developed wiki environment thus allows users to build, query and view OWL knowledge bases via a user-friendly multilingual natural language interface. As a further feature, the underlying multilingual grammar is integrated into the wiki and can be collaboratively edited to extend the vocabulary of the wiki or even customize its sentence structures. This work demonstrates the combination of the existing technologies of Attempto Controlled English and Grammatical Framework, and is implemented as an extension of the existing semantic wiki engine AceWiki.

Keywords: semantic wiki, multilinguality, controlled natural language, Attempto Controlled English, Grammatical Framework.

1 Introduction

Wikis are user-friendly collaborative environments for building knowledge bases in natural language. The most well-known example is Wikipedia, an encyclopedia that is being built by around 100,000 users in hundreds of different languages, with numerous other wikis for smaller domains. Semantic wikis [4] combine the main properties of wikis (ease of use, read-write, collaboration, linking) with knowledge engineering technology (structured content, knowledge models in the form of ontologies, automatic reasoning). Semantic wiki editors simultaneously work with the natural language content and its underlying formal semantics representation. The resulting wikis offer more powerful content management functions, e.g. dynamically created pages based on semantic queries and detection of semantic errors in the content, but have to somehow meet the challenge of keeping the user interface as simple as expected from wikis. The existing semantic wiki engines (e.g. Semantic Mediawiki[1], Freebase[2]) support the

[1] http://semantic-mediawiki.org/
[2] http://www.freebase.com/

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 427–441, 2013.

inclusion of semantics in the form of RDF-like subject-predicate-object triples, e.g. typed wikilinks (predicates) between two articles (the subject and the object).

Our approach to semantic wikis is based on controlled natural language (CNL) [30]. A CNL is a restricted version of a natural language. For CNLs like Attempto Controlled English (ACE) [9], the syntax is precisely defined, the sentences have a formal (executable) meaning, and they come with end-user documentation describing syntax, semantics and usage patterns. CNLs and their editing tools support the creation of texts that are natural yet semantically precise, and can thus function well in human-machine communication. CNL-based wikis — such as AceWiki [16], on which our approach is based — can offer greater semantic expressivity compared to traditional semantic wikis (e.g. OWL instead of RDF), but their user interface is easier to work with (because it is still based on natural language).

In this paper we describe a semantic wiki system with an underlying controlled natural language grammar implemented in Grammatical Framework (GF). The grammar restricts the wiki editors into a well-defined subset of ACE that is automatically translatable into the Web Ontology Language (OWL) [11] and thus enables automatic semantic reasoning over the wiki content. Additionally, the grammar facilitates a precise bidirectional automatic translation between ACE and language fragments of a number of other natural languages. The developed wiki environment thus allows users to build, query and view OWL knowledge bases via a user-friendly multilingual natural language interface. The underlying multilingual grammar is integrated into the wiki itself and can be collaboratively edited to extend the vocabulary and even customize the multilingual representations of ACE sentences. Our work demonstrates the combination of the existing technologies of ACE and GF, and is implemented by extending the existing ACE-based semantic wiki engine AceWiki with support to multilinguality and collaborative GF grammar editing. The main goal of this work is to explore natural language grammar based semantic wikis in the multilingual setting. As a subgoal we ported a fragment of ACE to several natural languages (other than English) in a principled way by implementing a shared abstract syntax. The wiki environment allows us to test the usefulness of this work and furthermore collaboratively improve the initial ports. The overall work is part of the the EU research project MOLTO[3].

This paper is structured as follows: in Section 2 we review related work; in Section 3 we introduce the core features of the existing tools and technologies employed in the rest of the paper (namely ACE, GF and AceWiki); in Section 4 we describe the multilingual GF-implementation of ACE; in Section 5 we discuss the extension of AceWiki based on the GF-implementation of ACE; in Section 6 we provide an initial evaluation of our system; in Section 7 we summarize our main results and outline future work.

2 Related Work

The related work falls into several categories such as multilingual CNLs, CNL-based wikis, multilingual wikis, multilingual ontologies, and ontology verbalization.

Many general purpose and domain-specific controlled natural languages have been developed based on many different natural languages [24]. However, there has not been

[3] http://www.molto-project.eu

an effort to bring them under the same semantic model or synchronize their development in a community-driven manner [22]. Our multilingual ACE grammar is an experiment in this direction. A multilingual version of ACE (in GF) was first investigated in [28]. Our current implementation is partly an extension of this work. A similar work is [12], which builds a bidirectional interface between a controlled fragment of Latvian and OWL, using ACE as an interlingua, and implementing the interface using GF.

The main CNL-based wiki that we are aware of is AceWiki which is also the basis of our work and will be discussed below. [10] describes the MoKi semantic wiki engine which offers a "lightly-structured access mode" for its structured content (OWL). In this mode the content is displayed as an uneditable ACE text; editing is supported for the simpler *isA* and *partOf* statements using templates that combine CNL with HTML-forms, or using a native OWL syntax. In terms of multilinguality our wiki system has some similarities with the OWL ontology editor described in [2] which allows the user to view the ontology in three CNLs, two based on English and one on Chinese. As the main difference compared to these systems, our system uses the CNLs as the only user interface for both editing and viewing.

The research on GF has not yet focused on a wiki-like tool built on top of a GF-based grammar or application. Tool support exists mostly for users constructing single sentences (not texts) and working alone (not in collaboration). A notable exception is [23], which investigates using GF in a multilingual wiki context, to write restaurant reviews on the abstract language-independent level by constructing GF abstract trees.

Even though the mainstream wiki engines generally allow for the wiki articles to be written in multiple languages, these different language versions exist independently of each other and only article-level granularity is offered by the system for interlinking the multilingual content. Some recent work targets that problem though, e.g. the EU project CoSyne[4] develops a technology for the multilingual content synchronization in wikis by using machine translation.

Ontology languages (such as RDF, OWL and SKOS) typically support language-specific labels as attachments to ontological entities (such as classes and properties). Although the ontological axioms can thus be presented multilingually, their keywords (e.g. *SubClassOf*, *some*, *only*) are still in English and their syntactic structure is not customizable. This is clearly insufficient for true ontology verbalization, especially for expressive ontology languages like OWL as argued in [6], which describes a sophisticated lexical annotation ontology to be attached to the domain ontology as linguistic knowledge. Our work can also be seen as attaching (multilingual) linguistic knowledge to a semantic web ontology. [7] discusses a multilingual CNL-based verbalization of business rules. It is similar to our approach by being implemented in GF but differs by not using OWL as the ontology language.

3 Underlying Technologies

3.1 Attempto Controlled English

Attempto Controlled English (ACE) [9] is a general purpose CNL based on first-order logic. ACE can be viewed as both a natural language understandable to every English

[4] `http://www.cosyne.eu/`

speaker, as well as a formal language with a precisely defined syntax and semantics understandable to automatic theorem proving software. ACE offers many language constructs, the most important of which are countable and mass nouns (e.g. 'man', 'water'); proper names ('John'); generalized quantifiers ('at least 2'); indefinite pronouns ('somebody'); intransitive, transitive and ditransitive verbs ('sleep', 'like', 'give'); negation, conjunction and disjunction of noun phrases, verb phrases, relative clauses and sentences; and anaphoric references to noun phrases through definite noun phrases, pronouns, and variables. Texts built from these units are deterministically interpreted via Discourse Representation Structures (DRS) [15], which can be further mapped to formats supported by existing automatic reasoners (e.g. OWL, SWRL, FOL, TPTP). The ACE sentence structures and their unambiguous interpretations are explained in the end-user documentation in the form of *construction* and *interpretation* rules.

The grammar of ACE and its mapping to DRS cannot be modified by the end-users but they can customize ACE in their applications by specifying a content word lexicon of nouns, verbs, adjectives, adverbs and prepositions and their mapping to logical atoms.

While originally designed for software specifications, in the recent years ACE has been developed with the languages and applications of the Semantic Web in mind. [13] describes ACE fragments suitable for mapping to and from languages like OWL, SWRL and DL-Query. ACE View [14] and AceWiki are ACE-based tools for building OWL ontologies. The study described in [20] provides evidence that ACE is a user-friendly language for specifying OWL ontologies, providing a syntax that is easier to understand and use compared to the standard OWL syntaxes.

3.2 Grammatical Framework

Grammatical Framework (GF) [27] is a functional programming language for building multilingual grammar applications. Every GF program consists of an *abstract syntax* (a set of functions and their categories) and a set of one or more *concrete syntaxes* which describe how the abstract functions and categories are linearized (turned into surface strings) in each respective concrete language. The resulting grammar describes a mapping between concrete language strings and their corresponding abstract trees (structures of function names). This mapping is bidirectional — strings can be *parsed* to trees, and trees *linearized* to strings. As an abstract syntax can have multiple corresponding concrete syntaxes, the respective languages can be automatically *translated* from one to the other by first parsing a string into a tree and then linearizing the obtained tree into a new string.

While GF can be used to build parsers and generators for formal languages, it is optimized to handle natural language features like morphological variation, agreement, and long-distance dependencies. Additionally, the GF infrastructure provides a *resource grammar library* (RGL), a reusable grammar library of the main syntactic structures and morphological paradigms currently covering about 30 natural languages [26]. As the library is accessible via a language-independent API, building multilingual applications remains simple even if the programmers lack detailed knowledge of the linguistic aspects of the involved languages. These features make GF a good framework for the implementation of CNLs, especially in the multilingual setting [29]. The development of GF has focused on parsing tools, grammar editors, and extending the grammar

library to new languages. The current algorithm for parsing GF grammars is based on Parallel Multiple Context-Free Grammars and allows for incremental parsing, which enables look-ahead editing [1].

3.3 AceWiki

AceWiki[5] [18] is a CNL-based semantic wiki engine, implemented in Java using the Echo Web Framework[6]. It uses ACE as the content language and OWL as its underlying semantic framework integrating its main reasoning tasks (consistency checking, classification and query answering) and making them available via the ACE-based interface.

The content of an AceWiki instance is written in a subset of ACE formally defined in a grammar notation called Codeco [19]. The grammar targets an OWL-compatible fragment of ACE, i.e. ACE sentences that are semantically outside of the OWL expressivity cannot be expressed in the wiki. This guarantees that all of the AceWiki content can be automatically translated to OWL in the background. Additionally, the grammar is used to drive a look-ahead editor which guides the input of a new sentence by proposing only syntactically legal continuations of the sentence.

The AceWiki content is structured into a set of articles, each article containing a sequence of entries which are either declarative sentences (corresponding to OWL axioms) or questions (corresponding to OWL class expressions). Additionally informal comments are supported. Upon every change in the wiki, an OWL reasoner determines its effect and possibly flags inconsistencies or updates the dynamically generated parts of the wiki (e.g. concept hierarchies and answers to questions).

The content words (proper names, nouns, transitive verbs, relational nouns and transitive adjectives) in the wiki sentences map one-to-one (i.e. link) to wiki articles. Semantically, content words correspond to OWL entities: proper names to OWL individuals, nouns to OWL classes, and the relational words to OWL properties.

4 Multilingual ACE

In order to provide a multilingual interface to AceWiki, we implemented the syntax of ACE in GF and ported it via the RGL API to multiple natural language fragments. (See the ACE-in-GF website[7] and [5] for more details of this work.) On the ACE side, the grammar implements the subset supported by the AceWiki Codeco grammar and can be thus automatically tested against the Codeco implementation to verify the coverage and precision properties. The implementation accesses the GF English resource grammar through the language-independent API (Figure 1). This API makes it easy to plug in other RGL-supported languages. Our current implementation targets 15 European languages. Most of them provide full coverage of the ACE syntactic structures, for some languages a few structures (e.g. verb phrase coordination, some forms of questions) have not been implemented yet.

[5] `http://attempto.ifi.uzh.ch/acewiki/`
[6] `http://echo.nextapp.com/`
[7] `http://github.com/Attempto/ACE-in-GF`

```
-- ACE noun phrase uses the RGL noun phrase structure
lincat NP = Syntax.NP ;
...
-- noun phrase with the determiner 'every' e.g. 'every country'
lin  everyNP = Syntax.mkNP every_Det ;
...
-- verb phrase with a passive transitive verb and a noun phrase
-- e.g. 'bordered by Germany'
lin v2_byVP v2 np = mkVP (passiveVP v2) (Syntax.mkAdv by8agent_Prep np) ;
```

Fig. 1. Fragment of a GF grammar for ACE listing the linearization rules for the functions `everyNP` and `v2_byVP`. There are around 100 such rules. This GF module (functor) implements the ACE sentence structures via RGL's API calls (e.g. `every_Det`, `mkVP`). A concrete language implementation parametrizes this functor with a concrete language resource (English in case of ACE) and possibly overrides some of the rules with language-specific structures. For the function categories, the grammar uses categories that are also used in the ACE user-level documentation, e.g. noun (`N`), transitive verb (`V2`), noun phrase (`NP`), relative clause.

While most of the multilingual grammar can be written in a language-neutral format, the lexicon modules are language dependent. Different languages have different morphological complexity, e.g. while the Codeco-defined AceWiki subset of ACE operates with two noun forms (singular and plural) and three verb forms (infinitive, 3rd person singular and past participle), other languages (e.g. Finnish) might need many more forms to be used in the various ACE sentence structures. Fortunately, we can make use of the RGL calls, e.g. `mkN` ("make noun") and `mkV2` ("make transitive verb"), to create the necessary (language-specific) lexicon structures from a small number of input arguments (often just the lemma form), using the so called smart paradigms [26].

In order to view an ACE text in another language, one needs to parse it to an abstract tree which can then be linearized into this language (Figure 2). This makes it possible to map various natural language fragments to the formal languages that are supported by ACE (e.g. OWL and TPTP) and verbalize such formal languages via ACE (if this is supported) into various natural language fragments (Figure 3). For example, the OWL-to-ACE verbalizer [13] can be used as a component in a tool that makes an OWL ontology viewable in a natural language, say Finnish. This tool must contain a lexicon, i.e. a mapping of each OWL entity to the Finnish word that corresponds to the ACE category that the verbalizer assigns to the OWL entity.

While the ACE concrete syntax is designed to be unambiguous, i.e. every supported sentence generates just a single abstract tree, the grammar in general does not guarantee this property for the other implemented languages. In some cases it seems to be better to let a user work with an ambiguous representation if it offers a simpler syntax and if the ambiguity can be always explained (e.g. via the ACE representation) or removed in the actual usage scenario (e.g. in a collaborative wiki environment).

5 AceWiki-GF

Our multilingual semantic wiki based on ACE and GF has been realized as an extension of AceWiki, and is thus (preliminarily) called AceWiki-GF. Extending AceWiki

```
if_thenS
  (vpS
    (termNP X_Var) (v2VP contain_V2 (termNP Y_Var)))
  (neg_vpS
    (termNP Y_Var) (v2VP contain_V2 (termNP X_Var)))

ACE:     if X contains Y then Y does not contain X
Dutch:   als X Y bevat , dan bevat Y niet X
Finnish: jos X sisältää Y:n niin Y ei sisällä X:ää
German:  wenn X Y enthält , dann enthält Y X nicht
Spanish: si X contiene Y entonces Y no contiene X
```

Fig. 2. Abstract tree and its linearizations into five languages which express the OWL asymmetric property axiom, which is assigned by the ACE-to-OWL mapping to the ACE sentence. The linearizations feature different word orders depending on the language. The tree abstracts away from linguistic features like word order, case, and gender, although it still operates with syntactic notions such as *negated verb phrase*.

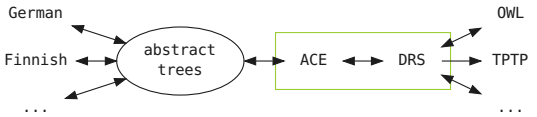


Fig. 3. Bidirectional mapping between a formal language like OWL and a natural language like Finnish facilitated by the multilingual GF-implementation of ACE and various mappings between ACE and other formal languages.

has allowed us to reuse its infrastructure (such as look-ahead editing, access to OWL reasoners, the presentation of reasoning results, and document navigation). In the following we only describe the main differences and extensions. (See Section 3.3 for the general discussion of the AceWiki engine.)

Because AceWiki is a monolingual engine, several modifications had to be done to accommodate multilinguality, i.e. to support viewing/editing in multiple languages depending on the users' preferences:

- the Codeco grammar/parser for ACE was replaced by the GF-implemented multilingual ACE grammar and a GF parser;
- the English-specific lexicon editor was replaced by a simple GF source editor which can be used to edit any GF grammar modules, among them lexicon modules;
- the atomic wiki entry, which for the monolingual AceWiki was an ACE sentence, was changed to a GF abstract tree set. The new representation is language-neutral and can furthermore represent ambiguity, as explained in Section 4;
- the notion of wiki article/page was extended to also include arbitrarily named pages (in AceWiki all pages are named by their corresponding OWL entity) and pages that represent editable grammar modules.

The existing AceWiki user interface has largely been preserved; the main additions are the disambiguation dialog and a menu for setting the content language, which also

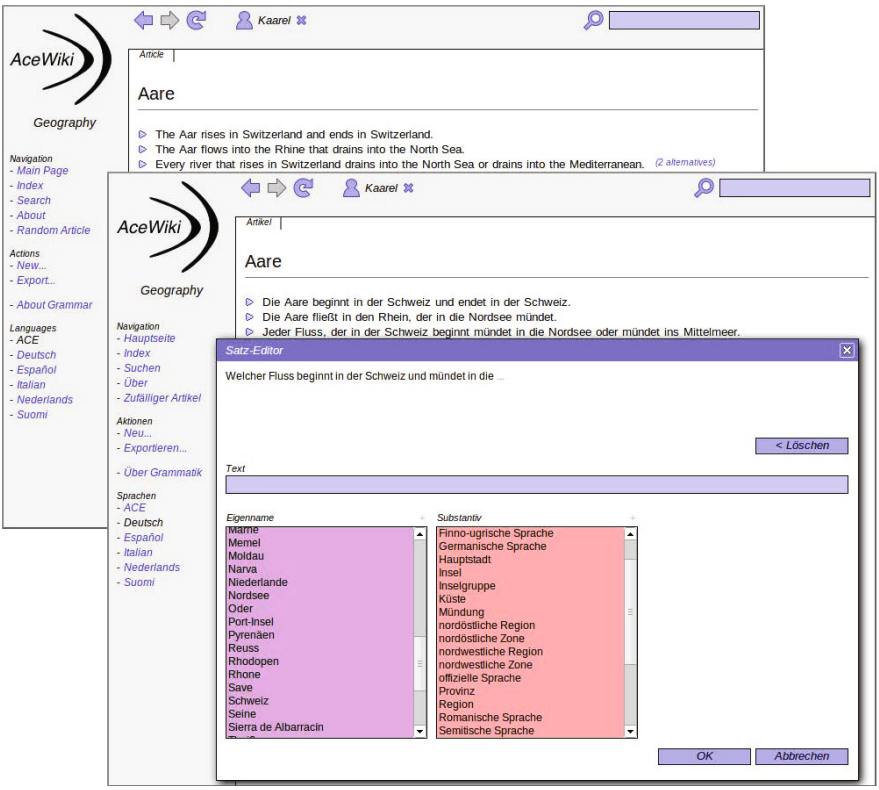


Fig. 4. Multilingual geography article displayed in ACE and German. The wiki language (of both the content and the user interface) can be changed in the left sidebar. Otherwise the user interface is the same as in AceWiki, with the look-ahead editor that helps to input syntactically controlled sentences, in this case offering proper names and common nouns as possible continuations.

determines the user interface language (Figure 4). The wiki still follows the main principle of CNL-based wikis, i.e. that formal notations are hidden. In our case the user does not see the trees which actually define the content but only interacts with the natural language sentences. (Experienced users can still look at the GF parser output providing information on syntax trees, translation alignment diagrams, etc.)

5.1 Structure and Linking

In general, AceWiki-GF follows the AceWiki structure — the wiki is a set of articles, each containing a sequence of sentences. New is the fact that also the grammar definition is part of the wiki and can be referenced from the articles using wikilinks.

A GF grammar is structured in a way that is naturally representable as a set of wiki articles. Each grammar module can be stored as a wiki article and linked to the modules that it imports. Furthermore, grammar modules have internal structure — sets of

categories and functions (which reference categories) — which can be linked to wiki content because the content is represented as a set of trees (i.e. structures of function names). One of the benefits of having a grammar definition as part of the wiki is that it provides an integrated documentation of the language that the wiki users are required to use. Note that the full grammar contains also modules which are part of the general RGL and thus not editable and also not part of the wiki. This resource is made accessible via external links to the online RGL browser[8].

5.2 Sentence Editing

The user interface for adding and modifying wiki entries is the same as in AceWiki, i.e. based on sentences and supporting the completion of a syntactically correct sentence by displaying a list of syntactically legal words that can follow the partially completed sentence. The language of the sentence depends on the chosen wiki language. In case an entry is ambiguous (i.e. parsing results in multiple trees) then the ambiguity is preserved. If viewed in another language, multiple different sentences can then occur as linearizations of the ambiguity. This allows the wiki users who work via the other language to resolve the ambiguity. A monolingual way to deal with ambiguity is to implement for every concrete syntax an additional "disambiguation syntax" [29], that overrides the linearizations of the ambiguous constructs to have an unambiguous, although possibly a more formal-looking notation. This syntax could be used to display the entry in the case of ambiguity.

We note that some syntax-aware editors, e.g. the GF Syntax Editor[9] or the OWL Simplified English editor [25], operate more on the abstract tree level and thus avoid the problem of ambiguous entries. These approaches also simplify smaller edits e.g. replacing a word in the beginning of the sentence. The fact that they abstract away from linguistic details like case and gender might make them preferable for users with only basic knowledge of the underlying language. It is therefore worth exploring these editing approaches also in the AceWiki-GF context.

5.3 Lexicon and Grammar Editing

Our wiki makes the grammar available as a set of interlinked grammar modules falling into the following categories:

- ACE resource grammar (about 30 modules which are typically identical to their English resource grammar counterparts, sometimes overriding certain structures);
- ACE application grammar, reflecting the AceWiki subset of ACE (one module);
- instantiation of this grammar for each supported language with additional modules that describe language-specific overriding of some of the functions;
- content word lexicon module(s) for each language.

In order to add a new word to the wiki, a line needs to be added to the lexicon wiki page, i.e. the page that corresponds to the lexicon module (Figure 5). Although editing the

[8] `http://www.grammaticalframework.org/lib/doc/browse/`
[9] `http://cloud.grammaticalframework.org/syntax-editor/editor.html`

```
Danish:  country_N = mkN "land" "landet" ;
Dutch:   country_N = mkN "land" neuter ;
Finnish: country_N = mkN "maa" ;
French:  country_N = mkN "pays" masculine ;
German:  country_N = mkN "Land" "Länder" neuter ;
Italian: country_N = mkN "paese" ;
Swedish: country_N = mkN "land" "landet" "länder" "länderna" ;
```

Fig. 5. Entries in the multilingual lexicon. Smart paradigms like `mkN` are used to create the internal structure of the entry. In many cases giving only the lemma form to the word class operator is sufficient to get a correct internal structure. In some cases further forms or information about gender (in some languages) needs to be added. This makes the user interface to the lexicon relatively simple and homogeneous across languages.

lexicon technically means editing the GF grammar, the lexicon module is conceptually much simpler than the general grammar module and maps one-to-one to the respective ACE lexicon structure (for English). The structure of lexicons in all the supported languages is roughly the same even if some languages are morphologically more complex (e.g. have more case endings). The language-specific lexical structures are hidden from the user behind language-neutral categories like `N` and `V2` and constructed by functions like `mkN` and `mkV2` which are capable of determining the full word paradigm on the basis of only one or two input forms. Thus, support for multilinguality does not increase the conceptual complexity of the wiki.

Wiki users experienced in GF are also able to modify the full grammar, although we do not see many compelling use cases for that as ACE itself is pre-defined and thus changing its grammar should not be allowed (e.g. it would break the functioning of the mapping to OWL). Its verbalization to other languages, however, is sometimes a matter of taste, and could be therefore made changeable by the wiki users, e.g. users can add an alternative formulation of an ACE sentence in some language by using a GF variant. Also, the possibility to define arbitrary GF operators can make certain lexicon entry tasks more convenient.

A change to the underlying grammar (even if only in the lexicon module) can have the following consequences for the content: (1) removing a function can render some of the wiki entries (the ones whose trees use this function) invalid, the user must then reformulate the respective sentences to conform to the new grammar/lexicon; (2) altering the linearization of a function might cause some sentences to become unparsable or ambiguous in the corresponding language. This does not have an immediate effect on the stored wiki content because the storage is based on trees, but if an existing sentence is submitted again to the parser then it might fail or result in more trees than before. A general change to a grammar module (e.g. removing a category) can also make the whole grammar invalid, which obviously should be avoided.

5.4 Underlying Semantic Representation

As in the original AceWiki, each AceWiki-GF entry has a corresponding OWL representation. It is obtained by linearizing the abstract tree of the entry as an ACE sentence (using the multilingual grammar) and then translating it to OWL (as defined in [13]).

In the ACE representation each content word is annotated with its word class information and the corresponding OWL entity, which is currently derived from the lemma form of the ACE word. Ambiguous wiki entries map in general to multiple OWL forms (although this is not necessarily the case). Such entries are not included in the underlying semantic representation.

5.5 Multilinguality

The content of our wiki can be currently made available in up to 15 languages, which form a subset of the RGL that has been tested in the context of the multilingual ACE grammar. In principle every RGL language (that exists now or will be added to the RGL in the future) can be plugged in, because we access the RGL via its language-neutral API. However, language-specific customization of some of the phrase structures is usually necessary as discussed in Section 4.

For a concrete wiki instance a smaller number of languages might be initially preferred and more translations of the wiki content could be added gradually. In addition to the wiki reader and the wiki editor, there is now a third type of a wiki user, namely the translator. Their main task is to translate all existing words by referencing the correct operators in the RGL morphological API and to check if the automatically generated translations are accurate with respect to ACE. The skillset for this task therefore includes the knowledge of ACE and the RGL morphology API.

5.6 Implementation

Apart from having been implemented as an extension of AceWiki, the discussed wiki engine is supported by two external (and independently developed) tools. First, the GF Webservice [3] provides linearization and parsing (and the related look-ahead) services for GF grammars. The GF Webservice has been recently extended to provide a GF Cloud Service API[10] which additionally allows for modifications to the grammar. Secondly, the ACE parser APE[11] provides the mapping of ACE sentences to the OWL form (as is the case also for the monolingual AceWiki). The current implementation of AceWiki-GF is available on GitHub[12] and can be used via some demo wikis[13].

6 Evaluation

In previous work, two usability experiments have been performed on AceWiki with altogether 26 participants [17]. The results showed that AceWiki and its editor component are easy to learn and use. Another study confirmed that writing ACE sentences with the editor is easier and faster than writing other formal languages [21]. It has also been demonstrated that ACE is more effective than the OWL Manchester Syntax in terms

[10] `http://cloud.grammaticalframework.org/gf-cloud-api.html`
[11] `http://github.com/Attempto/APE`
[12] `http://github.com/AceWiki/AceWiki`
[13] `http://attempto.ifi.uzh.ch/acewiki-gf/`

of understandability [20]. As these previous studies did not include the multilinguality features, the evaluations presented below focus on the multilingual grammar aspects.

We first evaluated how many syntactically correct sentences of the AceWiki ACE subset the multilingual grammar accepts. To that aim, we used the AceWiki Codeco testset which is an exhaustive set of sentences with length of up to 10 tokens (19,422 sentences, disregarding some deprecated ACE sentences) [18]. The GF-based ACE grammar successfully covers all these sentences.

Next, we measured the syntactic precision by randomly generating large numbers of sentences at different tree depths and parsing them with both the ACE parser and the Codeco parser. The precision of the grammar was found to be sufficient although not perfect. The main deficiency compared to the Codeco grammar is the lack of DRS-style anaphoric reference modeling. In practice this means that some accepted sentences will be rejected by the ACE-to-OWL translator on the grounds of containing unresolvable definite noun phrases. Ignoring such sentences the precision was 98% (measured at tree depth of 4 for which the sentence length is 11 tokens on average).

The ambiguity level of ACE sentences (of the Codeco testset) was found to be 3%. In these relatively rare cases, involving complex sentences, the grammar assigns two abstract trees to an input ACE sentence. This is always semantically harmless ambiguity (i.e. it would not manifest itself in translations) resulting from the rules for common nouns and noun phrases which accept similar input structures. While the coverage and precision are measures applicable only to the ACE grammar (because an external definition and a reference implementation of ACE exists), the ambiguity can be measured for all the languages implemented in the grammar by linearizing trees in a given language and checking if the result produces additional trees when parsed. Some semantically severe ambiguities were found using this method (e.g. occasional subject/object relative clause ambiguity in Dutch and German triggered by certain combinations of case and gender, double negation ambiguity in some Romance languages). These findings can either be treated in the grammar i.e. in the design of the respective controlled languages or highlighted in the wiki environment in a way that they can be effectively dealt with.

To measure the translation quality we looked at the translations of 40 ACE sentences using 20 lexicon entries. The sentences were verbalizations of a wide variety of OWL axiom structures (also used in [20]). We wanted to check whether the meaning in all the languages adheres to the precise meaning of OWL statements. The translations covered nine languages (Catalan, Dutch, Finnish, French, German, Italian, Spanish, Swedish, and Urdu) and were checked by native speakers to evaluate the translations with respect to the original ACE sentence and the ACE interpretation rules [5]. In general, the translations were found to be acceptable and accurate although several types of errors were found, mainly caused by the fact that the lexicon creators were not very familiar with the respective languages. Concretely, the following four error types were observed:

RGL Errors. Some problems (e.g. missing articles in Urdu) were traced to errors in the resource grammar library, and not in our ACE application grammar.

Incorrect Use of Smart Paradigms. Several mistakes were caused by an incorrect use of the RGL smart paradigms, either by applying a regular paradigm to an irregular word or simply providing the operator with an incorrect input (e.g. a genitive form instead of a nominative).

Stylistic Issues. A further problem were stylistic issues, i.e. structures that are understandable but sound unnatural to a native speaker, e.g. using an inanimate pronoun to refer to a person.

Negative Determiners. We experienced that translating sentences with negative determiners such as 'no', e.g. "every man does not love no woman" or "no man does not love a woman" can result in meaning shifts between languages. This was eventually handled by extending the RGL to include noun phrase polarity.

Most of these problems are easy to fix by a native speaker with GF skills. (We assume that if there is enough interest in the port of a particular wiki into another language, it should be possible to find such a person.) A more conclusive evaluation is planned that includes the wiki environment and uses a larger real-world vocabulary.

7 Discussion and Future Work

The main contribution of our work is the study of CNL-based knowledge engineering in a semantic wiki environment. The main novelty with respect to previous work is making the wiki environment multilingual. As the underlying technologies we have used Attempto Controlled English, which is a syntactically user-friendly formal language and provides a mapping to the expressive ontology language OWL, and Grammatical Framework, which was used to provide a multilingual bidirectional interface to ACE covering several natural languages. We have built the implementation on top of AceWiki, an existing monolingual semantic wiki engine. In order to make our system multilingual, the architecture of AceWiki was generalized. Although the underlying implementation has become more complex, the user interface has largely remained the same. On the (multilingual) lexicon editing side, this is mainly due to the support for smart paradigms that GF provides via its RGL. In the future, we plan to use the grammar-based approach also to implement the other aspects of the wiki, such as multilingual user interface labels (see [23]).

The current approach generates the OWL representations using the existing ACE-to-OWL translator. An alternative method is to implement this translator also in GF. In this way the users would have full control over what kind of OWL axioms can be generated because they can edit the OWL mapping (concrete syntax) in the wiki. The semantic aspects of the wiki could also be generalized to allow for any kind of ACE-based reasoning, offered by tools like RACE [8] or TPTP reasoners.

The presented work can be also extended in various more general directions. Although the current system is ACE-based, its general architecture allows for any grammar to be used as the basis of the wiki content as long as it is implemented in GF. Such alternative grammars might not map naturally to a language like OWL and are thus less interesting in the context of the Semantic Web. Examples are grammars for a tourist phrase book, a museum catalog, a technical manual, or a collection of mathematics exercises. Such wikis would mainly profit from the supported multilinguality and not so much from semantic web style reasoning, or may need other forms of reasoning.

Another direction is to improve the grammar editing features of the environment and to develop the system into a tool for collaboratively designing CNLs. The wiki users could take e.g. the ACE grammar as starting point and customize it for a specific

domain, possibly changing some of its original features and design decisions. The wiki sentences could then serve as unit/regression test sets to check the currently effective grammar implementation.

Acknowledgments. The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement FP7-ICT-247914. The authors would like to thank Norbert E. Fuchs for useful comments on the draft of this paper.

References

1. Angelov, K.: The Mechanics of the Grammatical Framework. PhD thesis, Chalmers University of Technology (2011)
2. Bao, J., Smart, P.R., Shadbolt, N., Braines, D., Jones, G.: A Controlled Natural Language Interface for Semantic Media Wiki. In: 3rd Annual Conference of the International Technology Alliance, ACITA 2009 (September 2009)
3. Bringert, B., Angelov, K., Ranta, A.: Grammatical Framework Web Service. In: Proceedings of EACL 2009 (2009)
4. Bry, F., Schaffert, S., Vrandečić, D., Weiand, K.: Semantic wikis: Approaches, applications, and perspectives. Reasoning Web. Semantic Technologies for Advanced Query Answering, 329–369 (2012)
5. Camilleri, J.J., Fuchs, N.E., Kaljurand, K.: Deliverable D11.1. ACE Grammar Library. Technical report, MOLTO project (June 2012), `http://www.molto-project.eu/biblio/deliverable/ace-grammar-library`
6. Cimiano, P., Buitelaar, P., McCrae, J., Sintek, M.: Lexinfo: A declarative model for the lexicon-ontology interface. Web Semantics: Science, Services and Agents on the World Wide Web 9(1), 29–51 (2011)
7. Davis, B., Enache, R., van Grondelle, J., Pretorius, L.: Multilingual Verbalisation of Modular Ontologies Using GF and *lemon*. In: Kuhn, T., Fuchs, N.E. (eds.) CNL 2012. LNCS, vol. 7427, pp. 167–184. Springer, Heidelberg (2012)
8. Fuchs, N.E.: First-Order Reasoning for Attempto Controlled English. In: Rosner, M., Fuchs, N.E. (eds.) CNL 2010. LNCS, vol. 7175, pp. 73–94. Springer, Heidelberg (2012)
9. Fuchs, N.E., Kaljurand, K., Kuhn, T.: Attempto Controlled English for Knowledge Representation. In: Baroglio, C., Bonatti, P.A., Małuszyński, J., Marchiori, M., Polleres, A., Schaffert, S. (eds.) Reasoning Web 2005. LNCS, vol. 5224, pp. 104–124. Springer, Heidelberg (2008)
10. Ghidini, C., Rospocher, M., Serafini, L.: Modeling in a Wiki with MoKi: Reference Architecture, Implementation, and Usages. International Journal on Advances in Life Sciences 4 (2012)
11. W3C OWL Working Group. OWL 2 Web Ontology Language Document Overview (Second Edition). W3C Recommendation. Technical report, W3C (December 11, 2012), `http://www.w3.org/TR/owl2-overview/`
12. Grūzītis, N.: Formal Grammar and Semantics of Controlled Latvian Language. PhD thesis, University of Latvia (2011)
13. Kaljurand, K.: Attempto Controlled English as a Semantic Web Language. PhD thesis, Faculty of Mathematics and Computer Science, University of Tartu (2007)
14. Kaljurand, K.: ACE View — an ontology and rule editor based on Attempto Controlled English. In: 5th OWL Experiences and Directions Workshop (OWLED 2008), Karlsruhe, Germany, October 26-27, 12 pages (2008)

15. Kamp, H., Reyle, U.: From Discourse to Logic. Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory. Kluwer Academic Publishers, Dordrecht (1993)
16. Kuhn, T.: AceWiki: A Natural and Expressive Semantic Wiki. In: Semantic Web User Interaction at CHI 2008: Exploring HCI Challenges (2008)
17. Kuhn, T.: How Controlled English can Improve Semantic Wikis. In: Lange, C., Schaffert, S., Skaf-Molli, H., Völkel, M. (eds.) Proceedings of the Fourth Workshop on Semantic Wikis, European Semantic Web Conference 2009. CEUR Workshop Proceedings, vol. 464. CEUR-WS (June 2009)
18. Kuhn, T.: Controlled English for Knowledge Representation. PhD thesis, Faculty of Economics, Business Administration and Information Technology of the University of Zurich (2010)
19. Kuhn, T.: A principled approach to grammars for controlled natural languages and predictive editors. Journal of Logic, Language and Information 22(1), 33–70 (2013)
20. Kuhn, T.: The understandability of OWL statements in controlled English. Semantic Web 4(1), 101–115 (2013)
21. Kuhn, T., Höfler, S.: Coral: Corpus access in controlled language. Corpora 7(2), 187–206 (2012)
22. Luts, M., Tikkerbär, D., Saarmann, M., Kutateladze, M.: Towards a Community-Driven Controlled Natural Languages Evolution. In: Rosner, M., Fuchs, N.E. (eds.) Pre-Proceedings of the Second Workshop on Controlled Natural Languages, CNL 2010. CEUR Workshop Proceedings, vol. 622, CEUR-WS (2010)
23. Meza-Moreno, M.S., Bringert, B.: Interactive Multilingual Web Applications with Grammatical Framework. In: Nordström, B., Ranta, A. (eds.) GoTAL 2008. LNCS (LNAI), vol. 5221, pp. 336–347. Springer, Heidelberg (2008)
24. Pool, J.: Can Controlled Languages Scale to the Web? In: 5th International Workshop on Controlled Language Applications (2006)
25. Power, R.: OWL Simplified English: A Finite-State Language for Ontology Editing. In: Kuhn, T., Fuchs, N.E. (eds.) CNL 2012. LNCS, vol. 7427, pp. 44–60. Springer, Heidelberg (2012)
26. Ranta, A.: The GF Resource Grammar Library. Linguistic Issues in Language Technology 2(2) (2009)
27. Ranta, A.: Grammatical Framework: Programming with Multilingual Grammars. CSLI Publications, Stanford (2011), ISBN-10: 1-57586-626-9 (Paper), 1-57586-627-7 (Cloth)
28. Angelov, K., Ranta, A.: Implementing Controlled Languages in GF. In: Fuchs, N.E. (ed.) CNL 2009. LNCS, vol. 5972, pp. 82–101. Springer, Heidelberg (2010)
29. Ranta, A., Enache, R., Détrez, G.: Controlled Language for Everyday Use: The MOLTO Phrasebook. In: Rosner, M., Fuchs, N.E. (eds.) CNL 2010. LNCS, vol. 7175, pp. 115–136. Springer, Heidelberg (2012)
30. Wyner, A., Angelov, K., Barzdins, G., Damljanovic, D., Davis, B., Fuchs, N., Hoefler, S., Jones, K., Kaljurand, K., Kuhn, T., Luts, M., Pool, J., Rosner, M., Schwitter, R., Sowa, J.: On controlled natural languages: Properties and prospects. In: Fuchs, N.E. (ed.) CNL 2009. LNCS, vol. 5972, pp. 281–289. Springer, Heidelberg (2010)

COALA – Correlation-Aware Active Learning of Link Specifications

Axel-Cyrille Ngonga Ngomo, Klaus Lyko, and Victor Christen

Department of Computer Science
AKSW Research Group
University of Leipzig, Germany
{ngonga,klaus.lyko,christen}@informatik.uni-leipzig.de

Abstract. Link Discovery plays a central role in the creation of knowledge bases that abide by the five Linked Data principles. Over the last years, several active learning approaches have been developed and used to facilitate the supervised learning of link specifications. Yet so far, these approaches have not taken the correlation between unlabeled examples into account when requiring labels from their user. In this paper, we address exactly this drawback by presenting the concept of the correlation-aware active learning of link specifications. We then present two generic approaches that implement this concept. The first approach is based on graph clustering and can make use of intra-class correlation. The second relies on the activation-spreading paradigm and can make use of both intra- and inter-class correlations. We evaluate the accuracy of these approaches and compare them against a state-of-the-art link specification learning approach in ten different settings. Our results show that our approaches outperform the state of the art by leading to specifications with higher F-scores.

Keywords: Active Learning, Link Discovery, Genetic Programming.

1 Introduction

The importance of the availability of links for a large number of tasks such as question answering [20] and keyword search [19] as well as federated queries has been pointed out often in literature (see, e.g., [1]). Two main problems arise when trying to discover links between data sets or even deduplicate data sets. First, naive solutions to Link Discovery (LD) display a quadratic time complexity [13]. Consequently, they cannot be used to discover links across large datasets such as DBpedia[1] or Yago[2]. Time-efficient algorithms such as PPJoin+ [21] and $\mathcal{HR}^3$ [11] have been developed to address the problem of the a-priori quadratic runtime of LD approaches. While these approaches achieve practicable runtimes even on large datasets, they do not guarantee the quality of the links that are returned by LD frameworks. Addressing this second problem of LD demands

[1] http://dbpedia.org
[2] http://www.mpi-inf.mpg.de/yago-naga/yago/

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 442–456, 2013.

the development of techniques that can compute accurate *link specifications* (i.e., aggregations of atomic similarity or distance measures and corresponding thresholds) for deciding whether two resources should be linked. This problem is commonly addressed within the setting of machine learning. While both supervised (e.g., [15]) and unsupervised machine-learning approaches (e.g., [17]) have been proposed to achieve this goal, we focus on supervised machine learning.

One of the main drawbacks of supervised machine learning for LD lies in the large number of links necessary to achieve both a high precision and a high recall. This intrinsic problem of supervised machine learning has been addressed by relying on active learning [18]. The idea here is to rely on *curious classifiers*. These are supervised approaches that begin with a small number of labeled links and then inquire labels for data items that promise to improve their accuracy. Several approaches that combine genetic programming and active learning have been developed over the course of the last couple of years and shown to achieve high F-measures on the deduplication (see e.g., [4]) and LD (see e.g., [15]) problems. Yet, so far, none of these approaches has made use of the correlation between the unlabeled data items while computing the set of most informative items. In this paper, we address exactly this drawback.

The basic intuition behind this work is that we can provide a better approximation of the real information content of unlabeled data items by taking the similarity of unlabeled items into account. We call this paradigm the correlation-aware active learning of link specifications and dub it COALA. A better approximation should ensure that curious classifiers converge faster. Consequently, we should be able to reduce the number of data items that the user has to label manually. We thus present and evaluate two generic approaches that implement this intuition. Overall, our contributions are as follows:

1. We describe the correlation-aware active learning of link specifications.
2. We present the first two generic approaches that implement this concept. The first is based on graph clustering while the second implements the spreading activation principle.
3. We combine these approaches with the EAGLE algorithm [15] and show in ten different settings that our approaches improve EAGLE's performance with respect to both F-score and standard deviation.

The approaches presented herein were included in the LIMES framework[3]. A demo of the approach can be accessed by using the SAIM interface [4]. The rest of this paper is structured as follows: We first present some of the formal notation necessary to understand this work. In addition, we give some insights into why the inclusion of correlation information can potentially improve the behavior of a curious classifier. Thereafter, we present two approaches that implement the paradigm of including correlation information into the computation of the most informative link candidates. We compare the two approaches with the state of

[3] `http://limes.sf.net`
[4] `http://saim.aksw.org`

the art in ten different settings and show that we achieve faster convergence and even a better overall performance in some cases. We finally present some related work and conclude.

2 Preliminaries

In this section, we present the core of the formal notation used throughout this paper. We begin by giving a brief definition of the problem we address. Then, we present the concept of active learning.

2.1 Link Discovery

The formal definition of LD adopted herein is similar to that proposed in [12]. Given a relation R and two sets of instances S and T, the goal of LD is to find the set $M \subseteq S \times T$ of instance pairs (s, t) for which $R(s, t)$ holds. In most cases, finding an explicit way to compute whether $R(s, t)$ holds for a given pair (s, t) is a difficult endeavor. Consequently, most LD frameworks compute an approximation of M by computing a set $\tilde{M} = \{(s, t) : \sigma(s, t) \geq \theta\}$, where σ is a (complex) similarity function and θ is a distance threshold. The computation of an accurate (i.e., of high precision and recall) similarity function σ can be a very complex task [6]. To achieve this goal, machine-learning approaches are often employed. The idea here is to regard the computation of σ and θ as the computation of a classifier $\mathcal{C} : S \times T \to [-1, +1]$. This classifier assigns pairs (s, t) to the class -1 when $\sigma(s, t) < \theta$. All other pairs are assigned the class $+1$. The similarity function σ and the threshold θ are derived from the decision boundary of $\mathcal{C}$.

2.2 Active Learning of Link Specifications

Learning approaches based on genetic programming have been most frequently used to learn link specifications [5,15,17]. Supervised batch learning approaches for learning such classifiers must rely on large amounts of labeled data to achieve a high accuracy. For example, the genetic programming approach used in [7] has been shown to achieve high accuracies when supplied with more than 1000 positive examples. Recent work has addressed this drawback by relying on active learning, which was shown in [15] to reduce the amount of labeled data needed for learning link specifications. The idea behind active learners (also called *curious classifiers* [18]) is to query for the labels of chosen pairs (s, t) (called *link candidates*) iteratively. We denote the count of iterations with t. The function $label : S \times T \to \{\oplus, \ominus, \otimes\}$ stands for the labeling function and encodes whether a pair (s, t) is (1) known be a positive example for a link (in which case $label(s, t) = \oplus$), (2) known to be a negative example (in which case $label(s, t) = \ominus$) or (3) is unclassified (in which case $label(s, t) = \otimes$). We denote classifiers, similarity functions, thresholds and sets at iteration t by using a superscript notation. For example, the classifier at iteration t is denoted

$\mathcal{C}^t$ while $label^t$ stands for the labeling function at iteration t. We call the set $\mathcal{P}^t = \{(s,t) \in S \times T : (label(s,t) = \otimes) \wedge (\mathcal{C}^t(s,t) = +1)\}$ the set *presumed positives*. The set $\mathcal{N}^t$ of *presumed negatives* is defined analogously. If $label(s,t) = \otimes$, then we call the class assigned by $\mathcal{C}$ to (s,t) the *presumed class* of (s,t). When the class of a pair (s,t) is explicit known, we simply use the expression (s,t)*'s class*. The set $\mathcal{C}^{+t} = \{(s,t) : \mathcal{C}^t(s,t) = +1\}$ is called the set of *positive link candidates* while the set $\mathcal{C}^{-t} = \{(s,t) : \mathcal{C}^t(s,t) = -1\}$ is called the set of *negative link candidates*. The query for labeled data is carried out by selecting a subset of $\mathcal{P}^t$ with the magnitude k^+ (resp. a subset of $\mathcal{N}^t$ with the magnitude k^-). In the following, we will assume $k = k^+ = k^-$. The selection of the k elements from $\mathcal{P}^t$ and $\mathcal{N}^t$ is carried out by using a function ifm : $S \times T \to \mathbb{R}$ that can compute how informative a pair (s,t) is for the $\mathcal{C}^t$, i.e., how well the pair would presumably further the accuracy of $\mathcal{C}^t$. We call $\mathcal{I}^{+t} \subseteq \mathcal{P}^t$ (resp. $\mathcal{I}^{-t} \subseteq \mathcal{N}^t$) the set of *most informative positive* (resp. *most informative negative*) link candidates. In this setting, the information content of a pair (s,t) is usually inverse to its distance from the boundary of $\mathcal{C}^t$.

Active learning approaches based on genetic programming adopt a *comittee*-based setting to active learning. Here, the idea is to learn m classifiers $\mathcal{C}_1, \ldots, \mathcal{C}_m$ concurrently and to have the m classifiers select the sets $\mathcal{I}^-$and $\mathcal{I}^+$. This is usually carried out by selecting the k unlabeled pairs (s,t) with positive (resp. negative) presumed class which lead to the highest disagreement amongst the classifiers. Several informativeness functions ifm have been used in literature to measure the disagreement. For example,the authors of [15] use the pairs which maximize

$$\text{ifm}(s,t) = (m - pos(s,t))(m - neg(s,t)), \tag{1}$$

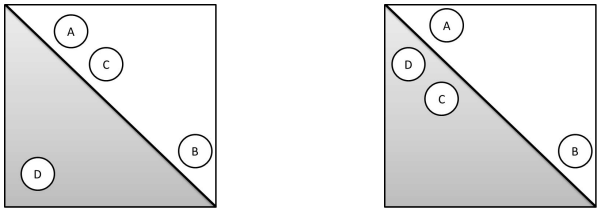
where $pos(s,t)$ stands for the number of classifiers which assign (s,t) the presumed class $+1$, while $neg(s,t)$ stands for the number of classifiers which assign (s,t) the class -1. The authors of [7] on the other hand rely on pairs (s,t) which maximize the entropy score

$$\text{ifm}(s,t) = H\left(\frac{pos(s,t)}{m}\right) \text{ where } H(x) = -x\log(x) - (1-x)\log(1-x). \tag{2}$$

Note that these functions do not take the correlation between the different link candidates into consideration.

3 Correlation-Aware Active Learning of Link Specifications

The basic insight behind this paper is that the correlation between the features of the elements of $\mathcal{N}$ and $\mathcal{P}$ should play a role when computing the sets $\mathcal{I}^+$ and $\mathcal{I}^-$. In particular, two main factors affect the information content of a link candidate: its similarity to elements of its presumed class and to elements of the other class. For the sake of simplicity, we will assume that the presumed class of the link candidate of interest is $+1$. Our insights yet hold symmetrically for link candidates whose presumed class is -1.



(a) Intra-correlation (b) Inter-correlation

Fig. 1. Examples of correlations within classes and between classes. In each subfigure, the gray surface represent $\mathcal{N}$ while the white surface stands for $\mathcal{P}$. The oblique line is $\mathcal{C}$'s boundary.

Let $A = (s_A, t_A), B = (s_B, t_B) \in \mathcal{P}$ to be two link candidates which are equidistant from $\mathcal{C}$'s boundary. Consider Figure 1a, where $\mathcal{P}= \{A, B, C\}$ and $\mathcal{N}= \{D\}$. The link candidate B is on on average most distant from any other elements of $\mathcal{P}$. Thus, it is more likely to be a statistical outlier than A. Hence, making a classification error on B should not have the same impact as an erroneous classification of link candidate A, which is close to another presumably positive link candidate, C. Consequently, B should be considered less informative than A. Approaches that make use of this information are said to exploit the *intra-class correlation*. Now, consider Figure 1b, where $\mathcal{P}= \{A, B\}$ and $\mathcal{N}= \{C, D\}$. While the probability of A being an outlier is the same as B's, A is still to be considered more informative than B as it is located closer to elements of $\mathcal{N}$ and can thus provide more information on where to set the classifier boundary. This information is dubbed *inter-class correlation*.

4 Approaches

Several approaches that make use of these two types of correlations can be envisaged. In the following, we present two approaches for these purposes. The first makes use of intra-class correlations and relies on graph clustering. The second approach relies on the spreading activation principle in combination with weight decay. We assume that the complex similarity function σ underlying $\mathcal{C}$ is computed by combining n atomic similarity functions $\sigma_1, \ldots, \sigma_n$. This combination is most commonly carried out by using metric operators such as min, max or linear combinations.[5] Consequently, each link candidate (s, t) can be described by a vector $(\sigma_1(s, t), \ldots, \sigma_n(s, t)) \in [0, 1]^n$. We define the *similarity of link candidates* $sim : (S \times T)^2 \to [0, 1]$ to be the inverse of the Euclidean distance in the space spawned by the similarities σ_1 to σ_n. Hence, the similarity of two link candidates (s, t) and (s', t') is given by:

[5] See [12] for a more complete description of a grammar for link specifications.

$$sim((s,t),(s',t')) = \frac{1}{1+\sqrt{\sum_{i=1}^{n}(\sigma_i(s,t)-\sigma_i(s',t'))^2}}. \quad (3)$$

Note that we added 1 to the denominator to prevent divisions by 0.

4.1 Graph Clustering

The basic intuition behind using clustering for COALA is that groups of very similar link candidates can be represented by a single link candidate. Consequently, once a representative of a group has been chosen, all other elements of the group become less informative. An example that illustrates this intuition is given in Figure 2. We implemented COALA based on clustering as shown in Algorithm 1. In each iteration, we begin by first selecting two sets $\mathcal{S}^+ \subseteq \mathcal{P}$ resp. $\mathcal{S}^- \subseteq \mathcal{N}$ that contain the positive resp. negative link candidates that are most informative for the classifier at hand. Formally, $\mathcal{S}^+$ fulfills

$$\forall x \in \mathcal{S}^+ \; \forall y \in \mathcal{P}, y \notin \mathcal{S}^+ \rightarrow \text{ifm}(y) \leq \text{ifm}(x). \quad (4)$$

The analogous equation holds for $\mathcal{S}^-$. In the following, we will explain the further steps of the algorithm for $\mathcal{S}^+$. The same steps are carried out for $\mathcal{S}^-$. First, we

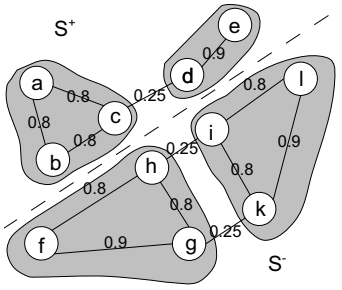


Fig. 2. Example of clustering. One of the most informative single link candidate is selected from each cluster. For example, d is selected from the cluster $\{d, e\}$.

compute the similarity of all elements of $\mathcal{S}^+$ by using the similarity function shown in Equation 3. In the resulting similarity matrix, we set all elements of the diagonal to 0. Then, for each $x \in \mathcal{S}^+$, we only retain a fixed number ec of highest similarity values and set all others to 0. The resulting similarity matrix is regarded as the adjacency matrix of an undirected weighted graph $G = (V, E, sim)$. G's set of nodes V is equal to $\mathcal{S}^+$. The set of edges E is a set of 2-sets[6] of link candidates. Finally, the weighted function is the similarity

[6] A n-set is a set of magnitude n.

function *sim*. Note that *ec* is the minimal degree of nodes in G. In a second step, we use the graph G as input for a graph clustering approach. The resulting clustering is assumed to be a partition $\mathcal{V}$ of the set V of vertices of G. The informativeness of partition $V_i \in \mathcal{V}$ is set to $\max_{x \in V_i} \text{ifm}(x)$. The final step of our approach consists of selecting the most informative node from each of the k most informative partitions. These are merged to generate $\mathcal{I}^+$, which is sent as query to the oracle. The computation of $\mathcal{I}^-$ is carried out analogously. Note that this approach is generic in the sense that it can be combined with any graph clustering algorithm that can process weighted graphs as well as with any informativeness function ifm. Here, we use BorderFlow [16] as clustering algorithm because (1) it has been used successfully in several other applications [9,10] and (2) it is parameter-free and does not require any tuning.

Algorithm 1. COALA based on Clustering

input : mappingSet set of mappings, exampleCount number of examples, edgesPerNode maximal number of edges per node
output: list of mappings for the oracle oracleList

$\mathcal{S}^-$:=get closest negative mapppings(*mappingSet*)
$\mathcal{S}^+$:= get closest positive mapppings(*mappingSet*)
clusterSet:= $\emptyset$
for set $\in \{\mathcal{S}^-, \mathcal{S}^+\}$ **do**
  G := buildGraph(set,edgesPerNode)
  clusterSet $\leftarrow$ *clustering*(G)
  visitedClusters := $\emptyset$, *addedElements* :=0
  sortedMappingList := sortingByDistanceToClassfier(mappingSet)
  repeat
  (s, t):= next(*sortedMappingList*)
  partition:=getPartition((s, t))
  if *partition* $\notin$ *visitedClusters* **then**
    oracleList:=add((s, t))
    addedElements:=+1
    visitedClusters:=addCluster(*partition*)
  until *addedElements* = exampleCount

4.2 Spreading Activation with Weight Decay

The idea behind spreading activation with weight decay (WD) is to combine the intra- and inter-class correlation to determine the informativeness of each link candidate. Here, we begin by computing the set $\mathcal{S} = \mathcal{S}^+ \cup \mathcal{S}^-$, where $\mathcal{S}^+$ and $\mathcal{S}^-$ are described as above. Let $\mathfrak{s}_i$ and $\mathfrak{s}_j$ be the i^{th} and j^{th} elements of $\mathcal{S}$. We then compute the quadratic similarity matrix $\mathcal{M}$ with entries $m_{ij} = sim(\mathfrak{s}_i, \mathfrak{s}_j)$ for $i \neq j$ and 0 else. Note that both negative and positive link candidates belong to $\mathcal{S}$. Thus, $\mathcal{M}$ encodes both inter- and intra-class correlation. In addition to $\mathcal{M}$,

we compute the activation vector $\mathcal{A}$ by setting its entries to $a_i = \text{ifm}(\mathfrak{s}_i)$. In the following, $\mathcal{A}$ is considered to be a column vector. The spreading of the activation with weight decay is then carried out as shown in Algorithm 2.

Algorithm 2. COALA based on Weight Decay

input : mappingSet set of mappings, r fix point exponent, exampleCount number of examples
output: oracleList list of mapping for the oracle
1 $\mathcal{M}$:= buildAdjacenceMatrix(mappingSet)
2 $\mathcal{A}$:= buildActivationVector(mappingSet)
3 **repeat**
4 $\mathcal{A} := \mathcal{A}/max_{\mathcal{A}}$
5 $\mathcal{A} := \mathcal{A} + \mathcal{M} \times \mathcal{A}$
6 $\mathcal{M} := (\forall m_{ij} \in \mathcal{M} : m_{ij} := m_{ij}^{\mathsf{r}})$
7 **until** $\forall m_{ij} \in \mathcal{M} | m_{ij} \neq 1 : m_{ij} \leq \epsilon$
8 oracleList:= getMostActivatedMapping($\mathcal{A}$,exampleCount)

In a first step, we normalize the activation vector $\mathcal{A}$ to ensure that the values contained therein do not grow indefinitely. Then, in a second step, we set $\mathcal{A} = \mathcal{A} + \mathcal{M} \times \mathcal{A}$. This has the effect of propagating the activation of each $\mathfrak{s}$ to all its neighbors according to the weights of the edges between $\mathfrak{s}$ and its neighbors. Note that elements of $\mathcal{S}^+$ that are close to elements of $\mathcal{S}^-$ get a higher activation than elements of $\mathcal{S}^+$ that are further away from $\mathcal{S}^-$ and vice-versa. Moreover, elements at the center of node clusters (i.e., elements that are probably no statistical outliers) also get a higher activation than elements that are probably outliers. The idea behind the weight decay step is to update the matrix by setting each m_{ij} to m_{ij}^r, where $r > 1$ is a fix exponent. This is the third step of the algorithm. Given that $\forall i \forall j\ m_{ij} \leq 1$, the entries in the matrix get smaller with time. By these means, the amount of activation transferred across long paths is reduced. We run this three-step procedure iteratively until all non-1 entries of the matrix are less or equal to a threshold $\epsilon = 10^{-2}$. The k elements of $\mathcal{S}^+$ resp. $\mathcal{S}^-$ with maximal activation are returned as $\mathcal{I}^+$resp. $\mathcal{I}^-$. In the example shown in Figure 3, while all nodes from $\mathcal{S}^+$ and $\mathcal{S}^-$ start with the same activation, two nodes get the highest activation after only 3 iterations.

5 Evaluation

The goal of our evaluation was to study the improvement in F-score achieved by integrating the approaches presented above with a correlation-unaware approach. We chose to use EAGLE [15], an approach based on genetic programming. We ran a preliminary experiment on one dataset to determine good parameter settings for the combination of EAGLE and clustering (CL) as well as the combination EAGLE and weight decay (WD). Thereafter, we compared the F-score achieved by EAGLE with that of CL and WD in ten different settings.

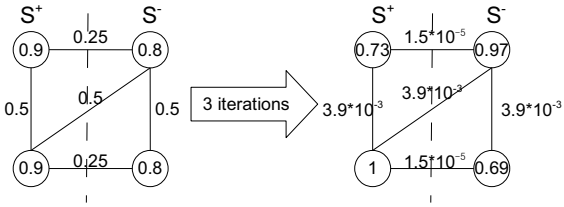


Fig. 3. Example of weight decay. Here r was set to 2. The left picture shows the initial activations and similarity scores while the right picture shows the results after 3 iterations. Note that for the sake of completeness the weights of the edges were not set to 0 when they reached ϵ.

5.1 Experimental Setup

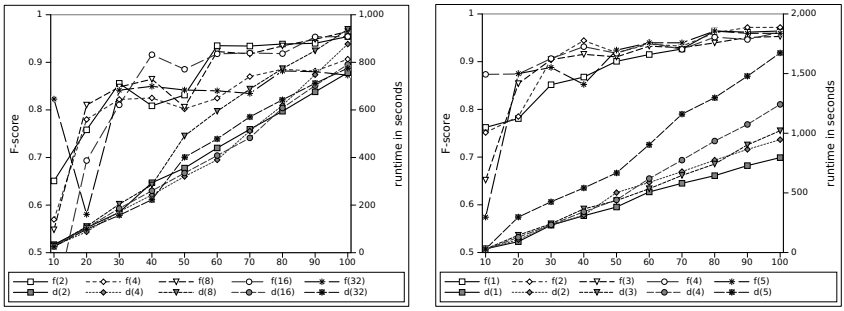
Throughout our experiments, we set both mutation and crossover rates to 0.6. Individuals were given a 70% chance to get selected for reproduction. The population sizes were set to 20 and 100. We set $k = 5$ and ran our experiments for 10 iterations, evolving the populations for 50 generations each iteration. We ran our experiments on two real-world datasets and three synthetic datasets. The synthetic datasets consisted of the datasets from the OAEI 2010 benchmark[7]. The real-world datasets consisted of the ACM-DBLP and Abt-Buy datasets, which were extracted from websites or databases [8] [8]. The ACM-DBLP dataset consists of 2,617 source and 2,295 target publications with 2,224 links between them. The Abt-Buy dataset holds 1,092 links between 1,081 resp. 1,092 products. Note that this particular dataset is both noisy and incomplete. All non-RDF datasets were transformed into RDF and all string properties were set to lower case. Given that genetic programming is non-deterministic, all results presented below are the means of 5 runs. Each experiment was ran on a single thread of a server running JDK1.7 on Ubuntu 10.0.4 and was allocated maximally 2GB of RAM. The processors were 2.0GHz Quadcore AMD Opterons.

5.2 Results

Parametrization of WD and CL. In a preliminary series of experiments we tested for a good parametrization of both WD and CL. For this purpose we ran both approaches on the DBLP-ACM dataset using 5 different values for the r exponent for weight decay and the clustering ec parameter. The tests were ran with a population of 20, $r = \{2, 4, 8, 16, 32\}$ and $ec = \{1, 2, 3, 4, 5\}$. Figures 4a and 4b show the results of achieved F-scores and runtimes. In both plots $f(p)$ and $d(p)$ denote the F-score and runtime of the particular method using the

[7] http://oaei.ontologymatching.org/2010/

[8] http://dbs.uni-leipzig.de/en/research/projects/object_matching/fever/benchmark_datasets_for_entity_resolution



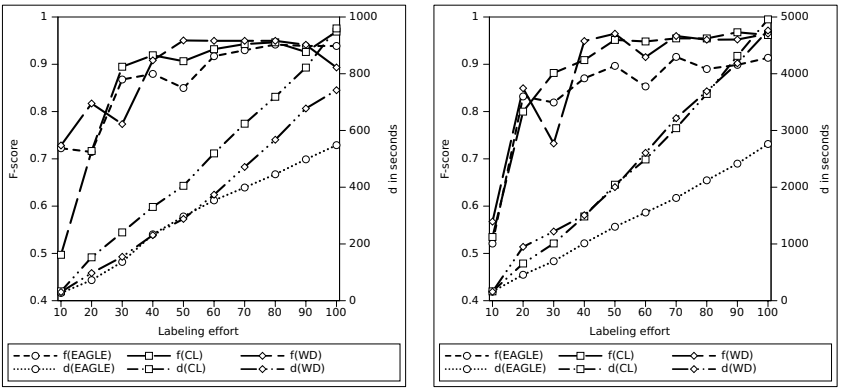
(a) r parameter of WD (b) ec for CL

Fig. 4. Testing different r and ec parameter for both approaches on the DBLP-ACM dataset. $f(p)$ denotes the F-score accieved with the method wusing the parameter p, while $d(p)$ denotes the reuired run time.

p parameter. Figure 4a suggests that $r = 2$ leads to a good accuracy (especially for later inquiries) while requiring moderate computation resources. Similarly, $r = 16$ promises fast convergence and led to better results in the fourth and fifth iterations. Still, we chose $r = 2$ for all experiments due to an overall better performance. The test for different ec parameters led us to use an edge limit of $ec = 3$. This value leads to good results with respect to both accuracy and runtime as Figure 4b suggests.

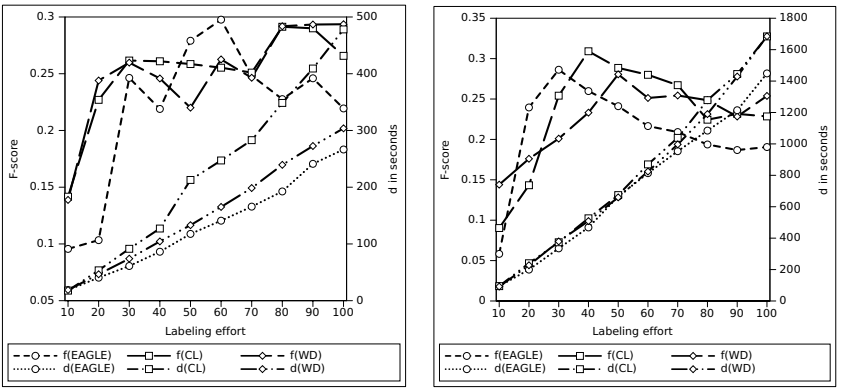
Runtime and F-Score. Figures 5 - 9 show the results of both our approaches in comparison to the EAGLE algorithm. And a summary of the results is given in Table 1. Most importantly, our results suggest that using correlation information can indeed improve the F-score achieved by curious classifiers. The average of the results achieved by the approaches throughout the learning process (left group of results in Table 1) shows that already in average our approaches outperform EAGLE in 9 from 10 settings. A look at the final F-scores achieved by the approaches show that one of the approaches WD and CL always outperform EAGLE both with respect to the average F-score and the standard deviation achieved across the 5 runs except on the Restaurant data set (100 popultion), where the results of CL and EAGLE are the same. This leads us to conclude that the intuition underlying this paper is indeed valid. Interestingly, the experiments presented herein do not allow declaring CL superior to WD or vice-versa. While CL performs better on the small population, WD catches up on larger populations and outperform CL in 3 of 5 settings. An explanation for this behavior could lie in WD taking more information into consideration and thus being more sensible to outliers than CL. A larger population size which reduces the number of outliers would then be better suited to WD. This explanation is yet still to be proven in larger series of experiments and in combination with other

link discovery approaches such as RAVEN. Running WD and CL is clearly more time-demanding than simply running EAGLE. Still the overhead remains within acceptable boundaries. For example, while EAGLE needs approx. 2.9s for 100 individuals on the Abt-Buy dataset while both WD and CL require 3.4s (i.e., 16.3% more time).



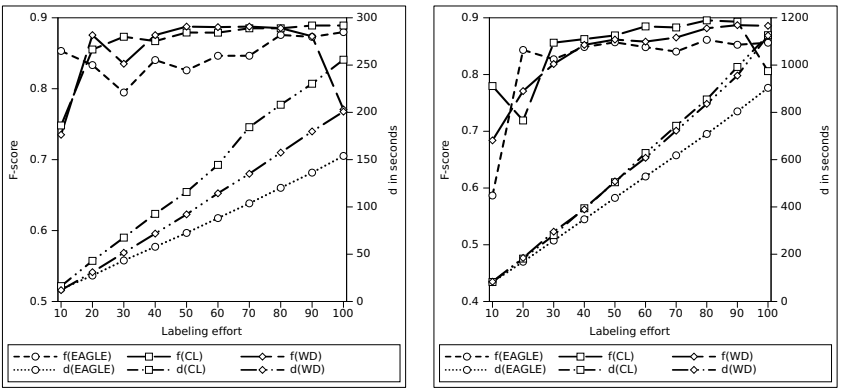
(a) Population = 20 (b) Population = 100

Fig. 5. F-score and runtime on the ACM-DBLP dataset. f(X) stands for the F-score achieved by algorithm X, while d(X) stands for the total duration required by the algorithm.



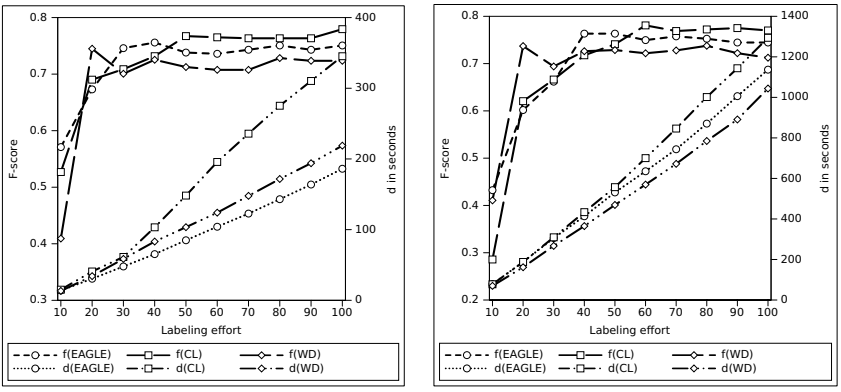
(a) Population = 20 (b) Population = 100

Fig. 6. F-score and runtime on the Abt-Buy dataset



(a) Population = 20 (b) Population = 100

Fig. 7. F-score and runtime on the OAEI 2010 Person1 dataset

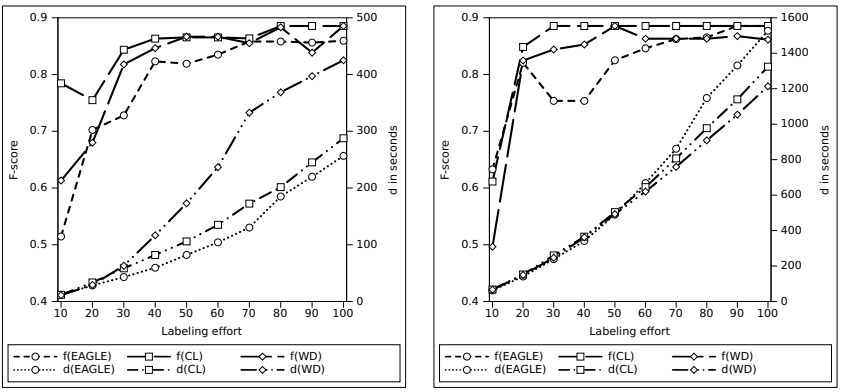


(a) Population = 20 (b) Population = 100

Fig. 8. F-score and runtime on the OAEI 2010 Person2 dataset

6 Related Work

The number of LD approaches has proliferated over the last years. Herein, we present a brief overview of existing approaches (see [11,7] for more extensive presentations of the state of the art). Overall, two main problems have been at the core of the research on LD. First, the time complexity of LD was addressed. In [13], an approach based on the Cauchy-Schwarz inequality was used to reduce the runtime of LD processes based on metrics. The approach $\mathcal{HR}^3$ [11] rely on space tiling in spaces with measures that can be split into independent measures across the dimensions of the problem at hand. Especially, $\mathcal{HR}^3$ was shown to be the first approach that can achieve a relative reduction ratio r' less or equal



(a) Population = 20 (b) Population = 100

Fig. 9. F-score and runtime on the OAEI 2010 Restaurant dataset

Table 1. Comparison of average F-scores achieved by EAGLE, WD and CL. The top section of the table shows the results for a population size of 20 while the bottom part shows the results for 100 individuals. Best scores are in bold font. Abt stands for Abt-Buy, DBLP for DBLP-ACM and Rest. for Restaurants.

	Average values			**Final values**		
DataSet	**EAGLE**	**WD**	**CL**	**EAGLE**	**WD**	**CL**
Abt	0.22± 0.06	**0.25 ± 0.07**	0.25 ± 0.08	0.22± 0.05	**0.29 ± 0.03**	0.27 ± 0.05
DBLP	0.87± 0.1	**0.89± 0.09**	0.87± 0.08	0.94± 0.02	0.89± 0.13	**0.97± 0.0**
Person1	0.85± 0.05	0.85± 0.06	**0.87± 0.03**	0.88± 0.02	0.77± 0.25	**0.89± 0.01**
Person2	0.72± 0.05	0.69± 0.11	**0.73± 0.08**	0.75± 0.02	0.72± 0.09	**0.78± 0.0**
Rest.	0.79± 0.13	0.82± 0.08	**0.85± 0.05**	0.51± 0.36	0.61± 0.28	**0.78± 0.01**
Abt	0.21 ± 0.06	0.23± 0.07	**0.23± 0.05**	0.19 ± 0.04	**0.25± 0.04**	0.23± 0.04
DBLP	0.87± 0.1	0.89± 0.09	**0.89± 0.08**	0.91± 0.03	**0.96± 0.01**	0.96± 0.02
Person1	0.82± 0.05	**0.84± 0.07**	**0.84± 0.07**	0.86± 0.02	**0.89± 0.01**	0.81± 0.18
Person2	**0.7± 0.09**	0.69± 0.1	0.69± 0.07	0.74± 0.03	0.71± 0.08	**0.77± 0.03**
Rest.	0.81± 0.11	0.82± 0.06	**0.85± 0.03**	**0.89± 0.0**	0.86± 0.02	**0.89± 0.0**

to any given relative reduction ratio $r > 1$. Concepts from the deduplication research field were also employed for LD. For example, standard blocking approaches were implemented in the first versions of SILK[9] and later replaced with MultiBlock [6], a lossless multi-dimensional blocking technique. KnoFuss [17] also implements blocking techniques to achieve acceptable runtimes. Moreover, time-efficient string comparison algorithms such as PPJoin+ [21] were integrated into the hybrid framework LIMES [12]. Other LD frameworks can be found in the results of the ontology alignment evaluation initiative [3]. The second problem

[9] `http://wifo5-03.informatik.uni-mannheim.de/bizer/silk/`

that was addressed is the complexity of link specifications. Although unsupervised techniques were newly developed (see, e.g., [17]), most of the approaches developed so far abide by the paradigm of supervised machine learning. For example, the approach presented in [5] relies on large amounts of training data to detect accurate link specification using genetic programming. RAVEN [14] is (to the best of our knowledge) the first active learning technique for LD. The approach was implemented for linear or Boolean classifiers and shown to require a small number of queries to achieve high accuracy. While the first active genetic programming approach was presented in [4], similar approaches for LD were developed later [7,15]. Still, none of the active learning approaches for LD presented in previous work made use of the similarity of unlabeled link candidates to improve the convergence of curious classifiers. Yet, works in other research areas have started considering the combination of active learning with graph algorithms (see e.g., [2]).

7 Conclusion

We presented the first generic LD approaches that make use of the correlation between positive and negative link candidates to achieve a better convergence. The first approach is based on clustering and only makes use of correlations within classes while the second algorithm makes use of both correlations within and between classes. We compared these approaches on 5 datasets and showed that we achieve better F-scores and standard deviations than the EAGLE algorithm. Thus, in future work, we will integrate our approach into other algorithms such as RAVEN. Moreover, we will measure the impact of the graph clustering algorithm utilized in the first approach on the convergence of the classifier. Our experimental results showed that each of the approaches we proposed has its pros and cons. We will thus explore combinations of WD and CL.

References

1. Auer, S., Lehmann, J., Ngonga Ngomo, A.-C.: Introduction to linked data and its lifecycle on the web. In: Polleres, A., d'Amato, C., Arenas, M., Handschuh, S., Kroner, P., Ossowski, S., Patel-Schneider, P. (eds.) Reasoning Web 2011. LNCS, vol. 6848, pp. 1–75. Springer, Heidelberg (2011)
2. Bodó, Z., Minier, Z., Csató, L.: Active learning with clustering. Journal of Machine Learning Research - Proceedings Track 16, 127–139 (2011)
3. Euzenat, J., Ferrara, A., van Hage, W.R., Hollink, L., Meilicke, C., Nikolov, A., Ritze, D., Scharffe, F., Shvaiko, P., Stuckenschmidt, H., Sváb-Zamazal, O., dos Santos, C.T.: Results of the ontology alignment evaluation initiative 2011. In: OM (2011)
4. de Freitas, J., Pappa, G., da Silva, A., Gonçalves, M., Moura, E., Veloso, A., Laender, A., de Carvalho, M.: Active learning genetic programming for record deduplication. In: 2010 IEEE Congress on Evolutionary Computation Evolutionary Computation (CEC), pp. 1–8 (2010)

5. Isele, R., Bizer, C.: Learning linkage rules using genetic programming. In: OM. CEUR Workshop Proceedings, vol. 814 (2011)
6. Isele, R., Jentzsch, A., Bizer, C.: Efficient multidimensional blocking for link discovery without losing recall. In: Marian, A., Vassalos, V. (eds.) WebDB (2011)
7. Isele, R., Jentzsch, A., Bizer, C.: Active learning of expressive linkage rules for the web of data. In: Brambilla, M., Tokuda, T., Tolksdorf, R. (eds.) ICWE 2012. LNCS, vol. 7387, pp. 411–418. Springer, Heidelberg (2012)
8. Köpcke, H., Thor, A., Rahm, E.: Comparative evaluation of entity resolution approaches with fever. Proc. VLDB Endow. 2(2), 1574–1577 (2009)
9. Morsey, M., Lehmann, J., Auer, S., Ngonga Ngomo, A.-C.: DBpedia SPARQL Benchmark – Performance Assessment with Real Queries on Real Data. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 454–469. Springer, Heidelberg (2011)
10. Ngonga Ngomo, A.C.: Parameter-free clustering of protein-protein interaction graphs. In: Proceedings of MLSB Symposium (2010)
11. Ngonga Ngomo, A.-C.: Link Discovery with Guaranteed Reduction Ratio in Affine Spaces with Minkowski Measures. In: Cudré-Mauroux, P., et al. (eds.) ISWC 2012, Part I. LNCS, vol. 7649, pp. 378–393. Springer, Heidelberg (2012)
12. Ngonga Ngomo, A.C.: On link discovery using a hybrid approach. Journal on Data Semantics 1, 203–217 (2012)
13. Ngonga Ngomo, A.C., Auer, S.: LIMES - A Time-Efficient Approach for Large-Scale Link Discovery on the Web of Data. In: Proceedings of IJCAI, pp. 2312–2317 (2011)
14. Ngonga Ngomo, A.C., Lehmann, J., Auer, S., Höffner, K.: RAVEN – Active Learning of Link Specifications. In: Proceedings of OM@ISWC (2011)
15. Ngonga Ngomo, A.-C., Lyko, K.: EAGLE: Efficient active learning of link specifications using genetic programming. In: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 149–163. Springer, Heidelberg (2012)
16. Ngonga Ngomo, A.-C., Schumacher, F.: BorderFlow: A local graph clustering algorithm for natural language processing. In: Gelbukh, A. (ed.) CICLing 2009. LNCS, vol. 5449, pp. 547–558. Springer, Heidelberg (2009)
17. Nikolov, A., d'Aquin, M., Motta, E.: Unsupervised learning of link discovery configuration. In: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 119–133. Springer, Heidelberg (2012)
18. Settles, B.: Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison (2009)
19. Shekarpour, S., Auer, S., Ngonga Ngomo, A.C., Gerber, D., Hellmann, S., Stadler, C.: Keyword-driven sparql query generation leveraging background knowledge. In: International Conference on Web Intelligence (2011)
20. Unger, C., Bühmann, L., Lehmann, J., Ngonga Ngomo, A.-C., Gerber, D., Cimiano, P.: Sparql template-based question answering. In: Proceedings of WWW (2012)
21. Xiao, C., Wang, W., Lin, X., Yu, J.X.: Efficient similarity joins for near duplicate detection. In: WWW, pp. 131–140 (2008)

Transductive Inference for Class-Membership Propagation in Web Ontologies

Pasquale Minervini, Claudia d'Amato, Nicola Fanizzi, and Floriana Esposito

LACAM, Dipartimento di Informatica — Università degli Studi di Bari "Aldo Moro"
via E. Orabona, 4 - 70125 Bari, Italia
firstname.lastname@uniba.it

Abstract. The increasing availability of structured machine-processable knowledge in the context of the Semantic Web, allows for inductive methods to back and complement purely deductive reasoning in tasks where the latter may fall short. This work proposes a new method for similarity-based class-membership prediction in this context. The underlying idea is the *propagation* of class-membership information among similar individuals. The resulting method is essentially non-parametric and it is characterized by interesting complexity properties, that make it a candidate for the application of transductive inference to large-scale contexts. We also show an empirical evaluation of the method with respect to other approaches based on inductive inference in the related literature.

1 Introduction

Standard reasoning services for the Semantic Web (SW) often rely on deductive inference. However, sometimes purely deductive approaches may suffer from limitations owing to the relative complexity of reasoning tasks, the inherent incompleteness of the knowledge bases and the occurrence of logically conflicting (incorrect) pieces of knowledge therein.

Approximate approaches based on both deductive and inductive inference have been proposed as a possible solutions to these limitations. In particular, various methods extend inductive learning techniques to tackle SW representations that are ultimately based on Description Logics (DL): they perform some sort of approximate reasoning efficiently by predicting assertions which were not derivable (or refutable) from the knowledge base and even coping with potential cases of inconsistency, since they are essentially data-driven (see [14], for a recent survey). Approximate data-driven forms of class-membership prediction could be useful for addressing cases such as the one illustrated in Ex. 1:

Example 1 (Academic Citation Network). Let us consider a knowledge base representing a *Bibliographic Citation Network* where papers, venues and authors are linked by relations such as writtenBy, publishedIn and citedBy. Assuming that specializations of paper based on the topics are also given, e.g. by means of disjoint classes such as MachineLearningPaper and DatabasePaper, one may want to ascertain the membership of an instance (a new paper) to either class. Owing to the Open-world assumption which is typically made when reasoning with SW representations, this task may not lead to a definite (positive or negative) conclusion in absence of explicit assertions.

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 457–471, 2013.

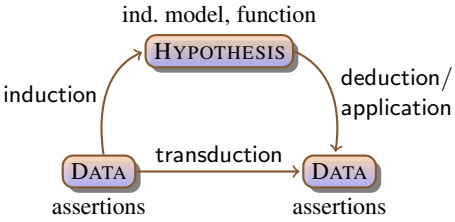


Fig. 1. Transductive and inductive inference

Bridging the gap caused by missing data can be cast as a *statistical learning* problem [18] for which efficient solutions may be found by adapting techniques proposed in the related literature. In principle, they may serve for completions even for large and Web-scale knowledge bases.

A variety of approaches to the class-membership prediction problem have been proposed in the literature. Among the various approaches, *discriminative* methods proposed so far tend to ignore unlabeled instances (individuals for which the target value of such class-membership is unknown); however, accounting for unlabeled instances during learning can provide more accurate results if some conditions are met [3]. *Generative* methods, on the other hand, try to model a joint probability distribution on both instances and labels, thus facing a possibly harder learning problem than only predicting the most probable membership for any given instance.

Several approaches to the class-membership prediction problem belong to the former category. They are often based on a notion of similarity, such as the k-Nearest Neighbors (k-NN) algorithm applied to DL knowledge bases [4]. A variety of similarity (and dissimilarity) measures between either individuals or concepts have been proposed [5]: some are based on *features* and objects are described in terms of a set of them (e.g. see [9]), some on a *semantic-network* structure that provides a form of background information (e.g. see [10]), while some rely on the *information content* (where both the semantic network structure and population are considered). Kernel-based algorithms have been proposed for various learning tasks from DL-based representations. This is made possible by the existence of a variety of kernel functions, either for concepts or individuals (e.g. see [6, 2, 14]). By (implicitly) projecting instances into a high-dimensional feature space, kernel functions allow to adapt a multitude of machine learning algorithms to structured representations. SW literature also includes methods for inducing classifiers from DL knowledge bases using some sort of RBF networks [7].

Also, methods based on a generative approach to learning have been proposed. In [15], each individual is associated to a *latent variable* which influences its attributes and the relations it participates in. A quite different approach is discussed in [13], which focuses on learning theories in a probabilistic extension of the $\mathcal{ALC}$ DL named CR$\mathcal{ALC}$. Extending our previous work [12], we propose a novel *transductive inference* method to be applied to class-membership prediction problem with knowledge bases expressed in standard SW representations. The nature of transductive inference, as opposed to induction, is illustrated in Fig. 1. Induction essentially generalizes existing data constructing an intermediate hypothesis (e.g. a classification function) that allows for making predictions on arbitrary individuals by deduction from the hypothesis (i.e. applying the

induced classifier); transduction aims at propagating information on class-membership from the individuals for which membership is explicitly known towards those for which this information is missing (i.e. predicting new assertions), exploiting some notion of similarity among individuals (with *smooth variations*). Note that no generalization is made in this case.

Example 2 (Academic Citation Network, cont'd). It may be quite expensive to inductively build an inductive classifier that, given an arbitrary previously unseen paper, outputs the class of papers representing its specific topic. If one assumes that the citedBy relation can be associated to an indicator that two papers are likely to deal with the same topics or, similarly, that the same is likely to hold for papers written by the same author, transductive inference may be exploited to find a topic (i.e. a class-membership) assignment which varies smoothly among similar papers, and is consistent with the membership of examples provided by some domain expert.

In this work, we propose a method for spreading class-membership information among individuals for which this information is neither explicitly available nor derivable through deductive reasoning. This is accomplished by first constructing a *semantic similarity graph* encoding similarity relations among individuals. Then, class-membership information is propagated by minimizing a cost function grounded on a graph-based regularization approach. The remainder of the paper is organized as follows. In Sect. 2, transductive inference and the corresponding variant to the classic class-membership prediction problem are defined. In Sect. 3 we describe the proposed method, the assumptions it relies on, and how it can be used for class-membership prediction also on larger knowledge bases. In Sect. 4, we provide empirical evidence for the effectiveness of the proposed transductive class-membership propagation method in comparison with other methods in literature. In Sect. 5 we provide a brief summary of this work and about further developments of the proposed method.

2 Preliminaries

In the following, instances are described by features ranging in a certain space X and their classification with respect a given concept is indicated by labels in Y. In a probabilistic setting, instances are assumed to be sampled i.i.d. from an unknown joint probability distribution P ranging over $X \times Y$; *generative* methods are characterized by building an estimate $\hat{P}$ of $P(X, Y)$ from a given sample of instances, that is used to infer $\hat{P}(Y \mid x) = \hat{P}(Y, x)/\hat{P}(x)$ for some instance $x \in X$ whose unknown label is to be predicted. On the other hand, *discriminative* methods focus on conditional distributions to identify $\arg\max_y P(y \mid x)$, for any given $(x, y) \in X \times Y$, that is an easier problem than estimating the joint probability distribution.

2.1 Semi-supervised Learning and Transductive Inference

Classic learning methods tend to ignore unlabeled instances. However, real-life scenarios are usually characterized by an abundance of unlabeled instances and a few labeled ones. This is also the case of class-membership prediction problem from formal ontologies: explicit class-membership assertions may be difficult to obtain during ontology

engineering tasks (e.g. due to availability of domain experts) and inference (e.g. since deciding instance-membership may have an intractable time complexity with knowledge bases described by expressive Web-ontology languages).

Making use of unlabeled instances during learning is commonly referred to in literature as *Semi-Supervised Learning* [3] (SSL). A variant of this setting known as *Transductive Learning* [18] refers to finding a labeling only to unlabeled instances provided in the training phase, without necessarily generalizing to further unseen instances, resulting in a possibly *simpler* learning problem [18]. If the marginal distribution of instances P_X is informative w.r.t. the conditional distribution $P(Y \mid x)$, accounting for unlabeled instances during learning can provide more accurate results [3]. A possible approach is including terms dependent on P_X into the objective function.

The method proposed in this work relies on the so-called *semi-supervised smoothness assumption* [3]: *if two instances $x_i, x_j \in X$ in a high-density region are close then so should be the corresponding labels $y_i, y_j \in Y$*. Learning smooth labeling functions, this can be exploited by transitivity along paths of high density.

We will face a slightly different version of the classic class-membership prediction problem, namely *transductive class-membership prediction*. It is inspired by the *Main Principle* [18]: "If you possess a restricted amount of information for solving some problem, try to solve the problem directly and never solve a more general problem as an intermediate step. It is possible that the available information is sufficient for a direct solution but is insufficient for solving a more general intermediate problem". In this setting, the learning algorithm only aims at estimating the class-membership relation of interest for a given training set of individuals, without necessarily being able to generalize to instances outside this sample. In this work, transduction and induction differ in the target of the regularization: the latter would target the hypothesis (i.e. the inductive model), while the former targets directly the results of predictions.

2.2 Transductive Class-Membership Learning Problem in DL

Transductive class-membership learning with DL knowledge bases can be formalized as a cost minimization problem: given a set of training individuals $\mathrm{Ind}_C(\mathcal{K})$ whose class-membership w.r.t. a target concept C is either known or unknown, find a function $f^* : \mathrm{Ind}_C(\mathcal{K}) \to \{+1, -1\}$ defined over training individuals and returning a value $+1$ (resp. -1) if the individual likely to be a member of C (resp. $\neg C$), minimizing a given cost function. More formally:

Definition 1 (Transductive Class-Membership Learning).

- **Given:**
  - *a* target *concept C in a knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$;*
  - *a set of training individuals $\mathrm{Ind}_C(\mathcal{K}) \subseteq \mathrm{Ind}(\mathcal{A})$ in $\mathcal{K}$ partitioned, according to their membership w.r.t. C, into the following sets:*
    - $\mathrm{Ind}_C^+(\mathcal{K}) = \{a \in \mathrm{Ind}_C(\mathcal{K}) \mid \mathcal{K} \models C(a)\}$ *positive examples,*
    - $\mathrm{Ind}_C^-(\mathcal{K}) = \{a \in \mathrm{Ind}_C(\mathcal{K}) \mid \mathcal{K} \models \neg C(a)\}$ *negative examples,*
    - $\mathrm{Ind}_C^0(\mathcal{K}) = \{a \in \mathrm{Ind}_C(\mathcal{K}) \mid \mathcal{K} \not\models C(a) \wedge \mathcal{K} \not\models \neg C(a)\}$ *unlabeled examples (i.e. whose concept-membership relation w.r.t. C is unknown);*

- *A* cost function $cost(\cdot) : \mathcal{F} \mapsto \mathbb{R}$, *specifying the* cost *associated to labeling functions* $f \in \mathcal{F}$ *of the form* $\mathrm{Ind}_C(\mathcal{K}) \mapsto \{+1, -1\}$;

– **Find** $f^* \in \mathcal{F}$ *minimizing* $cost(\cdot)$ *w.r.t. the training individuals in* $\mathrm{Ind}_C(\mathcal{K})$*:*

$$f^* \leftarrow \arg\min_{f \in \mathcal{F}} cost(f).$$

The function f^* determined by a proper transductive class-membership learning method can then be used to predict class-membership relations w.r.t. the target concept C for all training individuals (including those in $\mathrm{Ind}^0_C(\mathcal{K})$): it will return $+1$ (resp. -1) if an individual is likely to be a member of C (resp. $\neg C$). Note that the function is defined on the whole set of training individuals but it is not a generalization stemming from them; therefore, possibly, it may contradict class-membership assertions that are already available (thus being able to handle noisy knowledge). Since $\mathrm{Ind}_C(\mathcal{K})$ is finite, the space of labeling functions $\mathcal{F}$ is also finite, and each function $f \in \mathcal{F}$ can be equivalently expressed as a vector in $\{-1, +1\}^n$, where $n = |\mathrm{Ind}_C(\mathcal{K})|$.

In order to solve this problem, we propose a similarity-based, non-parametric and computationally efficient method for predicting missing class-membership relations. This method is essentially discriminative, and may account for unknown class-membership relations during learning.

3 Propagating Class-Membership Information among Individuals

A transductive method based on *graph-regularization*[1] [3] is presented allowing for class-membership prediction with knowledge bases expressed in DL. The method relies on a weighted *semantic similarity graph*, where nodes represent positive, negative and unlabeled examples of the transductive class-membership prediction problem, and weighted edges define similarity relations among such individuals.

Given an instance of the *transductive class-membership learning problem* (see Def. 1), the approach proposed in this work is outlined in Alg. 1 and summarized by the following basic steps:

1. Given a class-membership prediction task and a set of training individuals (either labeled and unlabeled), create an undirected *semantic similarity graph* (SSG) where two individuals are linked iff they are considered *similar* (that is, their class-membership is not likely to change from one individual to another).
2. Propagate class-membership information among similar individuals (transduction step), by minimizing a cost function based on a graph regularization approach (where the graph is given by the SSG) and defined over possible class-membership relations for training individuals.

This method can be seen as inducing a new metric, in which neighborhood relations among training individuals are preserved; and then, performing classic supervised learning using the new distance.

[1] In brief, *regularization* consists in introducing additional terms to an objective function to be optimized to prevent overfitting. These terms add usually some penalty for complexity and have the form of restrictions for smoothness, bounds on the vector space norm or number of model parameters.

Algorithm 1. Transductive Class-Membership Prediction via Graph-Based Regularization with the Semantic Similarity Graph

Input: Initial class-membership relations $\mathrm{Ind}_C^+(\mathcal{K})$, $\mathrm{Ind}_C^-(\mathcal{K})$ and $\mathrm{Ind}_C^0(\mathcal{K})$ w.r.t. a concept C and a knowledge base $\mathcal{K}$;
Output: $f^* \in \mathcal{F}$
{Compute the Semantic Similarity Graph (SSG) G, encoding neighborhood relations among individuals in $\mathrm{Ind}_C(\mathcal{K})$.}
$G \leftarrow semanticSimilarityGraph(\mathrm{Ind}_C(\mathcal{K}))$;
{Minimize a cost function *cost* defined over a set of labeling functions $\mathcal{F}$. The cost function is based on the SSG G and enforces smoothness in class-membership relations among similar individuals as well as consistency with initial class-membership relations.}
$f^* \leftarrow \arg\min_{f \in \mathcal{F}} cost(f, G, Ind_C(\mathcal{K}))$;
return f^*;

Example 3 (Academic Citation Network (cont.d)). Assuming that papers written by the same authors or cited by the same articles (where such information is encoded by the *writtenBy* and *citedBy* roles respectively) have a tendency to have similar domain-memberships, we can construct a SSG in which each paper is linked to its k most similar papers, and rely on this structure to propagate domain-membership information.

In the following, the procedure for building a SSG among individuals in the training set $\mathrm{Ind}_C(\mathcal{K})$ is illustrated. As regards the labeling process of unlabeled training examples, namely the transductive step, a optimal labeling function f^* has to be found by minimizing a given cost function. For defining a cost over the space of the labeling functions $f \in \mathcal{F}$, the proposed method (see Sect. 3.2) aims at finding a labeling function that is both consistent with the given labels, and changes smoothly between similar instances (where similarity relations are encoded in the SSG). This is formalized through a *regularization by graph framework*, using the loss function as a measure of consistency to the given labels, and a measure of smoothness among the similarity graph as a regularizer.

3.1 Semantic Similarity Graph

A similarity graph for a set of training examples is a graph where the set of nodes is given by the training examples and edges between nodes connect similar training examples with respect to a given similarity measure. Edges are labeled with the corresponding computed similarity values.

A similarity graph can be modeled as a weighted adjacency matrix $\mathbf{W}$ (or, briefly, weight matrix), where $\mathbf{W}_{ij}$ represents the similarity value of x_i and x_j. Specifically, $\mathbf{W}$ is often obtained as a k-Nearest Neighbor (NN) graph [3] where each instance is connected to the k most similar instances in the graph, or to those with a similarity value above a given threshold ϵ, while the remaining similarity values are set to 0.

For building such a similarity graph given the individuals in $\mathrm{Ind}_C(\mathcal{K})$, a solution is relying on the family of dissimilarity[2] measures defined in [14], since they do not

[2] A dissimilarity measure $d \in [0, 1]$ can be transformed in a similarity measure $s = 1 - d$ [5].

constrain to any particular family of DLs. Since this measure is a *semantic similarity measures*, following the formalization in [5], we call the resulting similarity graph as the *semantic similarity graph* (SSG).

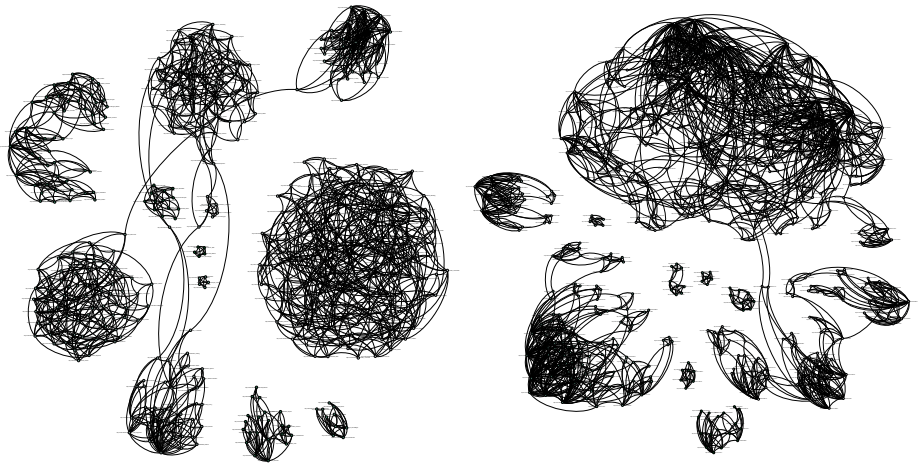
The adopted dissimilarity measure is briefly recalled in the following. Given a set of concept descriptions $F = \{F_1, \ldots, F_n\}$ in $\mathcal{K}$ and a weight vector $\mathbf{w} = (w_1, \ldots, w_n)$, the family of dissimilarity measures $d_p^F : Ind(\mathcal{K}) \times Ind(\mathcal{K}) \mapsto [0, 1]$ is defined as:

$$d_p^F(x_i, x_j) = \left[\sum_{k=1}^{|F|} w_k |\delta_k(x_i, x_j)|^p \right]^{\frac{1}{p}} \tag{1}$$

where $p > 0$, $Ind(\mathcal{K})$ is the set of all individuals in the knowledge base $\mathcal{K}$, $x_i, x_j \in Ind(\mathcal{K})$ and $\forall k \in \{1, \ldots, n\}$ results:

$$\delta_k(x_i, x_j) = \begin{cases} 0 \text{ if } (\mathcal{K} \models F_i(x) \wedge \mathcal{K} \models F_i(y)) \vee (\mathcal{K} \models \neg F_i(x) \wedge \mathcal{K} \models \neg F_i(y)) \\ 1 \text{ if } (\mathcal{K} \models F_i(x) \wedge \mathcal{K} \models \neg F_i(y)) \vee (\mathcal{K} \models \neg F_i(x) \wedge \mathcal{K} \models F_i(y)) \\ u_k \text{ otherwise} \end{cases}$$

where u_k can reflect the degree of uncertainty on the membership w.r.t the k-th feature in the concept committee [14]. We proposed such a measure in our previous work [12] for building the SSG among a set of individuals in a knowledge base. Such a dissimilarity measure can be used to obtain a kernel function among individuals by simply turning it into a similarity measure [14].



(a) AIFB Affiliations (Persons)

(b) AIFB Affiliations (Articles)

Fig. 2. Semantic Similarity Graphs for individuals representing persons and articles in the AIFB Affiliations ontology (5-NN graphs obtained using the Full SubTree kernel [11] with parameters $d = 1$ and $\lambda = 0.9$)

An alternative approach for obtaining the SSG among a set of individuals in a knowledge base, by relying more on the corresponding network structure, is by means of *graph* and RDF kernels: a kernel provides an (implicitly) mapping for individuals into an embedding space, by calculating their inner product. A recently proposed kernel for RDF data is the Full SubTree (FST) kernel [11].

Let $k : Ind(\mathcal{K}) \times Ind(\mathcal{K}) \rightarrow \mathbb{R}$ be a kernel function defined over individuals in a knowledge base $\mathcal{K}$. Since k corresponds to an embedding function ϕ mapping individuals to points in an embedding space, that is $\forall x_i, x_j \in Ind(\mathcal{K}) : k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$, it is immediate to derive the Euclidean distance in the embedding space among two individuals [16]: $||\phi(x_i) - \phi(x_j)|| = \sqrt{k(x_i, x_i) - 2k(x_i, x_j) + k(x_j, x_j)}$.

Two examples of k-NN SSGs among individuals in the AIFB Affiliations ontology (representing instances of the concepts Person and Article), which is also used in empirical evaluations in Sect.4, are shown in Fig. 2. In both cases, a clustered structure emerges from the graphs. In the case of the SSG modeling the similarity relations among instances of the Person concept, an highly connected subgraph groups persons working in the EOrg research group; another connected component (composed by two highly connected subgraphs) groups persons in the BIK research group; two connected components group persons affiliated to the CoM research group; and three single connected components group respectively persons with no available affiliation (the larger component) and affiliated to the WBS and EffAlg research groups. Also instances of the Article concept tend to be grouped into different components of their SSG. Similarly to the previous example, articles tend to be grouped according to their research group affiliation, such as CoM or EffAlg. However, some articles affiliated to different research group share one or more authors, causing the presence of a few connections among the different clusters.

In this work, we propose to leverage such emerging structures in class-membership prediction tasks. The underlying idea is to *propagate* class-membership information among similar individuals, assuming that such information tends not to vary within regions of the instance space with an high density of instances (due to the semi-supervised smoothness assumption discussed in Sect. 2).

3.2 Transductive Inference via Quadratic Cost Criteria

In this section the transductive step is illustrated. It basically consists in labeling the unlabeled training examples. For doing this, a optimal labeling function f^* has to be found by minimizing a given cost function (see Def. 1). For determining a cost over the space of the labeling functions $f \in \mathcal{F}$, the method finds a function that is: 1) consistent with the given labels; 2) changes smoothly between similar instances (encoded in the semantic similarity graph). The first issue is addressed by adopting the loss function as a measure of consistency with respect to the given labels. The second issue is addressed by regularizing the labeling of the function with respect to the structure of the semantic similarity graph.

For addressing the consistency issue, the quadratic cost criteria [3, ch. 11] are considered where the adopted label space $\{-1, +1\}$ is the one for the binary classification case. We relax this label space to the interval $[-1, +1]$ that allows to express the confidence associated to a labeling. Consequently, also the labeling functions space $\mathcal{F}$ is

relaxed to functions of the form $f : \mathrm{Ind}_C(\mathcal{K}) \mapsto [-1, +1]$. Labeling functions can be equivalently represented as vectors $\mathbf{y} \in [-1, +1]^n$ where n is the number of the training examples. Let $\hat{\mathbf{y}} \in [-1, +1]^n$ be a possible labeling for n instances. $\hat{\mathbf{y}}$ can be seen as a $(l + u) = n$ dimensional vector, where the first l indices refer to already labeled instances, and the last u to unlabeled instances: $\hat{\mathbf{y}} = [\hat{\mathbf{y}}_l, \hat{\mathbf{y}}_u]$. The consistency of $\hat{\mathbf{y}}$ with respect to the original labels is then formulated in the form of a quadratic cost: $\sum_{i=1}^{l}(\hat{y_i} - y_i)^2 = ||\hat{\mathbf{y}}_l - \mathbf{y}_l||^2$.

To regularize the labellings with respect to the graph structure, the *graph Laplacian* [3] can be exploited. Let $\mathbf{W}$ be the weight matrix corresponding to the similarity graph G, and let $\mathbf{D}$ be the diagonal matrix obtained from $\mathbf{W}$ as $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$ namely by summing the elements in each column of $\mathbf{W}$. Hence, two alternative definitions for the graph Laplacian can be considered [3]:

- Unnormalized graph Laplacian: $\mathbf{L} = \mathbf{D} - \mathbf{W}$;
- Normalized graph Laplacian: $\mathcal{L} = \mathbf{D}^{-0.5}\mathbf{L}\mathbf{D}^{-0.5} = \mathbf{I} - \mathbf{D}^{-0.5}\mathbf{W}\mathbf{D}^{-0.5}$.

Following [1], a possible graph-based regularization factor is $0.5\sum_{i,j=1}^{n}\mathbf{W}_{ij}(\hat{y_i} - \hat{y_j})^2 = \hat{\mathbf{y}}^T\mathbf{L}\hat{\mathbf{y}}$; in alternative it is possible to resort to the normalized graph Laplacian [19, 20], using the slightly different regularization factor $\hat{\mathbf{y}}^T\mathcal{L}\hat{\mathbf{y}}$.

For preventing overfitting, an additional regularization term, in the form of $||\hat{\mathbf{y}}||^2$ (or $||\hat{\mathbf{y}}_u||^2$, as in [19]), can be added. This additional low norm regularizer on $\hat{\mathbf{y}}$ helps avoiding overfitting and preventing arbitrary labellings in connected components of the semantic similarity graphs containing only unlabeled instances.

Putting the pieces together, two quadratic cost criteria in the form proposed in the literature are obtained, namely Regularization on Graph [1] (RG) and Consistency Method [19] (CM):

- **RG:** $cost(\hat{\mathbf{y}}) = ||\hat{\mathbf{y}}_l - \mathbf{y}_l||^2 + \mu\hat{\mathbf{y}}^T\mathbf{L}\hat{\mathbf{y}} + \mu\epsilon||\hat{\mathbf{y}}||^2$;
- **CM:** $cost(\hat{\mathbf{y}}) = ||\hat{\mathbf{y}}_l - \mathbf{y}_l||^2 + \mu\hat{\mathbf{y}}^T\mathcal{L}\hat{\mathbf{y}} + ||\hat{\mathbf{y}}_u||^2$.

Once the form of the cost function is determined, the minimum for the function has to be found. As a title of example, a closed form solution for the problem of finding a (global) minimum for the quadratic cost criterion in RG is showed.

Let $\mathbf{S}$ be the diagonal matrix $\mathbf{S} = diag(\mathbf{s}_1, \ldots, \mathbf{s}_n)$ obtained by setting $\mathbf{s}_i = 1$ iff $i \leq l$ and 0 otherwise. The first order derivative for the case of the cost function in RG can be written as:

$$\frac{1}{2}\frac{\partial cost(\hat{\mathbf{y}})}{\partial\hat{\mathbf{y}}} = (\mathbf{S} + \mu\mathbf{L} + \mu\epsilon\mathbf{I})\hat{\mathbf{y}} - \mathbf{S}\mathbf{y}.$$

The second order derivative is a positive definite matrix if $\epsilon > 0$, since $\mathbf{L}$ is positive semi-definite. Hence, setting the first order derivative to 0 leads to a global minimum:

$$\hat{\mathbf{y}} = (\mathbf{S} + \mu\mathbf{L} + \mu\epsilon\mathbf{I})^{-1}\mathbf{S}\mathbf{y},$$

showing that $\hat{\mathbf{y}}$ can be obtained either by matrix inversion or by solving a (possibly sparse) linear system.

In this way, this work leverages quadratic cost criteria to efficiently solve the transductive class-membership prediction problem. An advantage of quadratic cost criteria is

that their minimization ultimately reduces to solving a large sparse linear system [19, 3], a well-known problem in the literature whose time complexity is nearly linear in the number of non-zero entries in the coefficient matrix [17]. For large-scale datasets, a subset selection method is described in [3, ch. 18], which allows to greatly reduce the size of the original linear system.

4 Empirical Evaluation

In this section, we evaluate several (inductive and transductive) methods for class-membership prediction, with the aim of comparing the methods discussed in Sect. 3 with respect to other methods in the SW literature.

Specifically, we empirically compared a set of different methods for the class-membership prediction task. Those can be partitioned in transductive (Regularization on Graph [1] (RG), Consistency Method [19] (CM) and Label Propagation [21] (LP)) and inductive (Soft-Margin Support Vector Machines with L_1 norm (SM-SVM) and $\sqrt{l}$-Nearest Neighbors). Such inductive approaches have also been discussed in the task of class-membership prediction in [14], and previously in the context of inducing robust classifiers from ontological knowledge bases [8]. Implementations for the evaluated methods, as well as the dataset used in this work, are available online [3].

4.1 Evaluated Methods

LP is a graph-based transductive inference algorithm relying on the idea of propagating labeling information among similar instances through an iterative process involving matrix operations. It can be equivalently formulated under the quadratic criterion framework [3, ch. 11]. More formally it associates, to each unlabeled instance in the graph, the probability of performing a random walk until a positively (resp. negatively) example is found. Support Vector Machine classifiers, on the other hand, come in different flavors: the classic (Hard-Margin) SVM binary classifier aims at finding the hyperplane in the feature space separating the instances belonging to different classes, which maximizes the *geometric margin* between the hyperplane and nearest training points. The SM-SVM relaxes this method, by allowing for some misclassification in training instances (by relaxing the need of having perfectly linearly separable training instances in the feature space). We adopted this latter solution to handle the lack of perfect linear separability of the instances belonging to different classes. Note that the aforementioned methods can be seen as relying on a *change of representation*: instances of the prediction problem are represented as points in an embedding space, and implicitly described by means of their pairwise Euclidean distances, inner products (as in the case of kernel-based methods, such as SVM) or neighborhood relations. We evaluated different choices for such change of representation, consisting in different choices for the (dis-)similarity measure used to construct the k-Nearest Neighborhood graph, and the kernel function. Specifically, we evaluated the following choices:

[3] At the address `http://lacam.di.uniba.it/~nico/research/ontologymining.html`

Table 1. Results for a 10-fold cross validation obtained when predicting the affiliations of AIFB staff members to research groups, using the **Atomics** kernel (and the corresponding dissimilarity measure)

EffAlg	Match	Omission	Commission	F1
LP+Atomics	0.53 ± 0.189	0 ± 0	0.47 ± 0.189	0.488 ± 0.217
RG+Atomics	0.458 ± 0.166	0.01 ± 0.032	0.532 ± 0.158	0.405 ± 0.194
SM-SVM+Atomics	0.6 ± 0.125	0 ± 0	0.4 ± 0.125	0.555 ± 0.198
$\sqrt{l}$-NN+Atomics	0.5 ± 0	0 ± 0	0.5 ± 0	0.667 ± 0
CoM	Match	Omission	Commission	F1
LP+Atomics	0.533 ± 0.317	0 ± 0	0.467 ± 0.317	0.419 ± 0.39
RG+Atomics	0.475 ± 0.294	0 ± 0	0.525 ± 0.294	0.36 ± 0.333
SM-SVM+Atomics	0.517 ± 0.207	0 ± 0	0.483 ± 0.207	0.403 ± 0.31
$\sqrt{l}$-NN+Atomics	0.5 ± 0.167	0 ± 0	0.5 ± 0.167	0.517 ± 0.277
BIK	Match	Omission	Commission	F1
LP+Atomics	0.502 ± 0.116	0.037 ± 0.064	0.46 ± 0.117	0.451 ± 0.176
RG+Atomics	0.531 ± 0.089	0.005 ± 0.014	0.464 ± 0.083	0.488 ± 0.147
SM-SVM+Atomics	0.514 ± 0.068	0 ± 0	0.486 ± 0.068	0.337 ± 0.214
$\sqrt{l}$-NN+Atomics	0.522 ± 0.072	0 ± 0	0.478 ± 0.072	0.404 ± 0.125
EOrg	Match	Omission	Commission	F1
LP+Atomics	0.667 ± 0.167	0 ± 0	0.333 ± 0.167	0.65 ± 0.146
RG+Atomics	0.692 ± 0.157	0 ± 0	0.308 ± 0.157	0.667 ± 0.136
SM-SVM+Atomics	0.692 ± 0.197	0 ± 0	0.308 ± 0.197	0.647 ± 0.286
$\sqrt{l}$-NN+Atomics	0.717 ± 0.185	0 ± 0	0.283 ± 0.185	0.713 ± 0.174
WBS	Match	Omission	Commission	F1
LP+Atomics	0.504 ± 0.069	0.012 ± 0.028	0.484 ± 0.072	0.489 ± 0.081
RG+Atomics	0.512 ± 0.09	0 ± 0	0.488 ± 0.09	0.512 ± 0.101
SM-SVM+Atomics	0.603 ± 0.084	0 ± 0	0.397 ± 0.084	0.503 ± 0.131
$\sqrt{l}$-NN+Atomics	0.513 ± 0.097	0 ± 0	0.487 ± 0.097	0.522 ± 0.152

Atomics – a dissimilarity measure defined in [14] (outlined in Eq. 1) was used to construct the k-Nearest Neighborhood graph (with $p = 2$, using all atomic concepts in the ontology as features and weighting each concept with its associated entropy [14]). The corresponding kernel function was obtained as discussed in Sect. 3.

Full SubTree kernel (FST) – a kernel for RDF data proposed in [11]; it was used to construct a k-NN SSG as shown in Sect. 3. The optimal kernel parameters $(depth, \lambda)$ were found within the training set using a k-fold cross validation procedure (with $k = 10$), and varied in $\{1, 2\}$ and $\{0.1, 0.5, 0.9\}$ respectively.

4.2 Evaluation Procedure

Extending our previous results in [12], we are evaluating the proposed approach on a knowledge base in which a quantity of information is stored in the network structure rather than in the concept hierarchy. The empirical evaluation involved the metadata available in the Semantic Portal of the institute AIFB [4]. The ontology models key concepts within a research community: it comprises 44351 individuals and the Person, Document and Project FOAF concepts (among others) are associated to respectively 509, 4731 and 128 individuals, and roles include affiliation relationships between persons and research groups, authorship relations between persons and documents, and

[4] `http://www.aifb.kit.edu/web/Wissensmanagement/Portal`, as of 21 Feb. 2012

Table 2. Results for a 10-fold cross validation obtained when predicting the affiliations of AIFB staff members to research groups, using the **Full SubTree** kernel (and corresponding dissimilarity measure)

EffAlg	Match	Omission	Commission	F1
LP+FST	0.565 ± 0.167	0.09 ± 0.099	0.345 ± 0.201	0.611 ± 0.218
RG+FST	0.548 ± 0.154	0.08 ± 0.103	0.372 ± 0.187	0.58 ± 0.2
SM-SVM+FST	0.6 ± 0.125	0 ± 0	0.4 ± 0.125	0.587 ± 0.246
$\sqrt{l}$-NN+FST	0.57 ± 0.134	0 ± 0	0.43 ± 0.134	0.65 ± 0.129
CoM	Match	Omission	Commission	F1
LP+FST	0.617 ± 0.261	0.083 ± 0.136	0.3 ± 0.201	0.563 ± 0.35
RG+FST	0.583 ± 0.157	0.083 ± 0.136	0.333 ± 0.124	0.613 ± 0.106
SM-SVM+FST	0.55 ± 0.201	0 ± 0	0.45 ± 0.201	0.393 ± 0.298
$\sqrt{l}$-NN+FST	0.542 ± 0.148	0 ± 0	0.458 ± 0.148	0.575 ± 0.217
BIK	Match	Omission	Commission	F1
LP+FST	0.536 ± 0.107	0.077 ± 0.08	0.386 ± 0.114	0.556 ± 0.146
RG+FST	0.534 ± 0.13	0.06 ± 0.053	0.406 ± 0.13	0.53 ± 0.206
SM-SVM+FST	0.609 ± 0.075	0 ± 0	0.391 ± 0.075	0.443 ± 0.162
$\sqrt{l}$-NN+FST	0.559 ± 0.074	0 ± 0	0.441 ± 0.074	0.423 ± 0.132
EOrg	Match	Omission	Commission	F1
LP+FST	0.692 ± 0.258	0.075 ± 0.121	0.233 ± 0.222	0.65 ± 0.388
RG+FST	0.725 ± 0.249	0.067 ± 0.11	0.208 ± 0.201	0.69 ± 0.33
SM-SVM+FST	0.792 ± 0.163	0 ± 0	0.208 ± 0.163	0.793 ± 0.152
$\sqrt{l}$-NN+FST	0.717 ± 0.185	0 ± 0	0.283 ± 0.185	0.713 ± 0.174
WBS	Match	Omission	Commission	F1
LP+FST	0.583 ± 0.09	0.07 ± 0.044	0.347 ± 0.09	0.591 ± 0.101
RG+FST	0.64 ± 0.064	0.033 ± 0.043	0.327 ± 0.065	0.606 ± 0.058
SM-SVM+FST	0.632 ± 0.091	0 ± 0	0.368 ± 0.091	0.629 ± 0.108
$\sqrt{l}$-NN+FST	0.467 ± 0.094	0 ± 0	0.533 ± 0.094	0.314 ± 0.189

other knowledge inherent to the academic domain. The knowledge base consists also in 312738 axioms, 49 classes, 96 object properties and 184 data properties, resulting in a $\mathcal{ALEHO}(\mathcal{D})$ knowledge base (encoded in a OWL 2 RL fragment). The learning task, as defined in [2], consisted in predicting affiliations of AIFB staff members to research groups, which we denoted as class-membership relations. All knowledge inherent affiliation relations to research group was removed from the ontology before the experiment. As in [11], negative examples were artificially created (in the same number as positive examples) to mend the lack of training data (due to the Open World Assumption).

A DL reasoner [5] was employed to decide on the concept-membership of individuals to query concepts to be used as a baseline. Performance is measured employing the evaluation indexes proposed in [4], which take into account the specificity deriving from the presence of missing knowledge in the assertions considered as the baseline:

Match. Case of an individual that got the same label by the reasoner and the inductive classifier.

Omission Error. Case of an individual for which the inductive method could not determine whether it was relevant to the query concept or not while it was found relevant by the reasoner.

Commission Error. Case of an individual found to be relevant to the query concept while it logically belongs to its negation or vice-versa.

[5] Pellet v2.3.0 – `http://clarkparsia.com/pellet/`

To provide a term of comparison with methods and results in [2] and [11], we also provide results obtained by the F1-score metric (defined as the harmonic mean of precision and recall). Before evaluating on the test set, parameter tuning was performed for each of the methods via a k-fold cross validation ($k = 10$) within the training set, for finding the parameters with lower classification error in cross-validation. SM-SVM follows the implementation in [16, pg. 223]: the C parameter was allowed to vary in $\{10^{-4}, 10^{-3}, \ldots, 1\}$. The (μ, ϵ) parameters in RG and CM were respectively allowed to vary in $\{10^{-4}, 10^{-2}, 1, 10^{2}, 10^{4}\}$ and fixed to 10^{-4}. The number of neighbors for each node, needed for the construction of the SSG, was allowed to vary in $\{2, 3, 5, 7\}$.

RG, CM and LP give an indication of the uncertainty associated to a specific labeling by associating values in the set $[-1, +1]$ to each node; when such values are ≈ 0 (specifically, when the label was in the set $[-10^{-4}, 10^{-4}]$) we decided to leave the node unlabeled, so to try to provide more robust estimates (and thus a possibly lower commission error and match rates and higher omission error rates). This may happen e.g. when there are no labeled examples within a connected component of the SSG.

4.3 Discussion

From this empirical evaluation, it emerged that the Consistency Method (CM) discussed in Subsect. 3.2 (which we do not report in Tables 1,2 for brevity) may be too conservative: this was suggested by its low Match rate (always reported lower than 0.1) and high Omission rate (always reported higher than 0.9). This may be justified by the fact that its regularizer $||\hat{y}_u||$ is not weighted by any term, unlike Regularization on Graph (RG) (which weights the regularizer $||\hat{y}||$ by means of the term $\mu\epsilon$). The presence of such a regularization term influences the results of transductive methods. Inductive classification methods such as SVM and k-NN define straight decision boundaries in the instance space: a classification result may happen by chance.On the other hand, relaxing binary labels to continuous ones and pulling to 0 labels of unlabeled examples allows to provide more robust labellings: they will be less likely to be determined by chance, and more likely to be statistically justified.

Also from our previous work [12], the choice of the SSG strongly affects final results, and it is likely to be task-dependent: in this case, results obtained by using the Atomics kernel/dissimilarity measure were significantly worse than those obtained with the FST kernel. An explaination is that, in this knowledge base, (atomic) concept-membership relations tend not to carry much information w.r.t. the affiliation prediction task, while the network structure (exploited by the FST kernel) tends to be informative. For example, object properties encoding competence fields tend to encode homophily relations – persons sharing competence fields have a tendency to also have the same research group affiliation. A significant part of the classification error is caused by the fact that persons with not much available information other than their research group affiliation, are now clustered together with nodes where even such information is not available: this is of course non necessarily correct, since lack of information (given by the Open World Assumption) on both individuals does not necessarily imply the presence of a similarity relation between them. A graph kernel might capture similarity relations in case of full information (such as in the SSGs discussed in Sect. 3) but might have problems in case of missing information (such as in this case).

Co-authorship relations to articles, as discussed in Sect. 3, can also encode useful information; however, analysing the results, it emerges that such information is only available from the analysis of inverse roles, which have not been considered in our implementation of the FST kernel. It also emerges that potentially unuseful relations (such as shared first or last names) have concurred in estabilishing similarity relations among individuals. This suggests that simple graph or RDF kernel can fail exploiting the informativeness of potentially useful paths in the ontology's relational graph.

5 Conclusion and Future Work

This work proposes a method for transductive inference for class-membership prediction in Description Logic knowledge bases. It leverages unlabeled examples by propagating class-membership information among similar individuals in the knowledge base. The proposed method relies on graph regularization using quadratic cost criteria, whose optimization can be reduced to solving a (possibly sparse) linear system. In this work, we assumed information propagates homogeneously within the similarity graph defined over a set of individuals in the knowledge base. However, real world ontologies describe domains characterized by *heterogeneity*, either on individuals or on relations among them. For example, persons in the AIFB Affiliations ontology (see Sect. 4) can belong to different categories (e.g. according to their contract type) and be linked by multiple types of relations (for example, given by co-authored articles or shared competence fields), which can have a variable level of informativeness w.r.t. a specific prediction task. Considering multiple similarity measures boils down to defining a cost function with multiple graph-based regularizers, with the side effect of an increased number of parameters. In future work we aim at extending our approach to include multiple similarity relations among different types of instances, and working on methods to efficiently learn the regularization parameters.

References

[1] Belkin, M., Matveeva, I., Niyogi, P.: Regularization and semi-supervised learning on large graphs. In: Shawe-Taylor, J., Singer, Y. (eds.) COLT 2004. LNCS (LNAI), vol. 3120, pp. 624–638. Springer, Heidelberg (2004)

[2] Bloehdorn, S., Sure, Y.: Kernel methods for mining instance data in ontologies. In: Aberer, K., et al. (eds.) ISWC/ASWC 2007. LNCS, vol. 4825, pp. 58–71. Springer, Heidelberg (2007)

[3] Chapelle, O., Schölkopf, B., Zien, A. (eds.): Semi-Supervised Learning. MIT Press (2006)

[4] d'Amato, C., Fanizzi, N., Esposito, F.: Query answering and ontology population: An inductive approach. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 288–302. Springer, Heidelberg (2008)

[5] d'Amato, C., Staab, S., Fanizzi, N.: On the influence of description logics ontologies on conceptual similarity. In: Gangemi, A., Euzenat, J. (eds.) EKAW 2008. LNCS (LNAI), vol. 5268, pp. 48–63. Springer, Heidelberg (2008)

[6] Fanizzi, N., d'Amato, C.: Inductive concept retrieval and query answering with semantic knowledge bases through kernel methods. In: Apolloni, B., Howlett, R.J., Jain, L. (eds.) KES 2007, Part I. LNCS (LNAI), vol. 4692, pp. 148–155. Springer, Heidelberg (2007)

[7] Fanizzi, N., d'Amato, C., Esposito, F.: REDUCE: A reduced coulomb energy network method for approximate classification. In: Aroyo, L., et al. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 323–337. Springer, Heidelberg (2009)
[8] Fanizzi, N., d'Amato, C., Esposito, F.: Induction of robust classifiers for web ontologies through kernel machines. J. Web Sem. 11, 1–13 (2012)
[9] Hu, B., Dasmahapatra, S., Lewis, P.: Semantic metrics. Int. J. Metadata Semant. Ontologies 2(4), 242–258 (2007)
[10] Janowicz, K., Wilkes, M.: SIM-DL$_A$: A novel semantic similarity measure for description logics reducing inter-concept to inter-instance similarity. In: Aroyo, L., et al. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 353–367. Springer, Heidelberg (2009)
[11] Lösch, U., Bloehdorn, S., Rettinger, A.: Graph kernels for RDF data. In: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 134–148. Springer, Heidelberg (2012)
[12] Minervini, P., d'Amato, C., Fanizzi, N.: A graph regularization based approach to transductive class-membership prediction. In: Bobillo, F., et al. (eds.) Proceedings of the 8th International Workshop on Uncertainty Reasoning for the Semantic Web, URSW 2012. CEUR Workshop Proceedings, vol. 900, pp. 39–50. CEUR-WS.org (2012)
[13] Ochoa-Luna, J.E., Cozman, F.G.: An algorithm for learning with probabilistic description logics. In: Bobillo, F., et al. (eds.) Proceedings of the 5th International Workshop on Uncertainty Reasoning for the Semantic Web, URSW 2009. CEUR Workshop Proceedings, vol. 654, pp. 63–74. CEUR-WS.org (2009)
[14] Rettinger, A., Lösch, U., Tresp, V., d'Amato, C., Fanizzi, N.: Mining the Semantic Web: Statistical learning for next generation knowledge bases. Data Min. Knowl. Discov. 24(3), 613–662 (2012)
[15] Rettinger, A., Nickles, M., Tresp, V.: Statistical relational learning with formal ontologies. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) ECML PKDD 2009, Part II. LNCS (LNAI), vol. 5782, pp. 286–301. Springer, Heidelberg (2009)
[16] Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press (2004)
[17] Spielman, D.A., Teng, S.H.: Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In: Proceedings of the 36th ACM Symposium on Theory of Computing, STOC 2004, pp. 81–90. ACM (2004)
[18] Vapnik, V.N.: Statistical learning theory, 1st edn. Wiley (September 1998)
[19] Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: Thrun, S., Saul, L.K., Schölkopf, B. (eds.) NIPS. MIT Press (2003)
[20] Zhou, D., Huang, J., Schölkopf, B.: Learning from labeled and unlabeled data on a directed graph. In: Raedt, L.D., Wrobel, S. (eds.) ICML. ACM International Conference Proceeding Series, vol. 119, pp. 1036–1043. ACM (2005)
[21] Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation. CMU–CALD–02–107. Carnegie Mellon University (2002)

Measuring the Topical Specificity of Online Communities

Matthew Rowe[1], Claudia Wagner[2], Markus Strohmaier[3], and Harith Alani[4]

[1] School of Computing and Communications, Lancaster University, Lancaster, UK
m.rowe@lancaster.ac.uk
[2] Institute for Information and Communication Technologies, JOANNEUM RESEARCH, Graz, Austria
claudia.wagner@joanneum.at
[3] Knowledge Management Institute and Know-Center, Graz University of Technology, Graz, Austria
markus.strohmaier@tugraz.at
[4] Knowledge Media Institute, The Open University, Milton Keynes, UK
h.alani@open.ac.uk

Abstract. For community managers and hosts it is not only important to identify the current key topics of a community but also to assess the specificity level of the community for: a) creating sub-communities, and: b) anticipating community behaviour and topical evolution. In this paper we present an approach that empirically characterises the topical specificity of online community forums by measuring the abstraction of semantic concepts discussed within such forums. We present a range of concept abstraction measures that function over concept graphs - i.e. resource type-hierarchies and SKOS category structures - and demonstrate the efficacy of our method with an empirical evaluation using a ground truth ranking of forums. Our results show that the proposed approach outperforms a random baseline and that resource type-hierarchies work well when predicting the topical specificity of any forum with various abstraction measures.

1 Introduction

In social media applications such as message boards, online social networks or photo sharing sites, communities of users evolve around certain topics. Recent work by Belak et al. [2] examined the longitudinal changes of scientific communities and found *community drift* to be a salient factor where a given community creates new descendent communities that focus on specialised topics of the parent community. An examination of attention patterns (i.e. the factors that correlated with discussion activity and attention to content) undertaken in our prior work [13] found that the specificity of an online community forum's topic was a key feature in discerning attention patterns - e.g. in a community discussing the sport *Golf* the post had to fit the forum's topic exactly, while in a forum discussing *Work and Jobs* this was not a requirement. Recommending community

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 472–486, 2013.

forums to users who are new to a topic allows them to take advantage of the collective wisdom of the community and gain expertise and knowledge, however recommending a community which discusses specialisations of the initial topic may overwhelm the user and a general discussion community around the topic would therefore be more appropriate.

In each of these cases (community drift, attention patterns, and community recommendation) understanding the topical specificity of a community is important for: a) tracking community focus and for new community forums to be suggested to community managers that discuss specialist topics, derived from when a community forum becomes more general in its focus; b) enabling attention-patterns of communities with the same topical specificity to be examined, and therefore the theory that arose from our prior work [13] on community specialisation correlating with attention patterns to be tested, and; c) recommending communities to novice users that are more general in the topics which they focus on, thereby alleviating the potential of overwhelming the user. Given such motivations in this paper we explore the following research question: *Can we empirically characterise how specific a given community is based on what its users discuss?*

To examine this research question we present an approach that combines concept graphs, derived from linked open data, with network-theoretic measures to gauge the abstraction level of concepts discussed by users in community forums from the Irish community message board Boards.ie.[1] Our results indicate that harnessing the linked open data graph can indeed help label the specificity of a forum based on the concepts discussed therein. Our contributions in this paper are three-fold:

1. An approach to measure forum specificity using composite functions, abstraction measures, and concept graphs.
2. Abstraction measures from network-theory that function over concept graphs.
3. Experimental assessment of the performance of different combinations of: a) composite functions, b) abstraction measures, and c) concept graphs, over a community message board platform, and a novel evaluation measure that allows for top-k level-based rankings to be assessed.

We have structured the paper as follows: Section 2 describes related work in measuring and assessing properties of online communities, and existing approaches to measure specificity and abstraction of concepts. Section 3 provides preamble of concept models used to describe online community forums. Section 4 presents our method for measuring the specificity of a community forum by using a composite function to choose the most representative concept for the forum and measuring the concept's abstraction. Section 5 details our experiments in assessing the efficacy of our approach; we explain the evaluation measures used along with the experimental setup, and demonstrate how well our method performs with respect to a random baseline and experiment permutations. Section 6 relates our work to existing related work and highlights the salient findings from this paper and plans for future work, and section 7 concludes the paper.

[1] `http://www.boards.ie`

2 Related Work

In this section we describe related work in the areas of measuring community forum properties before then describing existing work measuring concept specificity and abstraction.

2.1 Measuring Community Forum Properties

Examining the topical properties of communities has been investigated in [2] in which changes in scientific community structures are examined. One salient finding from this work, after examining the longitudinal changes of the communities, is the notion of *community shift* in which a community's topic becomes more general over time, this subsequently leads to the creation of new communities where the prior community, which became more general, is their ancestor. The closest work to ours is described in [1] where Kan et al. model conversation patterns of users on Boarrds.ie and use these patterns to characterise different community forums and hierarchically cluster them, thereby attempting to reproduce the community hierarchy structure. Our work differs, however, in that we provide an empirical assessment of the accuracy of our approach, while [1] rely on an indirect, manual inspection. Additionally, we focus on the topical qualities of forum content, complimenting the work of [1] which only uses user posting behaviour.

The behaviour of online community members was examined in [9] by measuring their behaviour along five dimensions: engagement, popularity, initiation, focus dispersion, and contribution. Rowe et al. found differences between community types (i.e. ideas, communities of practice, teams) in terms of how users behaved. The behaviour measure of focus dispersion is similar to our work as it measures, at a micro-level (i.e. user), the spread of each user in their topics. However, unlike our work it does not consider how specific individual topics are, rather their distribution per user. Term distributions are also assessed in [12] where the topics of web forums and how they change over time are visualised. Trampus and Grobelnik identified topics by choosing the term with the highest Term Frequency-Inverse Document Frequency value in a given forum. In our work we use the notion of Concept Frequency-Inverse Forum Frequency to pick out the most representative concept for a given forum, similar to [12]. Mislove et al. compared the structural properties of Flickr, LiveJournal, Orkut and YouTube [6] by examining link symmetry, power law distributions of edges and nodes, and local clustering of users. Mislove et al. found high degrees of local clustering on the different platforms which contained densely populated subgroups of similar users - i.e. shared many common connections - however the authors focussed on network-structures of social networks, ignoring content and the topical characteristics of the networks.

2.2 Measuring Specificity/Abstraction

Related to our work is research in the area of social tagging systems in which researchers have been interested in understanding the different levels of tag

generality (or tag abstractness) that is essential for, amongst other things, identifying hierarchical relationships between concepts. For example, Schmitz et al. [10] suggest that if resources tagged with t_0 are often also tagged with t_1 but a large number of resources tagged with t_1 are not tagged with t_0, t_1 can be considered to subsume t_0. Heymann at al. [5] represent each tag t as a vector of resources tagged with the tag and compute the cosine similarity between these vectors. This means that they compute how similar the distributions of tags are over all resources. To create a taxonomy of tags, they sort the tags according to their closeness-centrality in the similarity graph. Benz et al. [3] and present a good overview about folksonomy-based methods to measure the level of generality of given tags and evaluate these methods by comparing them with several large-scale ontologies and taxonomies as grounded measures of word generality. Strohmaier et al. [11] present a comparative study of state-of-the-art folksonomy induction algorithms that they applied and evaluated in the context of five social tagging systems.

Unlike the above mentioned work, which aims to understand different levels of tag generality, we aim to understand different levels of community generality, and therefore specialisation. In message boards, like Boards.ie, communities form around certain tags such as sports or soccer and the aim of our work is to assess the specificity level of communities rather than assessing the specificity level of the tags around which communities are formed.

3 Preamble: Concept Models of Online Community Forums

Existing work on community forum properties examined the focus dispersion of users [9] and communities [2] without considering the specificity of the topics being discussed. As we will explain shortly concept graphs can be used to measure the level of specificity of a given community forum, however such a forum must first be represented using a model that can capture the concepts referred to within forum posts. The provided dataset for our experiments, from Boards.ie, includes a set of forums F in which posts are made. Posts are provided as a set of tuples $<u, s, t, f> \in P$, where user u posted message s at time t in forum f. The message s is composed of terms that we use to build the concept models for individual communities. The focus of a community can change and alter over time, therefore we must constrain a community's model to specific time snapshots - e.g. $t' \rightarrow t''$ where $t' < t''$. To ensure the provision of content from time-delimited forum posts we derive the set $S_f^{t't''}$ using the following construct that filters through all relevant posts' contents within the allotted time window:

$$S_f^{t't''} = \{s : <u, s, t, f> \in P, t' \leq t < t''\} \tag{1}$$

Concept models contain the distribution of concepts within a given community forum over an allotted time period - i.e. $t' \rightarrow t''$. Derivation of the concepts requires the use of concept extraction methods, we use Zemanta a third-party tool that provides a concept extraction service and is provided with the forum

posts as input. Given our set of post contents, $S_f^{t't''}$, we must derive concepts that characterise the forum in the time period. We do this by processing each post content $s \in S_f^{t't''}$ using a concept extraction tool $\Psi(s)$ to return the set of concepts related to the content of s. We build the concept model for the community by recording the frequency of concept occurrences in the input posts sets, returning $A_f^{t't''}$. This set is derived using the following construct:

$$A_f^{t't''}[c_i] = |\{c_i : c_i \in \Psi(s), s \in S_f^{t't''}\}| \quad (2)$$

4 Measuring Topical Specificity

Measuring the topical specificity of a community forum requires analysing posted content and then identifying how general or specific the concepts being discussed are. In this context, we interpret a community forum's specificity in relation to its parent forum such that the topics discussed in a child forum are a subset of those discussed in its parent (e.g. *Rugby* discusses more specialised topics than *Sports*).[2] In this section we discuss a range of abstraction measures that gauge how *abstract* a community forum's topics are. As we are interested in the *specificity* of the forum, and given that specificity is the antonym of abstraction, we take the reciprocal of the following abstraction measures ($a(c)$) for individual concepts: $1/a(c)$. In order to process the community forum f we must decide on which concept, based on those found within the forum's content, to process and return the abstraction measure for. As our abstraction measures rely on the network structures of concept graphs they can be expensive to compute, therefore we use composite functions that take a forum's set of concepts, and the frequency of concept occurrences $A_f^{t't''}$, assess each concept in the given set and returns the abstraction measure of the most representative concept. We begin this section by describing how we select which concept to return as the most representative for a forum through our composite functions, before moving on to define the abstraction measures used to assess the level of abstraction of a given concept.

4.1 Composite Functions

As mentioned previously, for a given forum f over a given time step $t' \rightarrow t''$ we are given a collection of concepts derived from posts within the window. We must decide on the best way to select from these concepts a single measure of forum specificity; we therefore provide two such functions for this task.

1. *Concept Frequency:* This function uses the frequency of the concept in the forum to pick out the most commonly discussed concept. The abstraction of the chosen concept is then measured using one of our abstraction measures - which are discussed below - and its reciprocal taken to return the specificity of the forum.

[2] It isnt necessarily the case that the more specialised forum will discuss a *single* topic (e.g. rugby could have children forums Rugby Union and Rugby League for the different codes).

2. *Concept Frequency-Inverse Forum Frequency:* This functions selects the most unique concept discussed in the forum with respect to all forums. This is a modification of the existing Term Frequency-Inverse Document Frequency measure used for term indexation. The *Concept Frequency-Inverse Forum Frequency* of each concept in a given forum is measured and the concept that returns the maximum value is chosen. The abstraction of this concept is then measured and the reciprocal of this value taken as the specificity of the forum. We define the *Concept Frequency-Inverse Forum Frequency* as follows:

$$cf{-}iff(c,f,F) = \frac{|A_f^{t't''}[c]|}{max\Big(\big\{A_f^{t't''}[c'] : c' \in A_f^{t't''}\big\}\Big)} \times \log \frac{|F|}{\big|\{f \in F : c' \in A_f^{t't''}, c' = c\}\big|} \tag{3}$$

4.2 Concept Abstraction Measures

The composite functions decide on which concept to measure based on either: a) the frequency of the concept in the forum, or b) the uniqueness of the concept with respect to the other forums. To measure concept abstraction we define five measures as follows, which either leverage the network structure surrounding a concept or use the semantics of relations in the concept graph.

Network Entropy. Our first measure of concept abstraction ($a(c)$) is based on work by [3] in which tag abstraction is measured through the uniformity of co-occurences. The general premise is that a more abstract tag should co-occur with many other tags, thus producing a higher entropy - as there is more uncertainty associated with the term. In the context of our work we can also apply the same notion, however we must adapt the notion of *co-occurrence* slightly to deal with concepts. To begin with we need to define certain preamble that will allow network entropy, and the below network-theoretic measures, to be calculated, using the same definition as laid out in [4]: let $G = \{V, E, L\}$ denote a concept-network, where $c \in V$ is the set of concept nodes, $e_{cc'} \in E$ is an edge, or link, connecting $c, c' \in V$ and $lb(e_{cc'}) \in L$ denotes a label of the edge - i.e. the predicate associating c with c'. We can define the weight of the relation between two concepts c and c' by the number of times they are connected to one another in the graph: $w(c, c') = |\{e_{cc'} \in E\}|$. From this weight measurement, derived from concept co-occurrence, we then derive the conditional probability of c appearing with c' as follows, using $ego(c)$ to denote the ego-network of the concept c - i.e. the triples in the immediate vicinity of c:

$$p(c'|c) = \frac{w(c, c')}{\sum\limits_{c'' \in ego(c)} w(c, c'')} \tag{4}$$

Now that we defined the conditional probability of c appearing with another concept c', we define the network-entropy of c as follows:

$$H(c) = -\sum_{c' \in ego(c)} p(c'|c) \log p(c'|c) \tag{5}$$

Network Centrality. Concepts in the semantic graph G play a role in connecting other concepts together, allowing agents using a *follow-your-nose* principle to traverse the concept space and find related terms. The importance of a concept in enabling such information flow can be gauged by its *centrality* in the network: the greater the centrality of the concept, the greater its importance. As [3] defines, the notion of centrality also allows a concept's abstraction to be measured, where the more central a concept is to the network, the greater its level of abstraction. Using this notion of centrality equating to abstraction, we provide two centrality measures as follows:

Degree-Centrality. The first measure uses the degree of the concept c to assess its centrality: the greater the degree of the concept, the greater its centrality in the network. The degree of c is derived by returning the ego-centric network of c and measuring its size, we maintain directions of the edges for this measure as we are concerned with the propensity of concept c to be connected from where it appears as the subject of a triple. The cardinality of the ego-centric network is then divided by the number of concepts in the concept-network with 1 subtracted (as c cannot connect to itself):

$$Cent_D(c) = \frac{|\{c' : c' \in V, e_{cc'} \in E\}|}{|V| - 1} \tag{6}$$

Eigenvector Centrality. Our second centrality measure gauges the position of the concept (c) in terms of the eigen structure of the adjacency matrix of the concept graph. The theory behind using such a measure is that the centrality of a concept depends on the centrality of those concepts with which it is connected. Let A denote the adjacency matrix of the concept network where $a_{ij} \in A$, $a_{ij} = 1$ where an edge exists between concept c_i and concept c_j and 0 otherwise. Let x_i denote the centrality score for c_i, where we define x_i as:

$$x_i = \frac{1}{\lambda} \sum_{j=1}^{|A|} a_{ij} x_j \tag{7}$$

We can rewrite Eq. 7 in a vector form such that $\mathbf{x} = \{x_1, x_2, ..., x_n\}$ denotes the vector of centrality measures for concepts $c_1, c_2, ..., c_n$ and rearrange into a solvable form: $\mathbf{Ax} = \lambda\mathbf{x}$. The λ here corresponds to the largest eigenvector of the adjacency matrix of the concept network $\mathbf{A}$, and λ_i corresponds to the eignvector centrality score for concept c_i. Therefore by solving $\mathbf{Ax} = \lambda\mathbf{x}$ we derive the centrality scores for all concepts.

Statistical Subsumption. Our next measure of concept abstraction relies on the semantics of a concept graph to identify concept subsumption. According to Schmitz et al. [10] concept c subsumes (is more general than) concept c' if $p(c|c') > \epsilon$ and $p(c'|c) < \epsilon$ for some threshold ϵ. As we are using the DBPedia graph as our knowledge base for concept relations we can exploit the semantics of the edges to detect subsumption and the hierarchical nature of the relations. For this we utilise SKOS semantics and subclass-of relations within DBPedia in

order to count how many concepts a given concept c is more general than (we use DBPedia datasets as our concept graphs which is explained in the following section).

$$SUB(c) = |\{c' : c' \in V, e_{cc'} \in E, lb(e_{cc'}) \in \{<skos{:}narrower>, <rdfs{:}subClassOf>\}| \tag{8}$$

Key Player Problem. The final measure of abstraction that we use is taken from Navigli & Lapatta [7] and attempts to measure the extent to which a given node in a network is a key player in the network's topology; that is, the extent to which it is important for information flow through the network. To compute this measure we measure the shortest distance - using the Bellman-Ford algorithm - from the concept to every other concept in the network and then take the sum of the reciprocal of these distances. This sum is then normalised by the number of concepts in the network excluding the one under analysis. We define this formally as:

$$KPP(c) = \frac{\sum_{c' \in V, c \neq c'} \frac{1}{d(c, c')}}{|V| - 1} \tag{9}$$

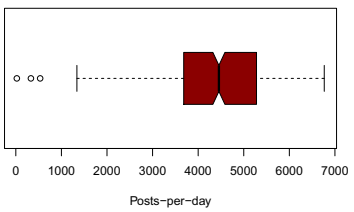
5 Experiments

In this paper we have defined how an online community forum can be modelled using the concepts discussed within its posts. We then described a method to assess the specificity of an online community forum by identifying the most representative concept and measuring the reciprocal of the concept's abstraction. Given the five different abstraction measures used and the two different composite functions, we must select the best combination to measure the specificity of a forum. Additionally, as we are using a concept graph from which to measure the abstraction of a given concept, we must also consider which source to use for the graph and examine how this affects performance.

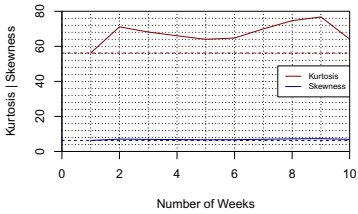
5.1 Experimental Setup

For our experiment we needed to decide which time period to analyse - i.e. setting $t' \rightarrow t''$ - and therefore: a) where to start the period from, and b) how large the period should be. For the former point (start of the period) we counted how many posts were made every day in 2005 and found that the distribution was not normally distributed and was instead bimodal. We fitted a Gaussian mixture model using Expectation-Maximisation and found two Gaussians, thereby rendering our decision to choose a representative date based on the mean of a single Gaussian limited. We instead plotted a boxplot of the distribution, as shown in Figure 1(a), and chose the median of 4,455 posts as being the indicative point of the post distribution, we then selected the date that had 4,455 posts as our *start date*: 23/3/2005. To decide on the window size from this *start date*, we then counted how many posts were made in each forum from the *start date*

within a k-week window, and found the densities to all be normally-distributed with variance in their tails and skews. We wanted to select the most *stable* distribution of posts across the forums and therefore measured the *kurtosis* and the *skewness* of each window size's distribution - as shown in Figure 1(b). We then chose the week that produced the minimum of these measures: 1 week. By choosing this time period we are provided with reduced variation in the forum post distribution and therefore a stable picture, with no large fluctuations, of community activity.



(a) Boxplot of posts density in 2005



(b) Kurtosis and Skewness of density distributions

Fig. 1. Plots of posts-per-day distribution in 2005 (1(a)) and the distribution properties of posts-per-forum in increasing week windows from 23/3/2005 (1(b))

The selected 1 week *experiment* period contained a total of 15,076 posts within 230 forums. We ran the text analysis tool Zemanta[3] based on prior work by Rizzo and Troncy [8], noting that this named entity recognition tool worked best on news story corpora,[4] over the post contents in the time period - 23/3/2005 $\rightarrow$ 30/3/2005 - and used the DBPedia mappings between entities and concepts to generate the concept sets: $A_f^{t't''}$. We extracted 24,752 unique entities from 15,076 posts.

Concept Graphs. Mappings are required between entities and concepts as Zemanta returns DBPedia URIs which may refer to both named entities and concepts. Therefore for the former we must then identify the concepts that the entities relate to. To do this we loaded the Ontology Infobox Types and Articles Categories DBPedia datasets into Jena TDB and queried the store for mappings between an entity and: a) the class that the entity is a type of; and b) the wikipedia category that the entity is related to. We then used two graphs to assess the specificity of a forum: a) the *DBPedia Ontology Graph*, which we refer to as the *Type graph*, containing the class structure in which classes form a strict hierarchy based on `rdfs:subClassOf` relations, and; b) the *SKOS Category Graph*, which we refer to as the *Category graph*, containing the category structure from wikipedia in which categories form a loose hierarchy based on

[3] `http://www.zemanta.com/`

[4] We also note that our domain differs from that of news, but the natural language structure is similar and does not contain abbreviated forms as with Microposts.

`show:broader` relations. Our evaluation therefore, not only looks for the optimum combination of abstraction measure and composite function, but also which concept graph to use: the Type graph or the Category graph.

Table 1. Example rankings of forums in two predicted ranks from model 1 (M1) and model 2 (M2) together with the ground truth. The label function $l(.)$ returns the level of the forum from the ground truth. Our evaluation measures (Kendall τ_b and *Impurity@k*) are provided with the ordered levels as input.

	GT		**M1**		**M2**	
Rank Index	d	$l(d)$	$\hat{d_1}$	$l(\hat{d_1})$	$\hat{d_2}$	$l(\hat{d_2})$
1	a	1	c	2	a	1
2	b	1	d	2	b	1
3	c	2	g	3	c	2
4	d	2	h	3	d	2
5	e	2	a	1	f	2
6	f	2	e	2	g	3
7	g	3	i	3	e	2
8	h	3	b	1	h	3
9	i	3	j	3	i	3
10	j	3	f	2	j	3

Evaluation Measures. To evaluate our approach we use the different combinations of: a) composite functions, b) abstraction measures, and c) concept graphs, to produce a predicted rank ($\hat{\boldsymbol{d}}$) - ordering the most specific forum to the most general - which is then compared against a ground truth rank ($\boldsymbol{d}$). The ground truth rank of the forums is derived from the hierarchical structure of Boards.ie which allows a given forum to be declared as either a parent or a child of another forum, thereby creating a nested structure. In this setting there are three levels that a given forum can be placed in: 1 is most specific, 3 is most general and 2 is in-between. In order to aid comprehension of our evaluation setting we present example rankings produced by two hypothetical models (M1 and M2) in Table 1 along with the ground truth (GT). We refer to this evaluation setting as *level-based ranking* as each model (M1, M2) returns a level ordering (using a label function $l(.)$) derived from the ordering of forums by their specificity values.

We use two evaluation measures for our experiments. The first measure is the Kendall τ_b coefficient which measures the difference in the number of concordant and discordant pairs and normalises this by the number of compared items - accounting for ties: -1 is a perfect negative correlation, 0 is no correlation and 1 is a perfect positive correlation. This measure yields 0.125 and 0.75 for model M1 and model M2 respectively from Table 1, indicating that M2 is better.

The second measure is a novel metric for level-based rankings called *Impurity@k* which assesses the rank up to a given point - i.e. top-k - by gauging the distance from each wrongly positioned forum to its true position in the ground truth, it is therefore equivalent to an *error measure*. The measure has a co-domain of $[0, 1]$ where 0 indicates that there are no wrongly positioned items and 1 indicates that bottom-ranked forums are ranked at the top. *Impurity@k* is derived by taking the set of outlier items (O) - derived as the set of specialised forums that are appear lower-down the rank than more general forums - and working out the distance in the rank between each outlier in the predicted rank and its true position. For model M1 from Table 1 the set of outliers contains

$O = \{a, b, f\}$ while for M2 the set contains $O = \{e\}$. For the true position we use the lowest position of a forum with the same hierarchy level as the outlier - e.g. forum a from M1 is in level 1 which has a lowest position of rank index 2 (forum b in the ground truth). We then gauge the displacement of the forum as a normalised value by setting $|F|$ as the denominator - e.g. for forum a this would be the difference between its rank index in M1 (5) and the lowest rank of a level 1 forum (2) thereby yielding 3/10 given that there are 10 forums under analysis. The normalised displacement values of each outlier are then summed and the average taken. We define this formally as:

$$impurity(k) = \frac{1}{|O|} \sum_{f \in O} \frac{|\hat{\boldsymbol{d}}^k(f) - levelrank(f, \boldsymbol{d}^k|}{|F|} \tag{10}$$

$$levelrank(f, \boldsymbol{d}^k) = \max(\{i : i = \boldsymbol{d}^k(g), l(g) = l(f), g \in F\}) \tag{11}$$

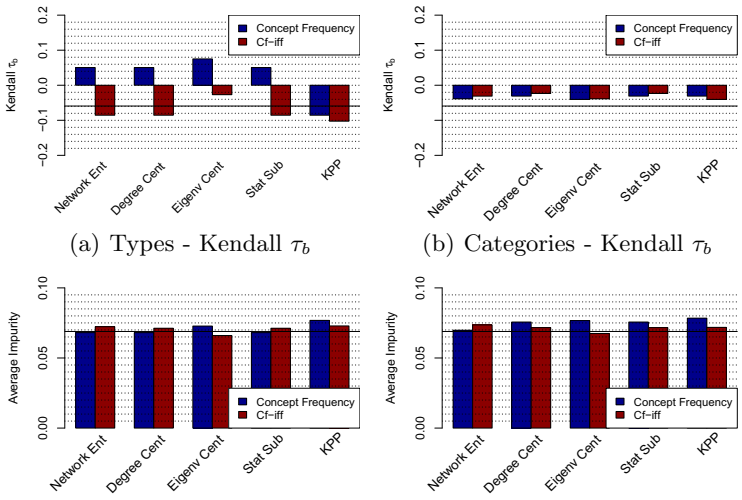
For *Impurity@k* we used six settings for k ($k \in \{1, 5, 10, 20, 50, 100\}$) and averaged the results of these values as a single measure. In doing so we concentrated on the upper-portion of the rank and therefore tested the performance of identifying topically-specific forums. For the rankings in Table 1, M1 and M2 produce *Impurity@10* values of 0.433 and 0.1 respectively (M2 is better).

Baseline Model: Knuth Shuffle. In order to aid comprehension of our results obtained using different model combinations, we compare the performance of each combination to a baseline model constructed using the Knuth Shuffle. To perform the shuffle we took the set of 230 ranked forums and iterated over the set, for each iterated forum we replaced it with a random indexed forum. Baseline measures were found to be 0.069 for *Impurity@k* and -0.0593 for Kendall τ_b.

5.2 Results

Figure 2 presents the results from different combinations of: a) composite functions, b) abstraction measures, and c) concept graphs. We see a marked difference between the performance of the Type graph (Figure 2(a)) and the Category graph (Figure 2(b)) in terms of the Kendall τ_b. We achieve the best performance when predicting the total rank using the Type graph and the Concept Frequency composite function, while using the Concept Frequency-Inverse Forum Frequency (CF-IFF) function achieves the worst performance (worse than our Knuth Shuffle baseline). This indicates that the Type graph contains sufficient information to gauge the specificity of all forums based on the classes of entities found within the forums' content. Using the frequency of the entity-types provides the best combination: achieving the best performance when using Eigenvector Centrality as the abstraction measure - we found this measure to be significantly better with the Concept Frequency function than the closest best performing combination of CF-IFF with Eigenvector Centrality when using the Type graph ($p < 0.05$ using the Sign test).

The *Impurity@k* results for the Type graph (Figure 2(c)) and the Category graph (Figure 2(d)) also show clear differences: the best performing model is the Type graph with CF-IFF and Eigenvector Centrality (lower error than the baseline) despite this model performing poorly when predicting the total rank. The worst performing model was the Category graph, Concept Frequency and the Key Player Problem (KPP) abstraction measure, which also performed poorly when predicting the total rank (kendall τ_b). For our earlier best performing model (Type graph with Concept Frequency and Eigenvector Centrality) we do slightly worse than the random baseline, thereby failing to achieve the best performance when focussing on top-k ranks.



(a) Types - Kendall τ_b (b) Categories - Kendall τ_b

(c) Types - Impurity@k (d) Categories - Impurity@k

Fig. 2. Plots of the results obtained when measuring forum specificity using: a) the DBPedia type graph, and b) the DBPedia SKOS Category graph. The black horizontal line indicates the performance of the Knuth Shuffle random baseline.

Our results indicate that when predicting the complete ranking of communities by their topical specificity using the DBPedia Type graph and Concept Frequency yields the best model (using Eigenvector Centrality). When concentrating on forums that are focussed on a specific topic and identifying forums that are more specific than one another, then the Concept Frequency-Inverse Forum Frequency (CF-IFF) function with the Type Graph and Eigenvector centrality is best. CF-IFF returns the most unique concept for a forum with respect to other forums and using this with the Eigenvector centrality measure returns a low centrality score for any concept on the periphery of the concept graph (returning forum-specific concepts that are unique). We validated our findings using the Mann-Whitney-Wilcoxon test setting the null hypothesis that there is no difference between the specificity values attributed to forums from different levels. We achieved low p-values for the Type Graph with Concept Frequency and Network Entropy, Degree Centrality and Statistical Subsumption ($p = 0.17$,

failing to reject the null hypothesis at $\alpha = 0.1$), while for Eigenvector Centrality with CF-IFF and the Type Graph we found a significant difference between the forum level specificity values ($p < 0.1$).

Table 2 presents top-10 ranks for four model combinations (using the Type Graph as this performed best overall) indicating that different rankings are produced by the models. Similarities are evident when the same composite function is used: *Discworld* appears at the top of both abstraction measures when using Concept Frequency - indicating that the concept selected from this forum has the same specificity levels for both abstraction measures - while *Subscribers*, despite being a mid-level forum, appears towards the top rank of each abstraction measure when using CF-IFF - indicating the existence of a concept unique to this forum which shares a similar specificity level across the measures. Such qualitative analysis indicates that despite the composite functions selecting the same concept to measure the abstraction of, the measures produce, in general, different rankings based on the concept's network position.

Table 2. Forum rankings using the Type Graph and different combinations of composite functions and abstraction measures. The integers in parentheses represent the level of the forum on Boards.ie: 1=most specific, 3= most general.

Concept Frequency		**CF-IFF**	
Network Entropy	**Eigenv' Cent'**	**Network Entropy**	**Eigenv' Cent'**
Discworld (1)	Discworld (1)	Languages (1)	Magic the Gathering (1)
The Cuckoo's Nest (2)	Angling (2)	Hunting (1)	Subscribers (2)
Models (2)	Paganism (1)	File Exchange (2)	Unreal (2)
Slydice Specials (1)	Feedback (2)	Game Threads (1)	LAN Parties (2)
Battlestar Galactica (1)	Personal Issues (2)	Magic the Gathering (1)	World of Warcraft (1)
FS Motors (1)	Mythology (2)	Bangbus (1)	Role Playing (2)
Gadgets (1)	Films (1)	Biology & Medicine (2)	Midwest (2)
FS Music Equipment (1)	Business Managem' (1)	Snooker & Pool (2)	Game Threads (1)
Pro Evolution Soccer (2)	Xbox (1)	Subscribers (2)	GAA (2)
Call of Duty (2)	Help Desk (2)	HE Video Players (1)	Midlands (2)
Anime & Manga (2)	DIT (2)	Discworld (1)	Discworld (1)

6 Discussions and Future Work

Existing research on social tagging systems [10,5] attempts to assess the specificity of a tag in order to build tag hierarchies. Our work is analogous to tag hierarchy construction as it will enable hierarchies of communities to be constructed in a similar vein to [1]. Our future work will compare results for hierarchical clustering of the forums using specificity values from the best performing model - i.e. Eigenvector Centrality with the Concept Frequency composite function and the Type graph - with the clustering from [1] in order to test how well our measures replicate forum hierarchies and structures. When exploring the longitudinal behaviour of scientific communities Belak et al. [2] identified *community shift* as being a prevalent phenomena where a community spawns new communities that are specialisations of their ancestor (parent community). Our work contributes to such explorations by performing specificity analysis of online community forums: if one can track the specificity of a community over

time, then one can identify topic shift and inform community managers as to which new topics could be used for community forums, identifying such events based on the increased generality of a community's topic.

In our prior work [13] we theorised that communities which focused on specific topics showed similar attention patterns - where a post starting a discussion thread had to match the community's topic of interest - while these specific topic communities differed from general discussion communities. The work presented in this paper provides the necessary means for empirically measuring the topical specificity of communities on Boards.ie and other community measure boards. Therefore our future work will involve grouping communities by their topical specificity - measured using Eigenvector Centrality as our abstraction measure, Concept Frequency as our composite function, and the DBPedia Type graph as our concept graph - and examining the attention patterns of specific communities vs general communities, thereby proving, or disproving, our earlier theory from [13]. In this paper we have considered a *semantic approach* to measure the topical specificity of online community forums, however there is the potential to also examine an alternative purely *social approach*: for instance, based on the notion of *Statistical Subsumption* which we explored as one of our abstraction measures, one could identify forum f_a as being more general than forum f_b if the set of authors who created posts on f_b is a subset of the authors who authored posts in f_a. Such insights and potentials for future work have been afforded as a result of the work discussed within this paper.

7 Conclusions

In this paper we presented an approach to measure the topical specificity of online community forums that used abstraction measures which functioned over concept graphs and composite functions to return a representative concept for a community, and thereby its specificity level. Motivated by our research question (*Can we empirically characterise how specific a given community is based on what its users discuss?*) the empirical assessment of forum specificity through our experiments showed the divergent performance between different composite functions, abstraction measures and concept graphs, where the use of a resource type-graph derived from the DBPedia Type Ontology provided a useful resource for predicting a complete ranking of forums by their specificity levels, outperforming the SKOS Category structure. We also found that using the Eigenvector Centrality measure and the Concept Frequency-Inverse Forum Frequency function provided the best combination for identifying differences in topic-specific communities to be discerned - this latter assessment being measured using our novel evaluation metric $Impurity@k$ that accounts for top-k ranked levels.

The results and findings from this work will inform our future work on examining attention patterns of online communities, and also enable the longitudinal assessment and tracking of a community forum's topical specificity, thereby allowing new communities to be recommended to managers based on topical generalisation - a natural life-cycle of communities as put forward by [1].

Acknowledgment. This work was supported in part by a DOC-fForte fellowship of the Austrian Academy of Science to Claudia Wagner.

References

1. Kan, A., Chan, J., Hayes, C., Hogan, B., Bailey, J., Leckie, C.: A Time Decoupling Approach for Studying Forum Dynamics. World Wide Web Internet and Web Information Systems, 1–24 (2011)
2. Belak, V., Karnstedt, M., Hayes, C.: Life-cycles and mutual effects of scientific communities. Procedia - Social and Behavioral Sciences 22, 37–48 (2011)
3. Benz, D., Körner, C., Hotho, A., Stumme, G., Strohmaier, M.: One tag to bind them all: Measuring term abstractness in social metadata. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) ESWC 2011, Part II. LNCS, vol. 6644, pp. 360–374. Springer, Heidelberg (2011)
4. Guéret, C., Groth, P., Stadler, C., Lehmann, J.: Assessing linked data mappings using network measures. In: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 87–102. Springer, Heidelberg (2012)
5. Heymann, P., Garcia-Molina, H.: Collaborative creation of communal hierarchical taxonomies in social tagging systems. Technical Report 2006-10, Computer Science Department (April 2006)
6. Mislove, A., Marcon, M., Gummadi, K.P., Druschel, P., Bhattacharjee, B.: Measurement and analysis of online social networks. In: SIGCOMM Conference on Internet Measurement, IMC 2007, pp. 29–42 (2007)
7. Navigli, R., Lapata, M.: An experimental study of graph connectivity for unsupervised word sense disambiguation. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 32(4), 678–692 (2010)
8. Rizzo, G., Troncy, R.: Nerd: Evaluating named entity recognition tools in the web of data. In: Workshop on Web Scale Knowledge Extraction (WEKEX 2011), Bonn, Germany, pp. 1–16 (2011)
9. Rowe, M., Fernandez, M., Alani, H., Ronen, I., Hayes, C., Karnstedt, M.: Behaviour analysis across different types of enterprise online communities. In: ACM Web Science Conference (2012)
10. Schmitz, P.: Inducing ontology from flickr tags. In: Proceedings of the Workshop on Collaborative Tagging at WWW 2006, Edinburgh, Scotland (May 2006)
11. Strohmaier, M., Helic, D., Benz, D., Körner, C., Kern, R.: Evaluation of folksonomy induction algorithms. Transactions on Intelligent Systems and Technology (2012)
12. Trampuš, M., Grobelnik, M.: Visualization of online discussion forums. In: Workshop on Pattern Analysis Applications (2010)
13. Wagner, C., Rowe, M., Strohmaier, M., Alani, H.: Ignorance isn't bliss: an empirical analysis of attention patterns in online communities. In: AES Conference on Social Computing (2012)

Broadening the Scope of Nanopublications*

Tobias Kuhn[1,2], Paolo Emilio Barbano[3], Mate Levente Nagy[4], and Michael Krauthammer[4,1]

[1] Department of Pathology, Yale University, New Haven, USA
[2] Chair of Sociology, in particular of Modeling and Simulation, ETH Zurich, Switzerland
[3] Department of Mathematics, Yale University
[4] Program for Computational Biology and Bioinformatics, Yale University
kuhntobias@gmail.com,
{mate.nagy,paoloemilio.barbano,michael.krauthammer}@yale.edu

Abstract. In this paper, we present an approach for extending the existing concept of nanopublications — tiny entities of scientific results in RDF representation — to broaden their application range. The proposed extension uses English sentences to represent informal and underspecified scientific claims. These sentences follow a syntactic and semantic scheme that we call AIDA (Atomic, Independent, Declarative, Absolute), which provides a uniform and succinct representation of scientific assertions. Such AIDA nanopublications are compatible with the existing nanopublication concept and enjoy most of its advantages such as information sharing, interlinking of scientific findings, and detailed attribution, while being more flexible and applicable to a much wider range of scientific results. We show that users are able to create AIDA sentences for given scientific results quickly and at high quality, and that it is feasible to automatically extract and interlink AIDA nanopublications from existing unstructured data sources. To demonstrate our approach, a web-based interface is introduced, which also exemplifies the use of nanopublications for non-scientific content, including meta-nanopublications that describe other nanopublications.

1 Introduction

Nanopublications have been proposed to make it easier to find, connect and curate core scientific statements and to determine their attribution, quality and provenance [6]. Small RDF-based data snippets — i.e. nanopublications — rather than classical narrative articles should be at the center of general scholarly communication [10]. In contrast to narrative articles, nanopublications support data sharing and mining, allow for fine-grained citation metrics on the level of individual claims, and give incentives for crowdsourced community efforts. In this paper, we propose an extension that allows for informal and underspecified representations and broadens the scope of the nanopublication approach.

* The work presented in this paper has been supported by the National Library of Medicine grant 5R01LM009956.

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 487–501, 2013.

The novelty of nanopublications lies in the combination of four ideas: (1) to subdivide scientific results into minimal pieces, (2) to represent these results — called *assertions* — in an RDF-based formal notation, (3) to attach RDF-based provenance information on this "atomic" level, and (4) to treat each of these tiny entities as a separate publication. Number (2) strikes us as problematic: Requiring formal representations for scientific results seems to be unrealistic in many cases and might restrict the range of practical application considerably. On the other hand, we think that the approach would be highly beneficial even if this restriction is dropped, and that at the same time it would become much more broadly applicable. Specifically, we propose to allow authors to attach English sentences to nanopublications, thus allowing for informal representations of scientific claims. We previously sketched this approach in a position paper [8].

To illustrate and motivate our approach, let us consider the following fictitious scenario: Giuseppe is a researcher in the biomedical area. Just now, he came to think of the possibility that gene X might accelerate the late stage of the course of disease Y, and he decides to investigate this further. The first questions that Giuseppe faces are: Has somebody else thought of gene X as a late-stage accelerator of disease Y? If so, is it an established fact or an open, maybe even controversial question? How much evidence is there on either side (i.e. the statement being true vs. false)? Who has worked on this question and what are their positions? With the current Web, it takes Giuseppe hours, probably days to answer these questions. As it so happens, a researcher named Isabelle is asking herself the same question. One of her experiments, designed for an entirely different purpose, showed some evidence that gene X might speed up the final stage of disease Y. She wonders whether this would be a new finding or not, but she only has time for a quick Web search, which does not reveal anything.

With our approach, this scenario would turn out differently in the future. Giuseppe and Isabelle would each access a nanopublication portal to enter their hypothesis "gene X accelerates the late stage of the course of disease Y." The system would retrieve related nanopublications, in particular those with matching sentences, including the ones that use different wording to express the same meaning (applying a mixture of automatic clustering and crowdsourcing). In an instant, the system would compile and present the relevant information: the amount of existing research; whether the statement is open, settled, or controversial; supporting and opposing researchers and evidence; and references to the most relevant articles. This saves Giuseppe days of work, and Isabelle gets a quick answer to her question. Furthermore, she can contribute to this global scientific knowledge base by publishing a nanopublication referring to her experiment that gave some weak evidence in favor the statement. This takes her only a few minutes and might later have a positive impact on her citation record.

These examples show that informal representations of scientific statements (i.e. plain English sentences) are sufficient for many purposes. In fact, it would most certainly have been difficult for Giuseppe and Isabelle to come up with a formal representation for their hypothesis. It is known that even people who use ontology languages professionally often mix up such fundamental concepts as

existential and universal quantification [11]. It is beyond question that formal representations have advantages that cannot be achieved with informal sentences, and we should express scientific claims in RDF form whenever possible. However, we think that in cases where formal representations are not practical (which could very well be the majority of cases), the scientific research community can derive substantial benefits from nanopublications with informal claims.

In the remainder of this paper, we will introduce a formalism for using English sentences in nanopublications. We discuss the constraints on such sentences and how they are generated, processed, and interlinked. We present evaluations on the ease of manually and automatically generating such nanopublications, and on sentence clustering for automatically linking related scientific claims.

2 Background

There are only a few existing approaches of embedding scientific results as separate English sentences in formal structures. They are briefly outlined below, and contrasted with our approach in the next section.

SWAN (Semantic Web Applications in Neuromedicine) [2] is an ontology that evolved from a web platform called *Alzforum*, which has been used by the Alzheimer research community since 1996 to discuss their ideas and findings [3]. Similar to the approach to be presented here, SWAN provides a formal RDF-based scaffold for scholarly communication, while using informal English sentences to describe claims and hypotheses. Another example is EXPO, an ontology for scientific experiments [13]. In this model, each research hypothesis has both a formal definition and a natural language equivalent, and the latter typically has the form of a single English sentence. As a third example, GeneRIF is a dataset describing gene and protein functions.[1] Each GeneRIF entry consists of a gene ID, a publication reference, and — most importantly — a short English sentence of less than 255 characters describing a function of the given gene. We will use this dataset in one of our evaluations.

In addition to the nanopublication initiative, there are a number of related approaches of formally representing scientific findings. The Biological Expression Language (BEL) is "a language for representing scientific findings in the life sciences in a computable form."[2] It is embedded into a relatively complex scripting language called *BEL Script*, where the formal statements can also be linked to sentences of the publication they were derived from. Other approaches focus on hypotheses that are automatically generated [14]. A different application scenario is employed by an approach called *structured digital abstracts* [12], which should make formal representations of main scientific results sufficiently simple to require them directly from paper authors. These formal abstracts could be submitted, reviewed, and published together with their papers.

The particular kind of English sentences that our approach uses can be considered a controlled natural language (CNL) [15]. A CNL is a language that is

[1] `http://www.ncbi.nlm.nih.gov/gene/about-generif`
[2] `http://www.openbel.org`

based on a certain natural language, while being more restrictive concerning lexicon, grammar, and/or semantics. Previous work investigated the use of a formal CNL to write scientific abstracts that can be automatically translated into logic [9]. The CNL to be presented below is, however, of an essentially different type: It is much less restricted and is not designed for automatic interpretation.

3 Approach

Our approach of using English sentences to describe scientific results differs from existing approaches — such as SWAN, EXPO, and GeneRIF — in the following four respects:

1. Our intended application range is very broad, covering science as a whole and beyond.
2. In our conceptualization, sentences exist independently from authors. In a certain sense, a sentence exists even if it has not yet been uttered by anybody. Conversely, a particular sentence might have been said by different persons at different points in time. None of these persons "owns" the sentence, but the sentence has an existence on its own and just happens to be mentioned (i.e. claimed, challenged, refuted, related, etc.) by people from time to time.
3. The sentences of our approach are not just any English sentences; we are more specific about what such sentences have to look like. We will introduce the concept of "AIDA sentences," which can be considered a controlled natural language.
4. Our approach allows for a (quasi-)continuum from fully informal to fully formal statements. Natural sentences can be assigned partial or complete formal representations in the form of RDF graphs, combining the advantages of natural and formal representations.

Number 2 might look like a purely philosophical issue, but it actually has very concrete consequences to our approach. For example, since sentences have their own independent existence, it is natural to give them URIs, making them first-class citizens in the RDF world. The formalization continuum of number 4 is shown in Figure 1, which also illustrates how statements can be interlinked regardless of their level of formality, and how they can be part of nanopublications. As a very simple (and in this sense untypical) example of a scientific claim, we borrow the sentence "malaria is transmitted by mosquitoes" from previous articles on nanopublications [6].

3.1 An Ocean of Nanopublications

Before we move on to explain the details of our approach, let us describe the general nanopublication idea in some more detail. We deliberately present it from our own particular perspective, which embodies a view that is broader than the original one, but in a straightforward way. As motivated above, we believe that the application range of nanopublications could be much broader than what they

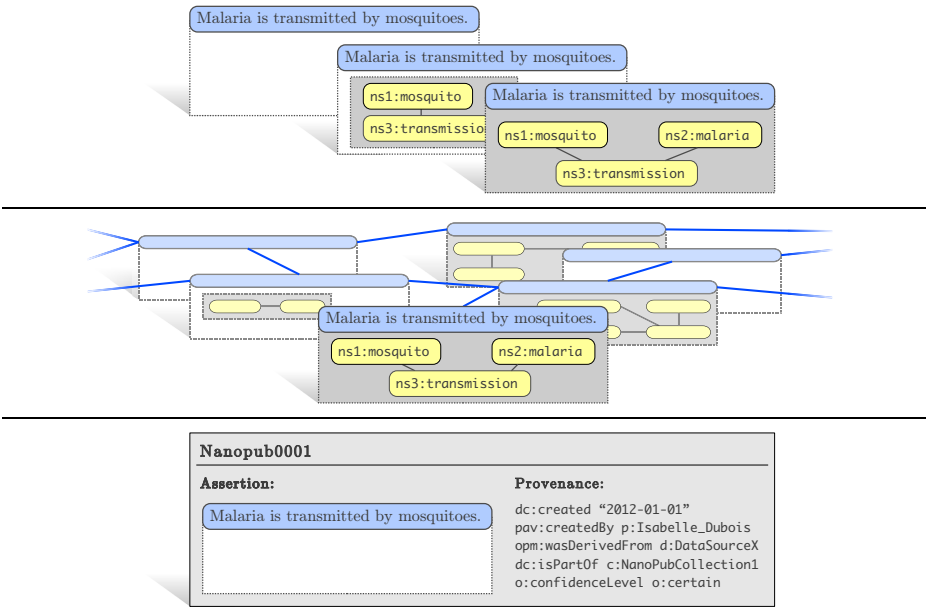


Fig. 1. In our approach, there is a continuum from formal to informal statements (top), which can be linked regardless of their level of formality (middle), and which can be asserted in nanopublications (bottom)

were initially designed for (which is, by the way, very similar to the origins of the RDF standard). Basically, nanopublications could become the basis for the *entire* Semantic Web. Whatever information one wants to share, it could be published in the form of one or more nanopublications. These can include scientific claims and experimental data, but also opinions, social relationships, events, properties of other nanopublications ("meta-nanopublications"), and much more. In general, they are supposed to come from a number of channels, including the following:

1. Authors provide nanopublications for their own (scientific) results.
2. Users create meta-nanopublications by assessing, interlinking, and correcting existing nanopublications, claims, authors, and other relevant entities.
3. Curators generate nanopublications for results others have found.
4. Data mining (especially text mining) generates new nanopublications from existing unstructured data sources.
5. The data contained in existing structured data sources is exported into the nanopublication format.
6. Bots generate new nanopublications by crawling through the mass of existing ones and inferring obvious and not-so-obvious new relations.

Channels 3, 4, and 5 are very important in the beginning to attain critical mass, but afterwards 1, 2, and 6 would gain importance. Channels 1, 2, 3, and 5 typically generate high-quality nanopublications, whereas 4 and 6 tend to have

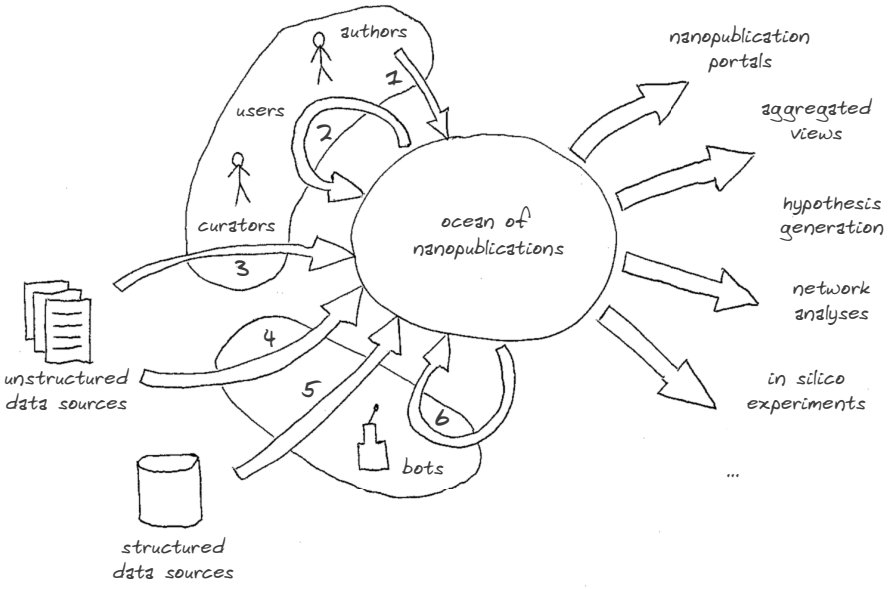


Fig. 2. Channels creating and using nanopublications

lower quality. Figure 2 shows these different channels and sketches some possible applications that consume nanopublications. In the middle of the picture, there is an ocean of nanopublications. At the moment, this is no more than a puddle, but the different channels should enlarge it to massive dimensions. A crucial question is whether these channels can produce enough nanopublications at the initial stage to let the ocean grow to a certain critical mass, at which point it would produce enough advantages for all participants to allow the system to run on its own. For that reason, the evaluations we will present below focus on the creation of nanopublications.

The agents that produce nanopublications can be humans or bots. We use the term *bot* to denote "robots without a body" or "named computer programs," i.e. agents that are made up only of software. Robot scientists [7] could become another important type of agent in the future.

3.2 AIDA

Let us turn now to the core of our approach, i.e. the particular kind of English sentences. They have to follow a scheme that we call AIDA (pronounced like the opera): the sentences have to be Atomic, Independent, Declarative and Absolute:

- **Atomic:** a sentence describing one thought that cannot be further broken down in a practical way
- **Independent:** a sentence that can stand on its own, without external references like "this effect" or "we"

- **Declarative:** a complete sentence ending with a full stop that could in theory be either true or false
- **Absolute:** a sentence describing the core of a claim ignoring the (un)certainty about its truth and ignoring how it was discovered (no "probably" or "evaluation showed that"); typically in present tense

The sentence "malaria is transmitted by mosquitoes," which we have encountered above, is an example of an AIDA sentence. The first three criteria basically reflect the nanopublication idea when applied to natural language instead of RDF.

The last AIDA criterion might look suspicious: After all, uncertainty is an essential aspect of scientific results. We are *not* proposing to omit this aspect, but it should be recorded separately in the provenance part of the nanopublication and should not be part of the sentence. We do not have a concrete proposal at this point, but the ORCA model of uncertainty [4] seems to be a very good candidate for integration. Once integrated, a user interface for creating AIDA nanopublications could look as follows:

☐ We hypothesize that this statement might be true:
☒ We think this statement is probably true:
☐ We think this statement is an established fact:

Malaria is transmitted by mosquitoes.

As mentioned above, each AIDA sentence should get its own URI to make it a first-class citizen in the RDF world. String literals would not work out, as we want to establish relations between sentences, and RDF literals are not allowed in subject position of triples. An additional requirement is that the actual AIDA sentence should be extractable from its URI without consulting external resources, and vice versa. These requirements can be met by a straightforward URI encoding:

```
http://purl.org/aida/Malaria+is+transmitted+by+mosquitoes.
```

No central authority is needed to approve new statements, but everybody can make up such URIs and immediately use them. We are aware that this goes against existing recommendations of keeping URLs opaque, but we think that when adding something essential such as a natural/formal continuum, a deliberate deviation from previous good practices is justified.

AIDA sentences can be interlinked by relations such as `hasSameMeaning`. The semantics of such relations is relatively straightforward for formal languages, but much less so for natural language, which is inherently vague and ambiguous. We employ a very pragmatic definition. In a nutshell, the semantics of an AIDA sentence is defined as the most frequent meaning English speakers assign to it. More specifically, in order to find *the* meaning of a given AIDA sentence, we mentally give it to all English speakers in the world, disregarding those who would say that they do not understand it. The remaining ones we ask (again mentally) about the most plausible meaning they would intuitively assign (without giving context information, as AIDA sentences are supposed to be independent).

The most frequent of the resulting meanings is considered *the* meaning of the AIDA sentence. On this basis, two sentences satisfy the `hasSameMeaning` relation, for example, if and only if we end up with equivalent meanings after going through the above mental exercise for each of them.

3.3 Creating and Clustering AIDA Nanopublications

The obvious channels that are supposed to provide us with AIDA nanopublications are channels 1, 3, and 4: Authors and curators manually write AIDA sentences, and text mining approaches automatically extract AIDA sentences from existing texts. However, bots can also produce AIDA nanopublications, inferring them from existing ones and interrelating them using NLP techniques (channel 5). In addition, users of nanopublication portals can link and correct existing AIDA sentences (channel 2). In the case of channel 6, we typically get formal representations "for free," but having complementary AIDA sentences can still be helpful for humans to make sense of the respective claims. The evaluation to be presented below focuses on channels 1, 3, 4, and 6.

As motivated in the introduction, the main benefit of AIDA nanopublications comes from interlinking them and relating them to other entities. The first problem we are facing is that a typical scientific statement can be expressed in more than one way. We, therefore, cannot expect that two AIDA sentences with the same meaning use exactly the same wording. To solve this problem, we propose to use a mixture of automatic clustering and crowdsourcing. The clustering has the function of finding candidate sentences that seem to have similar or even identical meanings for a given AIDA sentence. Users of nanopublication portals can then filter out the false positives (ideally by single mouse clicks). These user responses would be published as nanopublications via channel 2. One of our studies to be introduced below shows results on the quality of automatic clustering of sentences.

4 Implementation

Below, we introduce a prototype of a nanopublication portal and give some details on the RDF representations.

4.1 Nanopublication Portal

Figure 3 shows a prototype of a nanopublication portal called *nanobrowser* that we are developing to demonstrate our approach. It is based on Apache Wicket and the Virtuoso triple store, and its source code is available online.[3] Nanobrowser is in fact more than just a browser and could be called a *scientific/social/distributed/semantic wiki*. Users are presented small buttons such as "I agree," which generate and publish meta-nanopublications by single mouse clicks.

[3] `http://purl.org/nanobrowser`

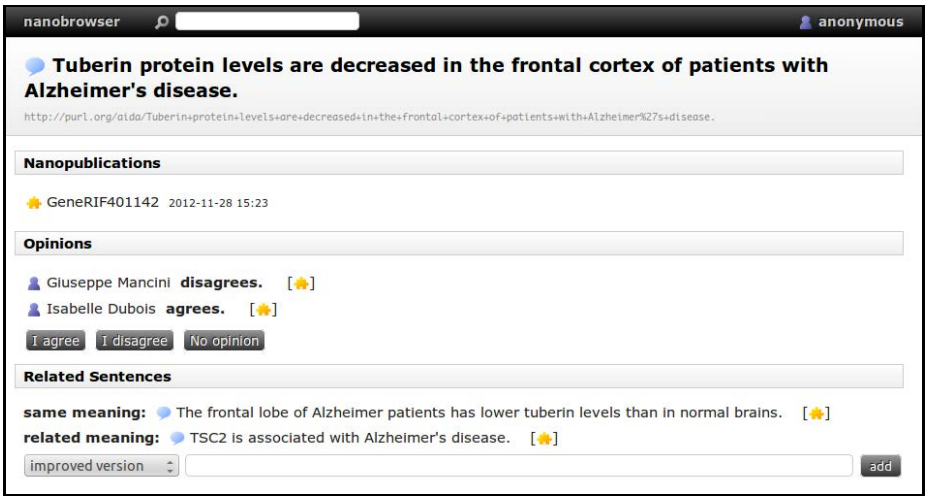


Fig. 3. Screenshot of the nanobrowser interface

We still have to investigate what kind of opinions scientists would like to give. Something like "I am (not) convinced" might be better suited than "I (dis)agree." In any case, users can in this way publish many nanopublications with little effort while browsing the knowledge base.

The screenshot of Figure 3 shows a page that summarizes the available information about a particular scientific statement. The shown example was automatically extracted from GeneRIF as part of the evaluation to be described below. The page shows related sentences and opinions from researchers, each associated with the meta-nanopublication that established the respective relation (yellow jigsaw puzzle icon). Users can track down the origin of every piece of information to see how and by whom it was published.

4.2 Extended Nanopublication Notation

Let us have a brief look under the hood, i.e. the actual RDF representation of nanopublications. The core part of a standard nanopublication is an *assertion* in the form of a named graph [1]:

```
<> {
  :Pub1 np:hasAssertion :Pub1_Assertion .
  ...
}
:Pub1_Assertion { ... }
```

The curly brackets after `:Pub1_Assertion` would contain the actual assertion in the form of a set of RDF triples. To allow for informal and underspecified assertions using AIDA sentences, we have to use a slightly more complex structure. With our approach, assertions consist of two subgraphs: a head and a body, where the body represents the actual (possibly unknown) formal representation:

```
<> {
  :Pub1 np:hasAssertion :Pub1_Assertion .
  :Pub1_Assertion np:containsGraph :Pub1_Assertion_Head .
  :Pub1_Assertion np:containsGraph :Pub1_Assertion_Body .
  ...
}
```

The head part is used to refer to different representations of the given assertion, such as the formal representation in the form of a named RDF graph or a natural representation in the form of an AIDA sentence:

```
:Pub1_Assertion_Head {
  :Pub1_Assertion
      npx:asSentence aida:Malaria+is+transmitted+by+mosquitoes. ;
      npx:asFormula :Pub1_Assertion_Body .
}
```

We can — but we are not obliged to — add a formalization of the given claim:

```
:Pub1_Assertion_Body { ... }
```

Partial representations can be defined in a straightforward way with the help of subgraphs, and we can use `rdf:about` to define that a certain entity must be part of a formalization without specifying a concrete triple:

```
:Pub1_Assertion_Body np:containsGraph :Pub1_Assertion_Body_Partial .
:Pub1_Assertion_Body rdf:about ns:malaria .
```

Later nanopublications can refer to `Pub1_Assertion_Body` to augment or correct the existing representation.

Overall, this extension is backwards compatible as long as `containsGraph` relations are considered when retrieving the assertion triples, and allows for a uniform and general representation of informal, underspecified, and fully formal assertions.

5 Evaluation

It is obvious from Figure 2 that there are many aspects to evaluate, most of which we have to leave to future work. As motivated above, we focus our evaluation on the left hand side of the picture, since this seems to be the critical part for the initial stage of our approach. The studies described below test some of the important aspects of the generation of AIDA nanopublications by both, humans and bots. Detailed supplementary material is available online.[4]

5.1 Manual Generation of Nanopublications

Our first evaluation tests aspects of channels 1 and 3, as described in Section 3: How easy or difficult is it for authors or curators to create nanopublications for their own or others' scientific results?

[4] `http://purl.org/tkuhn/aidapaper/supplementary`

To that aim, we asked biomedical researchers to rewrite short texts from scientific abstracts as one or more AIDA sentences. In a sense, these participants resemble curators who are supposed to create nanopublications for existing scientific results. With respect to the lack of training and experience, however, they rather resemble authors who occasionally create nanopublications for their own results. Some of the tested aspects are therefore relevant to channel 1 and others to channel 3.

Design. To get short original texts, we searched PubMed for articles with structured abstracts, i.e. abstracts that are divided into different parts like Introduction and Conclusions. According to our experience, the Conclusions section typically describes the general high-level scientific claims that correspond to the assertions of nanopublications. We took a random sample of PubMed abstracts that have a Conclusions section, excluding those that are not understandable without the broader context. Some of the resulting texts were shortened, so that each of them would lead to at most three AIDA sentences. We ended up with five such short texts.

We recruited 16 participants for this user study, all scientists with a background in biology and medicine. They had never heard of the AIDA concept before. They were directed to an online questionnaire that consisted of three parts: the first part briefly explained the AIDA concept; the second part showed the five short texts and asked for one to three AIDA sentences for each of them; the last part asked about the experienced difficulty of understanding the AIDA concept and of performing the rewriting tasks.

Below, one of the five short texts is shown as an example, together with two corresponding AIDA sentences as we got them from one of our participants:

> *Original text:* The results of this study showed that the hepatic reticuloendothelial function is impaired in cirrhotic patients, but the degree of impairment does not differ between patients with and without previous history of SBP. [PMID 11218245]
>
> *AIDA 1:* The hepatic reticuloendothelial function is impaired in cirrhotic patients.
>
> *AIDA 2:* The degree of hepatic reticuloendothelial function impairment does not differ between cirrhotic patients with and without previous history of SBP.

Results. The 16 participants created 163 sentences in total. On average, they needed 15.3 minutes to complete the study. This means that an average sentence only required 90 seconds to be created, including the initial overhead to learn the AIDA concept. We checked each of the 163 sentences whether it complied with the AIDA restrictions and whether it was an accurate representation of the original text. Some sentences contained minor mistakes, such as typos or missing copulas (e.g. "X helpful in Y" instead of "X *is* helpful in Y"). For the sentences that were not compliant with AIDA, we also checked which of the requirements they violated. The result for each sentence was based on two independent manual evaluations. Figure 4 shows the results.

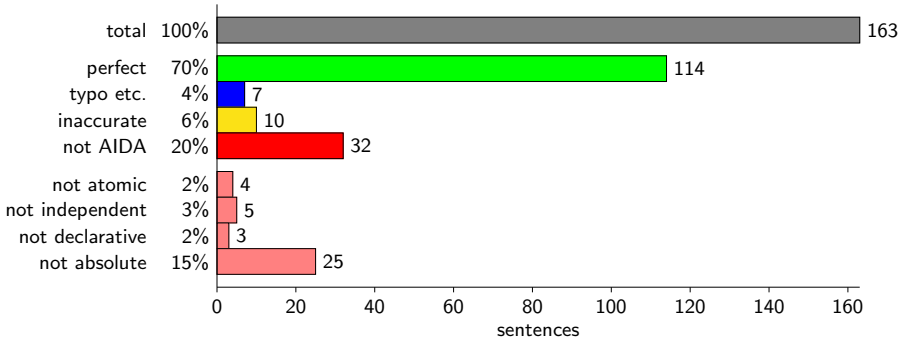


Fig. 4. Quality of the sentences created within the user study

70% of the resulting sentences were perfect, which means that they complied with the AIDA restrictions and were accurate representations of the original texts. An additional 4% were perfect, except that they contained typos and other minor mistakes. 6% were AIDA-compliant but not accurate with respect to the original. 20% violated one or several of the AIDA restrictions, mostly the absoluteness criterion (15%).

As there is no straightforward way of assessing the recall for partially correct sentences, we calculated the "recall for correct tasks" by looking only at the tasks for which a particular participant wrote no incorrect sentence (except for typos): 96% of these sentences covered all information of the initial text.

Next, we can have a look at the subjective experience of the participants, who had to specify their difficulty of understanding the AIDA concept and of performing the tasks. The possible answers were "very difficult," "difficult," "easy," and "very easy." All participants replied that understanding the concept of AIDA sentences was "easy" (but not "very easy"). Nobody found it difficult or very difficult. The task of rewriting the short texts in AIDA format was of medium difficulty, with a tendency towards easy: ten out of the 16 participants found it "easy"; the remaining six found it "difficult."

5.2 Automatic Generation of Nanopublications

Our second evaluation targets specific aspects of channels 4 and 6, as described in Section 3: How can we automatically extract nanopublications from text resources and then automatically relate them to each other?

For this part of the evaluation, we used the GeneRIF dataset,[5] which contains sentences that describe the functions of genes and proteins. We evaluated the quality with which we can extract AIDA nanopublications from this dataset. Then, we investigated how well we can cluster them according to their similarity.

[5] `ftp://ftp.ncbi.nih.gov/gene/GeneRIF/generifs_basic.gz`

Design. To automatically extract AIDA sentences, we tried to detect GeneRIF sentences that already follow the AIDA scheme. This was implemented as a set of simple regular expressions that filter out sentences that are unlikely to be AIDA-compliant. As many GeneRIF sentences start with phrases such as "these results clearly indicated that" or "the authors propose that," and do therefore not adhere to the absoluteness criterion, we defined additional regular expressions to identify and remove such sentence beginnings, so that the remaining sentence texts could be treated as AIDA candidates. The resulting extraction program was a simple script containing these regular expressions.

During the development of this extraction program, the GeneRIF dataset as of September 2012 was used, which included roughly 750 000 entries (including duplicate sentences). Upon completion of the extraction program in November 2012, we downloaded the latest version of GeneRIF, which had 16 865 new entries. We then ran our extraction program on these new entries, which led to 4 342 AIDA nanopublications. From these, we took a random sample of 250 unique sentences, which we manually checked for AIDA compliance.

As a next step, we extracted AIDA sentences from the entire GeneRIF dataset (119 088 unique sentences) and added the ones we obtained from the user study described above (94 unique sentences). On average, each of the five user study tasks led to 18.8 unique statements, which were closely interrelated in terms of meaning but used different wording. We then applied a clustering algorithm on the combined set of sentences to evaluate the quality with which similar or equivalent sentences can be grouped, using the user study sentences as a kind of gold standard. As input for the clustering algorithm, we transformed the sentences into word vectors of *tf-idf* values.

A plethora of unsupervised clustering methods have been developed in statistics and machine learning [5]. Most techniques require the user to define the number of clusters in advance, and those that do not, often require tuning of various parameters. Here, we use our own clustering method, specifically designed for sentence similarity, which deals with the large variation of neighbor density we observed in word vector space. Our algorithm goes as follows: (1) Given a point X (i.e. a sentence) in our dataset, we build a model of its local environment U_X by choosing a two-level set of nearest neighbors. (2) We repeatedly partition U_X with k-means (with $k = 3$) and consider the median distance d_X of the elements of the cluster C_X containing the base-point X. (3) If d_X lies above a given threshold, point X is considered an "isolate" and the cluster is discarded (this is to avoid "loose" clusters consisting of unrelated sentences, connected by low frequency words).

Results. Figure 5 shows the results of our automatic extraction of AIDA nanopublications from the GeneRIF dataset. The general results look very similar to the ones from the user study. 71% of the resulting AIDA sentences fully complied with the AIDA restrictions; an additional 3% did so, but contained minor mistakes such as typos. In contrast with the user study, non-atomic sentences were relatively frequent (14%). Creators of GeneRIF sentences are probably not

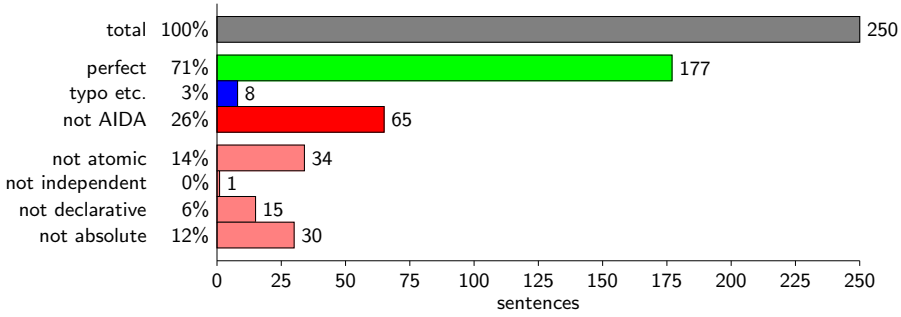


Fig. 5. Quality of the sentences extracted from the GeneRIF dataset

encouraged to write a separate sentence for each claim, and non-atomicity is difficult to detect with simple regular expressions.

To evaluate the sentence clustering, we looked at the clusters for the sentences from the user study. We know that the set of sentences from a particular task has a large internal overlap in terms of meaning (which does not mean that they are all similar, as the respective text may describe more than one claim). Our results show that sentences from the same task indeed end up in the same clusters. On average, 99.2% of the other objects that such a sentence encountered in its cluster were sentences from the same task. Furthermore, 84% of the sentences were connected to at least one other sentence from the same task. Below is an example of two sentences that our clustering algorithm successfully connected. They convey the same meaning, but are quite different on the surface level:

- Hepatic reticuloendothelial function is impaired to the same degree in cirrhotic patients with or without a previous history of SBP.
- History of spontaneous bacterial peritonitis does not affect impairment of hepatic reticuloendothelial function in cirrhotic patients.

For both evaluations on automatic processing (i.e. extraction and clustering), we applied very simple methods and we assume that we can achieve even better results with state-of-the-art NLP techniques, language resources, and ontologies.

6 Conclusions

The pace of modern science is such that it is very difficult to keep track of the latest research results. Our approach addresses this problem by allowing researchers to easily access and communicate research hypotheses, claims, and opinions within the existing nanopublication framework. Representing scientific claims as AIDA sentences makes the nanopublication concept much more flexible and significantly widens its practical applicability. Our results show that scientists are able to efficiently produce high-quality AIDA nanopublications, that it is feasible to extract such nanopublications from existing text resources, and that it is possible to cluster them by sentence similarity. Together, these findings suggest that our approach is practical, and that it may assist the nanopublication initiative to attain critical mass.

References

1. Carroll, J.J., Bizer, C., Hayes, P., Stickler, P.: Named graphs, provenance and trust. In: WWW 2005, pp. 613–622. ACM (2005)
2. Ciccarese, P., Wu, E., Wong, G., Ocana, M., Kinoshita, J., Ruttenberg, A., Clark, T.: The SWAN biomedical discourse ontology. Journal of Biomedical Informatics 41(5), 739–751 (2008)
3. Clark, T., Kinoshita, J.: Alzforum and SWAN: The present and future of scientific web communities. Briefings in Bioinformatics 8(3), 163–171 (2007)
4. de Waard, A., Schneider, J.: Formalising uncertainty: An ontology of reasoning, certainty and attribution (ORCA). In: SATBI+SWIM 2012 (2012)
5. Everitt, B.S.: Cluster Analysis, 3rd edn. Edward Arnold and Halsted Press (1993)
6. Groth, P., Gibson, A., Velterop, J.: The anatomy of a nano-publication. Information Services and Use 30(1), 51–56 (2010)
7. King, R.D., Rowland, J., Oliver, S.G., Young, M., Aubrey, W., Byrne, E., Liakata, M., Markham, M., Pir, P., Soldatova, L.N., Sparkes, A., Whelan, K.E., Clare, A.: The automation of science. Science 324(5923), 85–89 (2009)
8. Kuhn, T., Krauthammer, M.: Underspecified scientific claims in nanopublications. In: WoLE 2012, pp. 29–32. CEUR-WS (2012)
9. Kuhn, T., Royer, L., Fuchs, N.E., Schröder, M.: Improving text mining with controlled natural language: A case study for protein interactions. In: Leser, U., Naumann, F., Eckman, B. (eds.) DILS 2006. LNCS (LNBI), vol. 4075, pp. 66–81. Springer, Heidelberg (2006)
10. Mons, B., van Haagen, H., Chichester, C., den Dunnen, J.T., van Ommen, G., van Mulligen, E., Singh, B., Hooft, R., Roos, M., Hammond, J., Kiesel, B., Giardine, B., Velterop, J., Groth, P., Schultes, E.: The value of data. Nature Genetics 43(4), 281–283 (2011)
11. Rector, A.L., Drummond, N., Horridge, M., Rogers, J., Knublauch, H., Stevens, R., Wang, H., Wroe, C.: OWL pizzas: Practical experience of teaching OWL-DL: Common errors & common patterns. In: Motta, E., Shadbolt, N.R., Stutt, A., Gibbins, N. (eds.) EKAW 2004. LNCS (LNAI), vol. 3257, pp. 63–81. Springer, Heidelberg (2004)
12. Seringhaus, M.R., Gerstein, M.B.: Publishing perishing? Towards tomorrow's information architecture. BMC Bioinformatics 8(1), 17 (2007)
13. Soldatova, L.N., King, R.D.: An ontology of scientific experiments. Journal of the Royal Society Interface 3(11), 795–803 (2006)
14. Soldatova, L.N., Rzhetsky, A., King, R.D.: Representation of research hypotheses. J. Biomed Semantics 2(suppl. 2), S9 (2011)
15. Wyner, A., Angelov, K., Barzdins, G., Damljanovic, D., Davis, B., Fuchs, N., Hoefler, S., Jones, K., Kaljurand, K., Kuhn, T., Luts, M., Pool, J., Rosner, M., Schwitter, R., Sowa, J.: On controlled natural languages: Properties and prospects. In: Fuchs, N.E. (ed.) CNL 2009. LNCS, vol. 5972, pp. 281–289. Springer, Heidelberg (2010)

The Wisdom of the Audience: An Empirical Study of Social Semantics in Twitter Streams

Claudia Wagner[1], Philipp Singer[2], Lisa Posch[2], and Markus Strohmaier[2]

[1] JOANNEUM RESEARCH, IIS, 8010 Graz, Austria
claudia.wagner@joanneum.at
[2] Graz University of Technology, KTI, 8010 Graz, Austria
{philipp.singer,markus.strohmaier}@tugraz.at, lposch@sbox.tugraz.at

Abstract. Interpreting the meaning of a document represents a fundamental challenge for current semantic analysis methods. One interesting aspect mostly neglected by existing methods is that authors of a document usually assume certain background knowledge of their intended audience. Based on this knowledge, authors usually decide what to communicate and how to communicate it. Traditionally, this kind of knowledge has been elusive to semantic analysis methods. However, with the rise of social media such as Twitter, background knowledge of intended audiences (i.e., the community of potential readers) has become explicit to some extents, i.e., it can be modeled and estimated. In this paper, we (i) systematically compare different methods for estimating background knowledge of different audiences on Twitter and (ii) investigate to what extent the background knowledge of audiences is useful for interpreting the meaning of social media messages. We find that estimating the background knowledge of social media audiences may indeed be useful for interpreting the meaning of social media messages, but that its utility depends on manifested structural characteristics of message streams.

1 Introduction

To understand the meaning of social media documents and annotate them with ontological concepts or lightweight semantic annotations is a crucial problem since social media documents tend to be short, the language used tends to be informal and new topics may arise on social media which have not been covered anywhere else before. While existing semantic analysis methods can be used to understand and model the semantics of individual social media messages to some extent, one drawback of these methods is that they are limited to analyzing the content of the document without taking the social context into account. However, social media documents are created and published in a social environment where users communicate with imagined audience [6] [7]. As we know from communication theory, e.g., the Maxim of Quantity by Grice [2] or from Speech Act Theory [11], authors or speakers usually make their messages as informative as required but do not provide more information than necessary. This suggests that the background knowledge of an imagined audience for a given message may contribute to reveal the topics or concepts the message is about.

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 502–516, 2013.

This paper sets out to study this hypothesis. We use three datasets obtained from Twitter, a popular microblogging service. Since information consumption on Twitter is mainly driven by explicitly defined social networks, we approximate the imagined audience of a message using the social network of its author. In addition, we estimate the collective background knowledge of an audience by using the content published by the members of the audience. While the aim of this work is not to predict who will read a message, we want to approximate the collective background knowledge of a set of key audience users of a hashtag stream who are likely to be exposed to a message and might have the background knowledge to interpret it. We do that to assess the value of background knowledge for interpreting the semantics of microblog messages. More specifically, this work addresses following research questions:

RQ1: To What Extent Is the Background Knowledge of the Audience Useful for Guessing the Meaning of Social Media Messages?. To investigate this question, we conduct a classification experiment in which we aim to classify messages into hashtag categories. As shown in [5], hashtags can in part be considered as a manually constructed semantic grounding of microblog messages. We assume that an audience which can guess the hashtag of a given message more accurately can also interpret the meaning of the message more accurately. We will use messages authored by the audience of a stream for training the classifier and we will test the performance on actual messages of a stream.

RQ2: What Are the Characteristics of an Audience Which Possesses Useful Background Knowledge for Interpreting the Meaning of a Stream's Messages and Which Types of Streams Tend to have Useful Audiences?. To answer this question, we introduce several measures describing structural characteristics of an audience and its corresponding social stream. Then, we measure the correlation between these characteristics and the corresponding classification performance analyzed in RQ1. This shows the extent to which useful audiences can be identified based on structural characteristics.

The results of our experiments demonstrate that the background knowledge of a stream's audience is useful for the task of interpreting the meaning of microblog messages, but that the performance depends on structural characteristics of the audience and the underlying social stream. To our best knowledge, this is the first work which explores *to what extent* and *how* the background knowledge of an audience can be used to understand and model the semantics of individual microblog messages. Our work is relevant for researchers interested in learning semantic models from text and researchers interested in annotating social streams with lightweight semantics.

This paper is structured as follows: In Section 3 we give an overview about related research. Section 4 describes our experimental setup, including our methodology and a description of our datasets. Section 5 presents our experiments and empirical results. In Section 6 we discuss our results and conclude our work in Section 7.

2 Terminology

We define a *social stream* as a stream of data or content which is produced through users' activities conducted in an online social environment like Twitter where others see the manifestation of these activities. We assume that no explicitly defined rules for coordination in such environments exist. In this work we explore one special type of social streams, i.e., *hashtag streams*. A hashtag stream is a special type of a resource stream [13] and can be defined as a tuple consisting of users (U), messages (M), resources (R), a ternary relation (Y') and a function (ft). Specifically, it is defined as $S(R') = (U, M, R, Y', ft)$, where $Y' = \{(u, m, r) \,|\, r \in R' \vee \exists r' \in R', \tilde{m} \in M, u \in U : (u, \tilde{m}, r') \in Y\}$ and $R' \subseteq R$ and $Y' \subseteq Y$. In words, a hashtag stream consists of all messages containing one or several specific hashtags $r' \in R'$ and all resources (e.g., other hashtags, URLs or keywords) and users related to these messages.

In social online environments, information consumption is driven by explicitly defined social networks and therefore we can estimate the *audience* of a social stream by analyzing the incoming and outgoing links of the authors who created the stream. We call a user U_1 a *follower* of user U_2 if U_1 has established a unidirectional link with U_2 (in contrast user U_2 is a *followee* of user U_1), while we call a user U_3 a *friend* of user U_1 if U_1 has established a link with U_3 and vice versa. In this work, we assume that the union of the friends of all authors of a given hashtag constitute a hashtag stream's *audience*.

3 Related Work

Understanding and modeling the semantics of individual messages is important in order to support users in consuming social streams efficiently – e.g., via filtering social streams by users' interests or recommending tweets to users. However, one drawback of many state-of-the-art text mining approaches (such as *Bag of Words*) is that they suffer from the sparsity of microblog messages (i.e., the limited length of messages). Hence, researchers got interested in exploring those limitations and develop methods for overcoming them. Two commonly used strategies for improving short text classification are: (a) improving the classifier or feature representation and (b) using background knowledge for enriching sparse textual data.

Improving the Classifier or Feature Representation: Sriram et al. [12] present a comparison of different text mining methods applied on individual Twitter messages. Similar to our work, they use a message classification task to evaluate the quality of the outcome of each text mining approach. Limitations of their work are that they only use five broad categories (news, opinions, deals, events and private message) in which they classify tweets. Further, they perform their experiments on a very small set of tweets (only 5,407 tweets) which were manually assigned to the aforementioned categories. Their results show that the authorship plays a crucial role since authors generally adhere to a specific tweeting pattern i.e., a majority of tweets from the same author tend to be within

a limited set of categories. However, their authorship feature requires that tweets of the same authors occur in the trainings and test dataset.

Latent semantic models such as topic models provide a method to overcome data sparsity by introducing a latent semantic layer on top of individual documents. Hong et al. [3] compare the quality and effectiveness of different standard topic models in the context of social streams and examine different training strategies. To assess the quality and effectiveness of different topic models and training strategies the authors use them in two classification tasks: a user and message classification task. Their results show that the overall accuracy for classifying messages into 16 general Twitter suggest categories (e.g., Health, Food&Drinks, Books) when using topics as features is almost twice as accurate as raw TF-IDF features. Further their results suggest that the best performance can be achieved by training a topic model on aggregated messages per user. One drawback of their work is that they only use 274 users from 16 selected Twitter suggest directories[1], who may be very popular users that may be likely to mainly post messages about the assigned topic.

Enriching Sparse Textual Data with Background Knowledge: In [9] the authors present a general framework to build classifiers for short and sparse text data by using hidden topics discovered from huge text and Web collections. Their empirical results show that exploiting those hidden topics improves the accuracy significantly within two tasks: "Web search domain disambiguation" and "disease categorization for medical text". Hotho et al. [4] present an extensive study on the usage of background knowledge from WordNet for enriching documents and show that most enrichment strategies can indeed improve the document clustering accuracy. However, it is unclear if their results generalize to the social media domain since the vocabulary mismatch between WordNet and Twitter might be bigger than between WordNet and news articles.

Summary: Recent research has shown promising steps towards improving short text classification by enhancing classifiers and feature representation or by using general background knowledge from external sources to expand sparse textual data. However - to the best of our knowledge - using the background knowledge of imagined audiences to interpret the meaning of social media messages represents a novel approach that has not been studied before. The general usefulness of such an approach is thus unknown.

4 Experimental Setup

The aim of our experiments is to explore different approaches for modeling and understanding the semantics or the main theme of microblog messages using different kinds of background knowledge. Since the audience of a microblog message are the users who are most likely to interpret (or to be able to interpret) the message, we hypothesize that incorporating the background knowledge of the audience of such messages helps to better understand what a single message is about. In the following we describe our datasets and methodology.

[1] `http://twitter.com/invitations/suggestions`

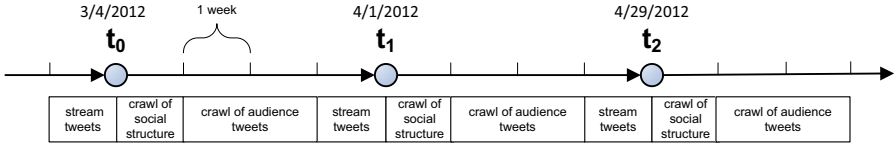


Fig. 1. Timeline of the crawling process

4.1 Datasets

In this work we use three Twitter datasets each consisting of a temporal snapshot of the selected hashtag streams, the social network of a stream's authors, their follower and followees and the tweets authored by the selected followers and followees (see Figure 1). We generate a diverse sample of hashtag streams as follows: In [10] the authors created a classification of frequently used Twitter hashtags by category, identifying eight broad categories (see Table 1). We decided to reuse these categories and sample from each category 10 hashtags. We bias our random sample towards active hashtag streams by re-sampling hashtags for which we found less than 1,000 messages when crawling (4. March 2012). For those categories for which we could not find 10 hashtags which had more than 1,000 messages (games and celebrity) we select the most active hashtags per category. Since two hashtags #bsb and #mj appeared in the sample of two different categories, we ended up having a sample of 78 different hashtags (see Table 1).

Table 1. Randomly selected hashtags per category (ordered alphabetically)

technology	idioms	sports	political	games	music	celebrity	movies
blackberry	factaboutme	f1	climate	e3	bsb	ashleytisdale	avatar
ebay	followfriday	football	gaza	games	eurovision	brazilmissesdemi	bbcqt
facebook	dontyouhate	golf	healthcare	gaming	lastfm	bsb	bones
flickr	iloveitwhen	nascar	iran	mafiawars	listeningto	michaeljackson	chuck
google	iwish	nba	mmot	mobsterworld	mj	mj	glee
iphone	nevertrust	nhl	noh8	mw2	music	niley	glennbeck
microsoft	omgfacts	redsox	obama	ps3	musicmonday	regis	movies
photoshop	oneofmyfollowers	soccer	politics	spymaster	nowplaying	teamtaylor	supernatural
socialmedia	rememberwhen	sports	teaparty	uncharted2	paramore	tilatequila	tv
twitter	wheniwaslittle	yankees	tehran	wow	snsd	weloveyoumiley	xfactor

Each dataset corresponds to one timeframe. The starting dates of the timeframes are March 4th (t_0), April 1st (t_1) and April 29th, 2012 (t_2). We crawled the most recent English tweets for each hashtag of our selection using Twitter's public search API on the first day of each timeframe and retrieved tweets that were authored within the last week. During the first week of each timeframe the user IDs of the followers and followees of streams's authors were crawled. Finally, we also crawled the most recent 3,200 tweets (or less if less were available) of all users who belong either to the top hundred authors or audience users of each hashtag stream. We ranked authors by the number of tweets they contributed to the stream and ranked audience users by the number of a stream's authors with whom they have established a bidirectional follow relation. Figure 1 illustrates

Table 2. Description of the datasets

	t_0	t_1	t_2
Stream Tweets	94,634	94,984	95,105
Audience Tweets	29,144,641	29,126,487	28,513,876
Stream Authors	53,593	54,099	53,750
Followers	56,685,755	58,822,119	66,450,378
Followees	34,025,961	34,263,129	37,674,363
Friends	21,696,134	21,914,947	24,449,705
Mean Followers per Author	1,057.71	1,087.31	1,236.29
Mean Followees per Author	634.90	633.34	700.92
Mean Friends per Author	404.83	405.09	454.88

this process. Table 2 depicts the number of tweets and relations between users that we crawled during each timeframe.

4.2 Modeling Twitter Audiences and Background Knowledge

Audience Selection. Since the audience of a stream is potentially very large, we rank members of the audience according to the number of authors per stream an audience user is friend with. This allows us to determine key audience members per hashtag stream (see Figure 2). We experimented with different thresholds (i.e., we used the top 10, 50 and top 100 friends) and got similar results. In the remainder of the paper, we only report the results for the best thresholds (c.f., Table 3).

Background Knowledge Estimation. Beside selecting an audience of a stream, we also needed to estimate their knowledge. Hence, we compared four different methods for estimating the knowledge of a stream's audience:

- The first method (*recent*) assumes that the background knowledge of an audience can be estimated from the most recent messages authored by the audience users of a stream.
- The second method (*top links*) assumes that the background knowledge of the audience can be estimated from the messages authored by the audience which contain one of the top links of that audience – i.e., the links which were recently published by most audience-users of that stream. Since messages including links tend to contain only few words due to the character limitations of Twitter messages (140 characters), we test two variants of this method: (1) we represented the knowledge of the audience via the plain messages which contain one of the top links (*top links plain*) and (2) (*top links enriched*) we resolved the links and enriched the messages with keywords and title information derived from meta-tags of html pages links are pointing to.
- Finally, the last method (*top tags*) assumes that the knowledge of the audience can be estimated via the messages authored by the audience which contain one of the top hashtags of that audience – i.e., the hashtags which were recently used by most audience users of that stream.

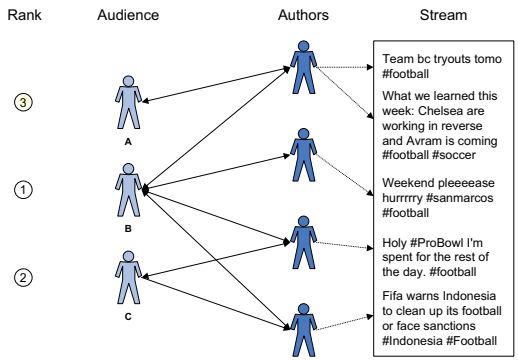


Fig. 2. To estimate the audience of a hashtag stream, we rank the friends of the stream's authors by the number of authors they are related with. In this example, the hashtag stream #football has four authors. User B is a friend of all four authors of the stream and is therefore most likely to be exposed to the messages of the stream and to be able to interpret them. Consequently, user B receives the highest rank. User C is a friend of two authors and receives the second highest rank. The user with the lowest rank (user A) is only the friend of one author of the stream.

4.3 Methods

In this section we present the text mining methods we used to extract content features from raw text messages. In a preprocessing step we removed all English stopwords, URLs and Twitter usernames from the content of our microblog messages. We also removed Twitter syntax such as *RT* or *via*. For stemming we used Porter Stemming. In the following part of this section we describe the text mining methods we used for producing semantic annotations of microblog messages.

Bag-of-Words Model. Vector-based methods allow us to represent each microblog message as a vector of terms. Different methods exist to weight these terms – e.g., term frequency (*TF*), inverse document frequency (*IDF*) and term frequency-inverse document frequency (*TF-IDF*). We have used different weighting approaches and have achieved the best results by using TF-IDF. Therefore, we only report results obtained from the TF-IDF weighting schema in this paper.

Topic Models. Topic models are a powerful suite of algorithms which allow discovering the hidden semantic structure in large collection of documents. The idea behind topic models is to model documents as arising from multiple topics, where each document has to favor few topics. Therefore, each document exhibits different topic proportions and each topic is defined as a distribution over a fixed vocabulary of terms, where few words are favored.

The most basic topic modeling algorithm is Latent Dirichlet Allocation (LDA) [1]. In our experiments we used MALLET's [8] LDA implementation and fitted an LDA model to our tweet corpus using individual tweets as trainings document. We chose the default hyperparameters ($\alpha = 50/T$, $\beta = 0.01$) and optimized them during training by using Wallach's fixed point iteration method [14]. We chose the number of topics T= 500 empirically by estimating the log likelihood of a model with T= 300, 500 and 700 on held out data. Given enough iterations (we used 2000) the Markov chain (which consists of topic assignments z for each token in the training corpus) has potentially converged and we can get estimates of the word distribution of topics ($\hat{\phi}$) and the topic distribution of documents ($\hat{\theta}$) by drawing samples from the chain. The estimated distributions $\hat{\phi}$ and $\hat{\theta}$ are predictive distributions and are later used to infer the topics of social stream messages.

4.4 Message Classification Task

To evaluate the quality and utility of audience's background knowledge for interpreting the meaning of microblog message, we conducted a message classification task using hashtags as classes (i.e., we had a multi-class classification problem with 78 classes). We assume that an audience which is better in guessing the hashtag of a Twitter message is better in interpreting the meaning of the message. For each hashtag stream, we created a baseline by picking the audience of another stream at random and compared the performance of the random audience with the real stream's audience. Our baseline tests how well a randomly selected audience can interpret the meaning of a stream's messages. One needs to note that a simple random guesser baseline would be a weaker baseline than the one described above and would lead to a performance of 1/78.

We extracted content features (via the aforementioned methods) from messages authored by the audience of a stream before t_1 and used them to train a classifier. That means messages of the audience of a stream were used as training samples to learn a semantic representation of messages in each hashtag class. We tested the performance of the classifier on actual messages of a stream which were published after t_1. In following such an approach, we ensured that our classifier does not benefit from any future information (e.g., messages published in the future or social relations which were created in the future). Out of several classification algorithms applicable for text classification such as Logistic Regression, Stochastic Gradient Descent, Multinomial Naive Bayes or Linear SVC, we could achieve the best results using a Linear SVC[2]. As evaluation metric we chose the weighted average *F1-score*.

4.5 Structural Stream Measures

To assess the association between structural characteristics of a social stream and the usefulness of its audience (see RQ2), we introduce the following measures

[2] `http://www.csie.ntu.edu.tw/~cjlin/liblinear/`

which describe structural aspects of those streams. We differ between static measures which only use information from one time point and dynamic measures which combine information from several time points.

Static Measures

- **Coverage Measures:** The coverage measures characterize a hashtag stream via the nature of its messages. For example the *informational coverage* measure indicates how many messages of a stream have an informational purpose - i.e., contain a link. The *conversational coverage* measures the mean number of messages of a stream that have a conversational purpose - i.e., those messages that are directed to one or several specific users. The *retweet coverage* measures the percentage of messages which are retweets. The *hashtag coverage* measures the mean number of hashtags per message in a stream.
- **Entropy Measures:** We use normalized entropy measures to capture the randomness of a stream's authors and their followers, followees and friends. We rank for each hashtag stream the authors by the number of tweets they authored and the followers, followees and friends by the number of authors they are related with. A high *author entropy* indicates that the stream is created in a democratic way since all authors contribute equally much. A high *follower entropy* and *friend entropy* indicate that the followers and friends do not focus their attention towards few authors but distribute it equally across all authors. A high *followee entropy* and *friend entropy* indicate that the authors do not focus their attention on a selected part of their audience.
- **Overlap Measures:** The overlap measures describe the overlap between the authors and the followers (*Author-Follower Overlap*), followees (*Author-Followee Overlap*) or friends (*Author-Friend Overlap*) of a hashtag stream. If these overlaps are one, the stream is consumed and produced by the same users who are interconnected. A high overlap suggests that the community around the hashtag is rather closed, while a low overlap indicates that the community is more open and that the active and passive part of the community do not extensively overlap.

Dynamic Measures. To explore how the social structure of a hashtag stream changes over time we measure the distance between the tweet-frequency distributions of a stream's authors at different time points and the author-frequency distributions of a stream's followers, followees or friends at different time points. We use a symmetric version of the *Kullback-Leibler (D_{KL}) divergence* which represents a natural distance measure between two probability distributions (A and B) and is defined as follows: $\frac{1}{2}D_{KL}(A||B) + \frac{1}{2}D_{KL}(B||A)$. The KL divergence is *zero* if the two distributions A and B are identical and approaches infinity as they differ more and more. We measure the KL divergence for the distributions of authors, followers, followees and friends.

5 Experiments

The aim of our experiments is to explore different methods for modeling and understanding the semantics of Twitter messages using background knowledge of different kinds of audiences. Due to space restrictions we only report results obtained when training our model on the dataset t_0 and testing it on the dataset t_1. We got comparable results when training on the dataset t_1 and testing on dataset t_2.

5.1 RQ1: To What Extent Is the Background Knowledge of the Audience Useful for Guessing the Meaning of Social Media Messages?

To answer this question we compared the performance of a classification model using messages authored by the audience of a stream (i.e., the top friends of a hashtag stream's authors) as training samples with the performance of a classification model using messages of a randomly selected audience (a baseline, i.e. the top friends of the authors of a randomly selected hashtag stream) as training samples. If the audience of a stream does not possess more knowledge about the semantics of the stream's messages than a randomly selected baseline audience, the results from both classification models should not differ significantly.

Our results show that all classifiers trained on messages authored by the audience of a hashtag stream clearly outperform a classifier trained on messages authored by a randomly selected audience. This indicates that the messages authored by the audience of a hashtag stream indeed contain important information. Our results also show that a TF-IDF based feature representation slightly outperforms a topical feature representation.

The comparison of the four different background knowledge estimation methods (see Section 4.2) shows that the best results can be achieved when using the most recent messages authored by the top 10 audience users and when using messages authored by the top 100 audience users containing one of the top

Table 3. Average weighted F1-Scores of different classification models trained on data crawled at t_0 and tested on data crawled at t_1. We either used words weighted via TF-IDF or topics inferred via LDA as features for a message. The table shows that all audience-based classification models outperformed a random baseline. For the random baseline, we randomly swapped audiences and hashtag streams. A classifier trained on the most recent messages of the top 10 friends of a hashtag stream yields the best performance.

Classification Model	F1 (TF-IDF)	F1 (LDA)
Baseline (Random audience: top 10 friends, Messages: recent)	**0.01**	**0.01**
Audience: top 10 friends, Messages: recent	**0.25**	**0.23**
Audience: top 100 users, Messages: top links enriched	0.13	0.10
Audience: top 100 users, Message selection: top links plain	0.12	0.10
Audience: top 100 users, Message selection: top tags	**0.24**	**0.21**

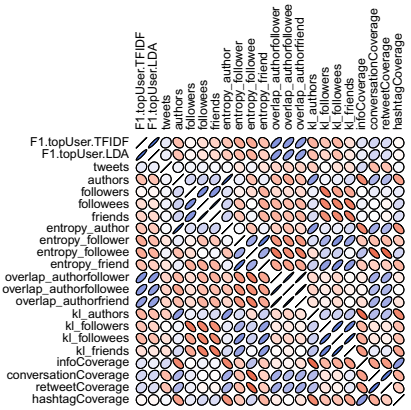
hashtags of the audience (see Table 3). Tweets containing one of the top links of the audience (no matter if enriched or not) are less useful than messages containing one of the top hashtags of the audience. Surprisingly, our message link enrichment strategies did not show a large boost in performance. A manual inspection of a small sample of links showed that the top links of an audience often point to multimedia sharing sites such as youtube[3], instagr.am[4] or twitpic[5]. Unfortunately, title and keywords which can be extracted from the meta information of those sites often contain information which is not descriptive.

To gain further insights into the usefulness of an audience's background knowledge, we compared the average weighted F1-Score of the eight hashtag categories from which our hashtags were initially drawn (see Table 4). Our results show that for certain categories such as sports and politics the knowledge of the audience clearly helps to learn the semantics of hashtag streams' messages, while for other streams – such as those belonging to the categories celebrities and idioms – background knowledge of the audience seems to be less useful. This suggests that only certain types of social streams are amenable to the idea of exploiting the background knowledge of stream audiences. Our intuition is that audiences of streams that are about fast-changing topics are *less useful*. We think that these audiences are only loosely associated to the topics of the stream, and therefore their background knowledge does not add much to a semantic analysis task. Analogously, we hypothesize audiences of streams that are narrow and stable are *more useful*. It seems that a community of tightly knit users is built around a topic and a common knowledge is developed over time. This seems to provide useful background knowledge to a semantic analysis task. Next, we want to understand the characteristics that distinguish audiences that are useful from audiences that are less useful.

Table 4. Average weighted F1-Score per category of the best audience-based classifier using recent messages (represented via TF-IDF weighted words or topic proportions) authored by the top ten audience users of a hashtag stream. We got the most accurate classification results for the category *sports* and the least accurate classification results for the category *idioms*.

		TFIDF		LDA	
category	**support**	**F1**	**variance**	**F1**	**variance**
celebrity	4384	0.17	0.08	0.15	0.16
games	6858	0.25	0.33	0.22	0.31
idioms	14562	**0.09**	0.14	**0.05**	0.05
movies	14482	0.22	0.19	0.18	0.18
music	13734	0.23	0.25	0.18	0.26
political	13200	0.36	0.22	0.33	0.21
sports	13960	**0.45**	0.19	**0.42**	0.21
technology	13878	0.22	0.20	0.22	0.2

[3] http://www.youtube.com
[4] http://instagram.com/
[5] http://twitpic.com/



(a)

feature	cor with F1 (TF-IDF)	cor with F1 (LDA)
overlap_authorfollower	0.675	0.655
overlap_authorfollowee	0.642	0.628
overlap_authorfriend	0.612	0.602
conversation_coverage	0.256	0.256
kl_followers	-0.281	–
kl_followees	-0.343	-0.302
kl_authors	-0.359	-0.307
entropy_author	-0.270	-0.400
entropy_friend	-0.307	–
entropy_follower	-0.400	-0.319
entropy_followee	-0.401	-0.368

(b)

Fig. 3. This matrix shows the Spearman rank correlation strength between structural stream properties and F1-Scores of two audience-based classification models averaged across all categories. The color and form of the ellipse indicate the correlation strength. Red means negative and blue means positive correlation. The rounder the ellipse the lower the correlation. The inspection of the first two columns of the correlation matrix reveals that several structural measures are correlated with the F1-Scores and Figure 3b shows which of those are indeed statistical significant.

5.2 RQ2: What Are the Characteristics of an Audience Which Possesses Useful Knowledge for Interpreting the Meaning of a Stream's Messages and Which Types of Streams Tend to Have Useful Audiences?

To understand whether the structure of a stream has an effect on the usefulness of its audience for interpreting the meaning of its messages, we perform a correlation analysis and investigate to what extent the ability of an audience to interpret the meaning of messages correlates with structural stream properties. We use the F1-scores of the best audience based classifiers (using TFIDF and LDA) as a proxy measure for the audience's ability to interpret the meaning of a stream's messages.

Figure 3a shows the strength of correlation between the F1-scores and the structural properties of streams across all categories. An inspection of the first two columns of the correlation matrix reveals interesting correlations between structural stream properties and the F1-scores of the audience-based classifiers. We further report all significant *Spearman rank correlation coefficients* ($p < 0.05$) across all categories in Figure 3b.

Figure 3a and Figure 3b show that across all categories, the measures which capture the overlap between the authors and the followers, friends and followees show the highest positive correlation with the F1-scores. That means, the higher the overlap between authors of a stream and the followers, friends and followees of the stream, the better an audience-based classifier performs. This is not

surprising since it indicates that the audience which is best in interpreting stream messages is an active audience, which also contributes to the creation of the stream itself (high author friend overlap). Further, our results suggest that the audience of a stream possesses useful knowledge for interpreting a stream's messages if the authors of a stream follow each other (high author follower and author followee overlap). This means that the stream is produced and consumed by a community of users who are tightly interconnected. The only significant coverage measure is the conversational coverage measure. It indicates that the audiences of conversational streams are better in interpreting the meaning of a stream's messages. This suggests that it is not only important that a community exists around a stream, but also that the community is communicative.

All entropy measures show significant negative correlations with the F1-Scores. This shows that the more focused the author-, follower-, followee- and/or friend-distribution of a stream is (i.e., lower entropy), the higher the F1-Scores of an audience-based classification model are. The entropy measures the randomness of a random variable. For example, the author-entropy describes how random the tweeting process in a hashtag stream is – i.e., how well one can predict who will author the next message. The friend-entropy describes how random the friends of hashtag stream's authors are – i.e., how well one can predict who will be a friend of most hashtag stream's authors. Our results suggest that streams tend to have a better audience if their authors and author's followers, followees and friends are less random.

Finally, the KL divergences of the author-, follower-, and followee-distributions show a significant negative correlation with the F1-Scores. This indicates that the more stable the author, follower and followee distribution is over time, the better the audience of a stream is. If for example the followee distribution of a stream changes heavily over time, authors are shifting their social focus. If the author distribution of a stream has a high KL divergence, this indicates that the set of authors of a stream are changing over time.

In summary, our results suggest that *streams which have a useful audience tend to be created and consumed by a stable and communicative community* – i.e., a group of users who are interconnected and have few core users to whom almost everyone is connected.

6 Discussion of Results

The results of this work show that messages authored by the audience of a hashtag stream indeed represent background knowledge that can help interpreting the meaning of streams' messages. We showed that the usefulness of an audience's background knowledge depends on the applied content selection strategies (i.e., how the potential background knowledge of an audience is estimated). However, since the audience of a hashtag stream is potentially very large, picking the right threshold for selecting the best subset of the audience is an issue. In our experiments we empirically picked the best threshold but did not conduct extensive experiments on this issue. Surprisingly, more sophisticated content selection

strategies such as top links or top hashtags were only as good or even worse than the simplest strategy which used the most recent messages (up to 3,200) of each top audience user.

Our work shows that not all streams exhibit audiences which possess knowledge useful for interpreting the meaning of a stream's messages (e.g., streams in certain categories like celebrities or especially idioms). Our work suggests that the utility of a stream's audience is significantly associated with structural characteristics of the stream.

Finally, our work has certain limitations. Recent research on users' hashtagging behavior [15] suggests that hashtags are not only used as topical or context marker of messages but can also be used as a symbol of community membership. In this work, we have mostly neglected the social function of hashtags. Although the content of a message may not be the only factor which influences which hashtag a user choses, we assume a "better" semantic model might be able to predict hashtags more accurately.

7 Conclusions and Future Work

This work explored whether the background knowledge of Twitter audiences can help in identifying the meaning of social media messages. We introduced different approaches for estimating the background knowledge of a stream's audience and presented empirical results on the usefulness of this background knowledge for interpreting the meaning of social media documents.

The main findings of our work are:

- The audience of a social stream possesses knowledge which may indeed help to interpret the meaning of a stream's messages.
- The audience of a social stream is most useful for interpreting the meaning of a stream's messages if the stream is created and consumed by a stable and communicative community – i.e., a group of users who are interconnected and have few core users to whom almost everyone is connected.

In our future work we want to explore further methods for estimating the potential background knowledge of an audience (e.g., using user lists or bio information rather than tweets). Furthermore, we want to compare our method directly to the proposed research in [4] and [9]. Combining latent and explicit semantic methods for estimating audience's background knowledge and exploiting it for interpreting the main theme of social media messages are promising avenues for future research.

Acknowledgments. This work was supported in part by a DOC-fForte fellowship of the Austrian Academy of Science to Claudia Wagner and by the FWF Austrian Science Fund Grant I677 and the Know-Center Graz.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. 3, 993–1022 (2003)
2. Paul Grice, H.: Logic and conversation. In: Cole, P. (ed.) Speech Acts. Syntax and semantics, vol. 3, pp. 41–58. Academic Press, New York (1975)
3. Hong, L., Davison, B.D.: Empirical study of topic modeling in twitter. In: Proceedings of the IGKDD Workshop on Social Media Analytics (SOMA) (2010)
4. Hotho, A., Staab, S., Stumme, G.: Wordnet improves text document clustering. In: Proc. of the SIGIR 2003 Semantic Web Workshop, pp. 541–544 (2003)
5. Laniado, D., Mika, P.: Making sense of twitter. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010, Part I. LNCS, vol. 6496, pp. 470–485. Springer, Heidelberg (2010)
6. Litt, E.: Knock, knock. Who's there? The imagined audience. Journal of Broadcasting and Electronic Media 56 (2012)
7. Marwick, A., Boyd, D.: I tweet honestly, i tweet passionately: Twitter users, context collapse, and the imagined audience. New Media and Society (2010)
8. McCallum, A.K.: Mallet: A machine learning for language toolkit (2002), `http://mallet.cs.umass.edu`
9. Phan, X.-H., Nguyen, L.-M., Horiguchi, S.: Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In: Proceedings of the 17th International Conference on World Wide Web, WWW 2008, pp. 91–100. ACM, New York (2008)
10. Romero, D.M., Meeder, B., Kleinberg, J.: Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter. In: Proceedings of the 20th International Conference on World Wide Web, WWW 2011, pp. 695–704. ACM, New York (2011)
11. Searle, J.: A taxonomy of illocutionary acts, pp. 334–369. University of Minnesota Press, Minneapolis (1975)
12. Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H., Demirbas, M.: Short text classification in twitter to improve information filtering. In: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, pp. 841–842. ACM, New York (2010)
13. Wagner, C., Strohmaier, M.: The wisdom in tweetonomies: Acquiring latent conceptual structures from social awareness streams. In: Semantic Search Workshop at WWW 2010 (2010)
14. Wallach, H.M.: Structured Topic Models for Language. PhD thesis, University of Cambridge (2008)
15. Yang, L., Sun, T., Zhang, M., Mei, Q.: We know what @you #tag: does the dual role affect hashtag adoption? In: Proceedings of the 21st International Conference on World Wide Web, WWW 2012, pp. 261–270. ACM, New York (2012)

Collecting Links between Entities Ranked by Human Association Strengths

Jörn Hees[1,2], Mohamed Khamis[3], Ralf Biedert[2,4],
Slim Abdennadher[3], and Andreas Dengel[1,2]

[1] Computer Science Department, University of Kaiserslautern, Germany
[2] Knowledge Management Department, DFKI GmbH, Kaiserslautern, Germany
{joern.hees,andreas.dengel}@dfki.de
[3] Computer Science & Engineering Department, German University in Cairo, Egypt
{mohammed.khamis,slim.abdennadher}@guc.edu.eg
[4] Tobii Technology AB, Stockholm, Sweden
rb@xr.io

Abstract. In recent years, the ongoing adoption of Semantic Web technologies has lead to a large amount of Linked Data that has been generated. While in the early days of the Semantic Web we were fighting data scarcity, nowadays we suffer from an overflow of information. In many situations we want to restrict the amount of facts which is shown to an end-user or passed on to another system to just the most important ones.

In this paper we propose to rank facts in accordance to human association strengths between concepts. In order to collect a ground truth we developed a Family Feud like web-game called "Knowledge Test Game". Given a Linked Data entity it collects other associated Linked Data entities from its players. We explain the game's concept, its suggestion box which maps the players' text input back to Linked Data entities and include a detailed evaluation of the game showing promising results. The collected data is published and can be used to evaluate algorithms which rank facts.

1 Introduction

Since its introduction in 2001 the Semantic Web [1] has gained much attention. While in the early days of the Semantic Web only few large, interlinked and publicly accessible RDF datasets were available, especially the Linking Open Data (LOD) project has changed this situation over the last years, generating one of the world's largest, decentralized knowledge bases [2]. Extracted from Wikipedia, DBpedia [3] is the most central of these datasets as it provides information about entities from a large variety of domains, provides URIs for these entities and thereby provides a bridge between many other domain specific datasets in Linked Data.[1]

[1] Also see http://lod-cloud.net/ the Linking Open Data cloud diagram by Richard Cyganiak and Anja Jentzsch.

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 517–531, 2013.

Despite being a huge success for the Semantic Web, the increasing amount of available Linked Data creates new problems. While in the beginning there was not nearly enough data available to answer simple real-world queries, nowadays it often is easier to answer very specific queries. Simple queries lack specificity and it is not rare that they return thousands of facts. Widely known examples of such queries are SPARQL's DESCRIBE queries. For a given concept `:c` of interest on many SPARQL endpoints a DESCRIBE just returns the union of all outgoing `{ :c ?p ?o. }` and incoming `{ ?s ?p :c. }` triples. The same holds true for the majority of resolvable URIs. Sometimes, the often alphabetically sorted results are even truncated without any sanity to reduce bandwidth consumption.

While this behavior is acceptable for debugging, it most certainly is not what should be happening in productive systems which try to use the gathered information and in the end present the results to users. When simply asked about a URI, servers should return useful information opposed to all information they know, as mentioned in the Linked Data Design Issues by Berners-Lee [4].

The problem with this rule is that it is unclear which information is useful for a client. It depends on the context of the client. Nevertheless, we can observe that clients who are in a specific context typically have a specific information need and are able to formulate more specific SPARQL queries than DESCRIBE or resolving URIs. Hence, in this paper we focus on a general purpose information need, as often encountered in search engines.

As human associations play a key role in human thinking, leading us from one thought to the next, we propose to rank Linked Data facts according to human association strengths between entities. This means that for an entity such as `dbpedia:Steve_Jobs` which is strongly associated to `dbpedia:Apple_Inc.` we will rank facts between these two entities higher than facts connecting `dbpedia:Steve_Jobs` and `dbpedia:Toy_Story` entities.

Note that associations should be distinguished from semantic similarity. Two entities can be associated (see above), semantically similar (`dbpedia:Steve_Jobs`, `dbpedia:Brin_Sergey`), or both (`dbpedia:IPhone`, `dbpedia:IPad`).

To the best of our knowledge, currently no heuristic for or dataset of human association strengths between Linked Data entities is available. Furthermore, collecting such a dataset is prone to subjectivity, it is extremely monotonous and tedious, and the immense amount of Linked Data would cause great expenses if it was collected with a traditional experiment with paid participants.

In this paper we present a web-game called "Knowledge Test Game" to overcome the aforementioned problems, following the "Games With A Purpose (GWAP)" approach by von Ahn and Dabbish [5]. For a given Linked Data entity the game collects other associated Linked Data entities by outsourcing the problem to its players. The game is not intended to collect and rank associations for all Linked Data entities. Rather it is intended to build a ground truth that can be used to benchmark existing or new ranking techniques for Linked Data. As a next step, well performing ranking techniques could then be used to streamline the acquisition of associations between Linked Data entities, possibly allowing for a more human like exchange of knowledge between machines in the future.

The remainder of this paper is structured as follows. In Section 2 we list related GWAPs. In Section 3 we explain the game's concept, its suggestion box which maps the players' text input back to Linked Data entities, before presenting a detailed evaluation of the game showing promising results in Section 4. The results of this evaluation are discussed in Section 5 before our conclusion and future work in Section 6.

2 Related Work

While many approaches to rank Linked Data exist [6], we are not aware of any approach to collect or approximate human association strengths between Linked Data entities which also distinguishes them from semantic similarities. Hence, we will mainly focus on GWAPs which are related to our "Knowledge Test Game" in this section.

In terms of game design, the Knowledge Test Game is an output-agreement game [5] and a game with a purpose for the Semantic Web [7]. Its gaming principles are influenced by *Common Consensus*, another Family Feud like web-game which asks its players to name common sense goals (e.g., "What can you do to watch TV?"). In contrast to Common Consensus our approach focuses on all associations and does not only collect textual player inputs, but also maps the entered answers back to existing Linked Data entities with its suggestion-box.

The Knowledge Test Game can be seen as a successor of *Associator* [8] which was a pair-game to collect free-text associations for given topics. Associator as Common Consensus did not attempt to match the entered strings back to Linked Data entities during play time.

Other GWAPs to rate Linked Data exist. *BetterRelations* [9], a pair game asks its player which of two facts they consider more important. Aside from not collecting free associations between entities, BetterRelatons suffers from noise issues that our approach overcomes by using its suggestions-box.

WhoKnows? [10], a single player game, judges whether an existing Linked Data triple is known by testing players with (amongst others) a multiple choice test or a hangman game. In contrast to our approach, WhoKnows restricts itself to a limited fraction of the DBpedia dataset and excludes triples not matched by a predefined domain ontology in a preprocessing step. Similarly, *RISQ!* [11], a Jeopardy like single player game that generates questions from DBpedia, restricts itself to the domain of people after excluding non-sense facts in a preprocessing step. It then rates the remaining facts by using predefined templates to generate questions (clues) about subjects and tests if they are correctly recognized from a list of alternatives. This greatly reduces noise issues, but eliminates the possibility to collect user feedback about triple qualities and problems in the extraction process. Furthermore, unlike in the three aforementioned games, players of the Knowledge Test Game are not limited in their choices to previously existing connections of Linked Data entities, but instead can freely associate between them and even introduce new entities, should they be missing.

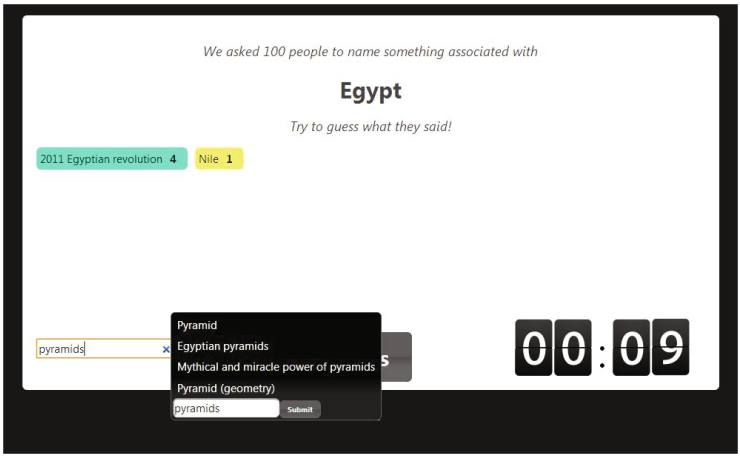


Fig. 1. A player has submitted 2 associations already for the topic "Egypt" (scored 4 points for the first one and 1 point for the other), and is now viewing the suggestions after guessing "pyramids"

3 The Knowledge Test Game

The Knowledge Test Game is a Game With A Purpose (GWAP), aiming at collecting and ranking associations. Players provide associations to Linked Data entities, whereas the associations themselves are Linked Data entities as well. The game is available on `http://www.knowledgetestgame.org` and through Facebook on `http://apps.facebook.com/knowledgetestgame/`.

3.1 Walkthrough

Each round of the Knowledge Test Game is referred to as a game instance, or simply a *game*. Each game has 2 to 10 players, all seeing the same *topic*, which is a Linked Data entity for which we would like to collect associations. Upon visiting the Knowledge Test Game homepage, players can choose to directly play a game or go through the *How to play* interactive tutorial. Furthermore, the players are able to authenticate themselves by logging in using their Google or Facebook accounts, or they can play anonymously as guests.

Joining a Game. When a player chooses to join a game, he either directly joins a random running game or creates a new one. A player can only join games that have less than 10 players, and have not been running for more than 70 % of its time. Additionally, the topic of the game being joined must be suitable according to the topic restrictions for that player (see Section 3.3).

Within a Game. Once a player joins a game (see Figure 1), he is presented with the statement: "We asked 100 people to name something associated with **Egypt** try to guess what they said!", where "Egypt" is the current game's topic. The mention of 100 people is a white lie in order to remind of the well known Family Feud TV show. This form of the question communicates to players that subjectivity should be avoided.

In a duration of 45 seconds, shown by a timer, the players are able to submit guesses resembling what they think is associated to the topic. For each submitted *guess*, the player gets a list of suggestions from which he can select the one most relevant to what he had in mind. The selected suggestion is then submitted as an *association* to that topic. If none of the suggestions were satisfactory, the player can still submit his guess as it is. The process of displaying relevant suggestions is managed by the suggestions-box, which is discussed in Section 3.2.

Throughout the game, each player can see the associations he submitted along with the score of each. The scores are increased dynamically when others have submitted the same association. This motivates players to enter associations that others would agree upon, consequently countering the subjective nature of the players' inputs.

The Recap Page. When the game's time is elapsed, the players are forwarded to the recap page, where they can see the associations submitted by all other players, as well as their scores. Players can then decide whether to join the next game in the series with the same players or join a new one.

3.2 The Suggestions-Box

The Knowledge Test Game offers a suggestions feature that enhances the data collection process, in addition to making the game more entertaining. The most important purpose of the suggestions-box, is to link the players' text input back to Linked Data. Each of the suggestions corresponds to a Linked Data entity. Since the topic is a Linked Data entity as well, linking the topic and an association results in connecting Linked Data entities.

The suggestions-box makes it easier to match submissions. Facilitating the matching process is in our interest as well as the players', since we will be getting more useful information, and the players will be getting more matches and consequently better scores.

The Knowledge Test Game does not rely on the submitted guess to find a match, but rather uses it as a clue to display relevant associations, and then collect the selected association afterwards. For example, if the current topic is "Egypt", and three different players submitted "pyramids", "the pyramids" and "Egyptian pyramids". It would be challenging to detect a match, although they could have meant the same thing. On the other hand, once the suggestions-box displays the suggestions for each of these guesses, the players would eventually pick the association that they meant, which could be "Egyptian pyramids", realizing that it best matches what they had in mind. Consequently, the three

players will get matches and therefore bonus points, and the game will give the association `dbpedia:Egyptian_pyramids` a higher rank.

Another immediate benefit of the suggestions-box is to distinguish ambiguities. When a player submits "pyramids" as a guess, he could have meant the geometric shape, the Egyptian pyramids, the Mayan pyramids, or anything else named pyramid. The suggestions-box clears these ambiguities, by allowing the player to further distinguish what he has meant by his guess.

The suggestions-box makes use of features from Google and Bing, which include auto-correction and being lenient towards different representations of the same word. Therefore the possible negative impact of using different dialects, or even languages, is absent. For example, submitting the British "organisation" and the American "organization" will result in two very similar, if not identical, suggestions lists. Players can even enter hints to the association instead of an exact association name. For example, a player can submit "c inventor" as a guess for "Deaths in 2011", and get a suggestions list that includes "Dennis Richie", who died in 2011, and who is also the inventor of C.

Furthermore, the suggestions-box can accept any language, including complex ones such as Arabic, or even transliteration[2] of Arabic words in English literals, and still yield relevant results. Nevertheless, regardless of the used language, the resulting suggestions always correspond to English Linked Data entities.

The Other Box. Players are also allowed to submit their guess as it is, by using the *other box* at the bottom of each suggestions list. Submitting a guess this way allows the player to come up with own associations which are not well represented or outside the scope of Wikipedia, at the expense of making it harder to match with other players. In order to get bonus points for an association submitted using the *other box*, other players have to submit the exact same string. In order to analyze the importance of such an association the game creates URIs of the form `ktg:<topic>/association/<association>`, creating new Linked Data entities (for a discussion of this effect see Section 5).

Approaches to Implement the Suggestions-Box. The goal was to present the players with associations relevant to the entered guess, in the context of the topic in question. Therefore, the retrieval method is a function of the player's *guess* and the game's *topic*.

The initial step was to manually collect associations for topics, to formulate a ground truth, with which we could benchmark different methods of collecting associations. We asked 9 participants to name associations to random topics, each coupled with one or more links to corresponding Wikipedia articles, ordered by relevance. We collected a total of 224 Wikipedia articles as associations to 32 different topics (full list is available at `http://goo.gl/hXhFt`).

[2] Transliterating Arabic words to English is common over the Internet in the Arab world. See `http://en.wikipedia.org/wiki/Arabic_Chat_Alphabet`.

Table 1. The mean *Recall@10* and mean *GamePlayability@10* achieved by each methods in attempt 1

	Mean Recall@10	Mean Game Playability@10
DBpedia Spotlight	26.17%	29.91%
Freebase	34.15%	39.28%
Bing	40.3%	48.6%
Google	49.69%	59.81%

After collecting the ground truth, we started testing different methods of retrieving these links in order to find a suitable one to be used for the suggestions-box. The first attempt to retrieve relevant links, was to query for the *Topic* and the entered *Guess*. We refer to this query as $T + G$.

To evaluate the results, we used *Recall@k* by calculating the percentage of the ground truth links retrieved out of the top k links obtained using the retrieval method. It was also significant to see if the retrieval method was able to retrieve any of the ground truth links at all. For this we defined a metric, called *GamePlayability@k*, which is 1 if any of the ground truth links exist within the first k retrieved links, and 0 otherwise.

In an effort to provide players with ten relevant suggestions for each guess, various APIs were evaluated to seek the highest *GamePlayability@10* and *Recall@10*. Among the tested APIs were DBpedia Lookup API, which was excluded for its strictness, as it expects a query string that is an exact substring of a URI's label. Wikipedia API had a very slow response rate for an interactive game, and was excluded accordingly. Finally, we tested the query using DBpedia Spotlight, Freebase, Bing and Google (see Table 1).

In the second attempt, we classified the results into three categories: those related to both the *Topic* and the *Guess* ($T + G$), those related to the *Guess* only (G), and those related to the *Topic* only (T). We reached a hypothesis that we can achieve better results by searching for $T + G$, in addition to promoting results common with G, and demoting those common with T. We refer to this merging process as `merge(T+G, G, T)`.

Google and Bing were preferred for this attempt because of their previous plausible results, and their quick response rate. Upon applying `merge`, there was a considerable increase in both the *Recall@10* and *GamePlayability@10*. Bing got a mean *Recall@10* and a mean *GamePlayability@10* of 71.34% and 77.57% respectively, while Google got 79.78% and 85.51%.

Google's results were better, while Bing had a faster response rate. We exploited this for the third attempt, by making three concurrent requests to each search engine. The final results are then passed to the merging algorithm again `merge(mergeGoogle, mergeBing, [])`, where `mergeGoogle` and `mergeBing` were the results of applying `merge(T+G, G, T)` on Google and Bing respectively.

This further increased the mean *Recall@10* and mean *GamePlayability@10* to 80.37% and 86.45% respectively, to reach the highest values we could achieve, without introducing any time overhead.

3.3 Topic Selection

Presenting players with topics that they are familiar with increases the fun factor of the game, as well as the validity of the results, since users with interest in a topic are more qualified to provide valid associations.

In order to focus on topics that are likely to be known, we collected the top most visited $10K$ Wikipedia articles in 2011[3]. Knowing that each of these articles corresponds to a Linked Data entity, the topics are randomly selected from their titles.

There are some restrictions in the context of topic selection that increase the validity of the players' submissions. These restrictions are shared by all the players within the same game. For example a topic cannot be played by the same player more than once, as we wanted to exclude possible influence from earlier games.

The Knowledge Test Game is also available on Facebook. By logging in using a Facebook account, the topic selection process is additionally influenced by the players' likes on Facebook, to make it more likely to get topics of interest.

If 50 unique players provided associations to a topic, the topic will be marked as *done*, and can be optionally prevented from appearing in future games. This gives the chance to analyze the collected associations, and to focus on other topics. The topic selection algorithm is biased towards closing topics as early as possible, meaning that if there are several topics available for a game, the one that was played most is preferred.

3.4 Generated Dataset

We keep track and log a lot of data based on the users' input. The data is made available online through `http://knowledgetestgame.org/export`. The main components of interest are the players' guesses. For every submission, the guess string provided by the player is stored along with the list of suggestions that he sees afterwards. Within the same record we also log the game's ID, the topic's name and URI, as well the player's ID and account type (the ID hides all potentially personal information about the player).

When a player selects an association from the list, the same record is updated to hold the association's URI and its index with respect to the suggestions list. The time of submitting the guess, and the time of choosing the associations are both stored. We also keep track of the time taken by the player, in milliseconds, to choose the association from the list. The number of occurrences and the score of the association across the game are also logged. Furthermore, each record holds "nth guess" and "nth association" which show the record's submission order as a guess and its order as an association by that player in the given game.

4 Evaluation

After the previous sections focused on the game, its suggestion box and topic selection, we will now provide a detailed evaluation of the game itself and of the generated output.

[3] Obtained from `http://dumps.wikimedia.org/`

4.1 The Game

First, the game's concept and its realization are evaluated by summarizing measurements and derived estimates. Afterwards, the outcomes of a questionnaire, which was presented to players of the game, are provided.

Measurements and Estimates. The game was run in several focused experiments, that added up to 26.6 hours of game-play time by humans. In these experiments the game was played by 267 different players who played a total of 1046 games together collecting 6882 ranked associations.

Using these numbers we can evaluate the game wrt. the *throughput*, *average lifetime play* and *expected contribution* metrics for Games with a Purpose defined by von Ahn and Dabbish [5].

The *throughput* is calculated by dividing the collected data (6882 ranked associations) by the total human game-play time (26.6 hours), resulting in ~ 259 ranked associations per human hour. At this rate if there were 50 players online for a day playing the game (a decent estimate for typical online games), we could collect about 310 800 ranked associations in a single day.

We can also compute the *average lifetime play* by dividing the total game-play time (26.6 hours) by the number of players (267), resulting in an average lifetime play of ~ 6 minutes per player, which is equivalent to the time needed for ~ 8 games.

Finally, we can calculate the *expected contribution* by multiplying the average lifetime play with the throughput, resulting in an expected contribution of $\sim$ 25.78 ranked associations per player.

Questionnaire. Apart from the metrics in the previous section, we conducted an online survey which was filled out by 21 players after playing the game. Most of the participants were students from Egypt and Germany, between 20 and 25 years old, had a computer science or engineering background, had played web games before and described their English skills as fluent. Besides these demographic questions, the survey consisted of 3 open and 13 5-point Likert scale questions. The 3 open questions were asked beforehand in order not to influence participants. The text of the questions was: "What did you like about the game?", "What did you dislike about the game?" and "What would you improve?".

Summarizing most players liked the idea of the game and described it as fun, mentally challenging and interesting to compare their own thoughts to those of others. Many participants mentioned that they enjoyed the topic mix and were surprised by the quality of the suggestions-box:

> It is very challenging, not only are you challenging the other players, but also your own knowledge The topics are very good. The recommended words are very good, Ex. I got the topic "Princess Diana" and I wanted to add the name of the man she was with in the car accident but I couldn't remember his name, I just know he was Egyptian, I wrote down "Egypt" and I found "Dodi Al Fayed".. very cool!:)

In the dislike section it was mentioned that some topics were too vague or unknown, that the suggestions-box sometimes was slow and that the 45 seconds per round were not sufficient to enter all your associations in some cases. Also some participants complained about the little information they got about other players which was in line with the improvements section.

Here we received a lot of feedback that can be grouped into the category enhancing the interaction with and information about other players. Many participants want to know more about the people they're playing with and suggested to introduce a chat after the game in the recap phase. Others want to be able to play with their friends. Also participants mentioned that they would want to see global high-scores after the round and live stats of other players in their game during the game, so they don't have to wait for the recap page to see their own performance. Furthermore, it was suggested to provide the ability to select categories of topics to play, to show photos for the topic or for vague topics to provide hints by showing some of the most often entered associations.

Table 2. Results of an online survey answered by 21 game players. Except for *Age*, users could select answers from a 5-point Likert scale. If not indicated otherwise the options were: 1 (Strongly disagree), 2 (Disagree), 3 (Neutral), 4 (Agree), 5 (Strongly agree).

Statement	μ	σ
"The game rules and concept were direct and straight forward."	4.5	0.8
"The How To Play tutorial was..." (useless ... useful)	4.8	0.4
"45 seconds for the game were..." (too short ... too long)	2.6	0.7
"The topics were clear and know to me."	4.0	0.8
"The suggestions were relevant to what I had in mind."	4.1	1.0
"The suggestions that I got for a guess influenced my following guesses."	4.0	0.9
"15 seconds for the recap page were..." (too short ... too long)	3.1	0.6
"I understood the recap page."	4.6	0.6
"I was interested in reading the scores in the recap page."	4.5	0.7
"Seeing my partner's answers influenced my guesses in the following games."	3.2	1.3
"I enjoyed the game."	4.5	0.7
"I would play it again"	4.3	1.2
"I played web games before."	4.0	1.2

The findings from the open questions were refined by 13 questions in which participants could select numerical values between 1 and 5 (5-point Likert scale). The results are summarized in Table 2. In general we can see that the game concept was easy to understand, people found the tutorial useful, knew the topics, found the suggestions relevant to what they had in mind, understood the recap page and were interested in it and that most people enjoyed the game and would play it again. The timing restrictions of 45 seconds per round was perceived as slightly too short, but 15 seconds for the recap page were just right.

The questionnaire identified a key problem, namely that many participants had the feeling the suggestions-box influenced their following guesses. This effect was later mitigated by reducing the suggestions from ten to four (see Section 5). The effect seems to be less pronounced for the recap page.

Before discussing these findings and possible solutions, we first want to present our evaluation of the data collected.

4.2 Data Quality

In order to assess the quality of the collected data, we aggregated the associations collected by the game for each topic. Focusing on topics for which the most associations were submitted by players, we counted the number of occurrences of each association and ordered them descending by counts. In this process we excluded associations which were submitted by less than two players as a provisional filter against noise.

After the first major experiment, the resulting ordered lists of associations for the 10 topics which were played most often were generated. With these lists we conducted another online questionnaire with 36 participants out of which 19 had played the game. The participants' demographics resembled those of the game players: they mainly were computer science students from Egypt and Germany, between 20 and 25 years old and described their own English skills as fluent. In the questionnaire for each of the topics we asked the participants to rate the ordering of the list of associations on a scale from 1 (Makes no sense at all) to 5 (Makes perfect sense). The histogram of the ratings can be found in Figure 2 and indicates that the majority of participants were very satisfied with the presented associations and their ordering. With $\mu = 4.2$ the average over all ratings ($\sigma = 0.9$) is close to its maximum of 5.

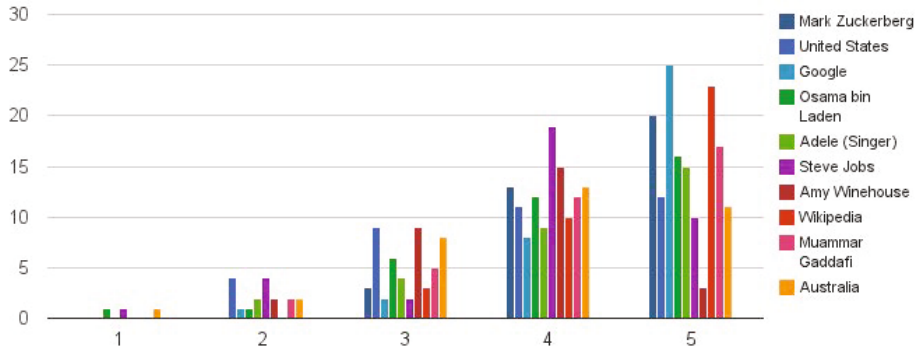


Fig. 2. Histogram of ratings for the ordered lists of associations. For each topic the participants could chose on a scale from 1 (Makes no sense at all) to 5 (Makes perfect sense).

After a second large experiment we chose another form to evaluate the generated association lists (an example can be seen in Table 3). We again conducted an online survey, this time with 17 participants, where they were asked to rank given randomized lists of the top-20 associations for the most often played topics. By then, we had 15 *done* topics (i.e. played by more than 50 players). Out of these 15 topics the 9 lists summarized in Table 4 were picked to form a ground truth, as they had been ordered manually by more than 5 participants. The ground truth was formed by averaging the individual ranks of the manually

Table 3. The most frequently submitted associations for the topic Mark Zuckerberg

Association	Times mentioned
Facebook	50
The Social Network	15
Chief Executive Officer	12
Rich	8
Millionaire	7
Social Network	6
Entrepreneur	5

Table 4. The 9 most often played topics. The associations are printed as titles here instead of the URIs of the corresponding DBpedia instances. Each topic's associations lists were presented in the questionnaire in a randomized order, where participants were asked to rank them. The resulting ranks were then compared with the nDCG to those generated with the Normalized Google Distance (NGD) and the game.

			nDCG	
Topic	Top-N Associations	Manual sorting	NGD	Game
Charlie Sheen	8	7 participants	0.860	0.969
Eminem	11	14 participants	0.870	0.931
Lady Gaga	18	9 participants	0.806	0.924
Mark Zuckerberg	7	15 participants	0.895	0.954
Osama bin Laden	12	7 participants	0.814	0.835
Transformers: Dark of the Moon	18	6 participants	0.768	0.926
United Kingdom	14	7 participants	0.806	0.873
World War II	17	17 participants	0.876	0.953
YouTube	10	17 participants	0.927	0.928
		μ	0.847	0.921
		σ	0.051	0.042

ordered lists of the participants and sorting the associations accordingly. Afterwards, the normalized Discounted Cumulative Gain (nDCG) was calculated to compare the manually ranked ground truth association lists with those retrieved by the game. As a relevance metric, we used a linear mapping of the top element to a relevance of 1 down to the last element with a relevance of $\frac{1}{n}$.

In order to differentiate our game's results from simple corpus based similarity metrics, we also re-ranked the ground truth lists according to the popular Normalized Google Distance (NGD) [12]. As the NGD calculates a similarity between pairs of entities only and cannot trivially be used to find the top candidates for a given topic we artificially enhanced the method by only focusing on the top-20 candidates in the ground truth. The nDCGs can be found in Table 4 as well. We discuss our results and findings in the next section.

5 Discussion

After detailing our evaluation in the previous section, we will now discuss our findings. In summary we were very satisfied with the results of our evaluations, as the game was well perceived and fun for the players and also collected associations of high quality.

We consider the achieved throughput of 259 associations per human hour quite satisfactory, as it means that on average less than 14 seconds were spent for typing in a guess string, waiting for the suggestions-box and selecting one of the alternatives. As many players complained that the suggestions-box was slow we investigated our server logs to find that under high load it seems our requests to Google were rate limited, resulting in an average response time of the suggestions-box of approx. 2.3 s. At the same time all 3 requests to Bing on average return within 250 ms. As we also got a lot of feedback that the quality of the suggestions-box is astonishing, we would like to keep using the merged results of Google and Bing. In order to decrease the delay we consider more aggressive caching. Also we plan to include incremental updates of the suggestions list to lower the waiting time and increase the throughput in future versions.

In order to solve ambiguity issues of the strings displayed in the suggestions list, we plan to display the `rdfs:comment` or a useful `rdf:type` from DBpedia in future versions. At the same time a `foaf:depiction` could be shown to make the suggestions visually recognizable. As queries to the online DBpedia will take additional time we again consider caching and asynchronous updates of the GUI.

The evaluation also revealed the problem that later guesses were likely to be influenced by the displayed suggestion lists for preceding guesses. Throughout the experiments we therefore collected the index (zero-based) of the association selected from the suggestions list. On average the second (1.04) suggestion was selected with a standard deviation of 1.7 in the first major test. Based on that, we recalculated the *Recall* and *GamePlayability* using different ks ranging from 1 to 10. *Recall*@4, which translates to showing 4 suggestions, was found to be a suitable compromise to mitigate the influence (see Figure 3). Another alternative we want to investigate in the future consists of further reducing the amount of suggestions and providing a "more" button.

We were very pleased with the evaluation of the data quality, as the game shows a high average nDCG of 0.921 (Table 4) in comparison to the ground truth. The comparison to a popular corpus based technique shows that even when enhanced with an oracle that only suggested the associations we consider correct, the corpus based technique still was not able to rank the associations as well as the game (average nDCG of 0.847).

Last but not least, we investigated a potential design issue of our approach, which links Linked Data entities to one another. Our approach thereby neglects the possibility that people could want to associate a Linked Data entity with one of its Literals. Hence, we studied the list of all associations which were submitted with the "other" option of the suggestions-box and all guesses for which no association was selected, coming to the conclusion that not a single one of them corresponded to a desired but missing literal value in the suggestion lists. Also we were surprised how seldom players seemed to have missed an association target. From this we conclude that even though theoretically possible it seems to be very rare that people would want to associate an entity with one of its literals or cannot find a desired association target in the domain of Wikipedia. Nevertheless, in future versions we plan to explicitly log events when one of the

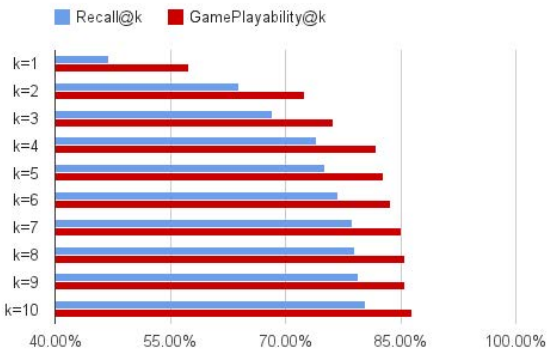


Fig. 3. The Recall@k and GamePlayability@k from k = 1 to k = 10

guesses matches one of the topic's literals and when newly created linked data entities are matched multiple times.

6 Conclusion and Outlook

In this paper we presented our idea to rank Linked Data facts according to human association strengths to cope with the increasing information overflow when performing simple queries on Linked Data entities. In order to collect a dataset of such association strengths between Linked Data entities we developed a game with a purpose called "Knowledge Test Game".

Our evaluations show good results wrt. throughput and perceived fun of the game, especially the quality of the suggestions box received a lot of positive feedback as it is even able to retrieve complex, clue based associations. Furthermore, collected data seems to be of very high quality.

Apart from the planned improvements mentioned in Section 5 our future work on the game will mainly focus on making it more desirable for players to stay in the game in order to collect more and more data, for example providing a chat on the recap page, global high-scores, an exponential scoring scheme, player ranks and permissions (such as reporting cheaters). We would also like to experiment with social gaming aspects such as team games by taking more advantage of the Facebook integration. Furthermore, we plan to provide a transparent single-player mode where players play against recorded sessions of other players in order to reduce waiting times, validate existing data and detect cheaters.

In terms of data quality we want to investigate other aggregation methods, for example taking the submission order of the associations into consideration. Also we would like to experiment with the thresholds to close topics as well as exclude noisy associations.

Last but not least we want to use the collected association data published at `http://knowledgetestgame.org/export` to evaluate existing or future methods to rank Linked Data according to human associations.

This work was financed by the University of Kaiserslautern PhD scholarship program and the BMBF project NEXUS (Grant 01IW11001).

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American 284(5), 34–43 (2001)
2. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data - The Story So Far. International Journal on Semantic Web and Information Systems 5(3), 1–22 (2009)
3. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia - A crystallization point for the Web of Data. Web Semantics: Science, Services and Agents on the WWW 7(3), 154–165 (2009)
4. Berners-Lee, T.: Linked Data - Design Issues (2009)
5. von Ahn, L., Dabbish, L.: Designing games with a purpose. Communications of the ACM 51(8), 58–67 (2008)
6. Hees, J., Roth-Berghofer, T., Biedert, R., Adrian, B., Dengel, A.: BetterRelations: Collecting Association Strengths for Linked Data Triples with a Game. In: Ceri, S., Brambilla, M. (eds.) Search Computing III. LNCS, vol. 7538, pp. 223–239. Springer, Heidelberg (2012)
7. Siorpaes, K., Hepp, M.: Games with a Purpose for the Semantic Web. IEEE Intelligent Systems 23(3), 50–60 (2008)
8. Hees, J., Roth-Berghofer, T., Dengel, A.: Linked Data Games: Simulating Human Association with Linked Data. In: LWA 2010, Kassel, Germany, pp. 255–260 (2010)
9. Hees, J., Roth-Berghofer, T., Biedert, R., Adrian, B., Dengel, A.: BetterRelations: Using a Game to Rate Linked Data Triples. In: Bach, J., Edelkamp, S. (eds.) KI 2011. LNCS, vol. 7006, pp. 134–138. Springer, Heidelberg (2011)
10. Kny, E., Kölle, S., Töpper, G., Wittmers, E.: WhoKnows? (October 2010)
11. Wolf, L., Knuth, M., Osterhoff, J., Sack, H.: RISQ! Renowned Individuals Semantic Quiz – A Jeopardy like Quiz Game for Ranking Facts. In: Proc. of the I-SEMANTICS, Graz, Austria, pp. 71–78. ACM (2011)
12. Cilibrasi, R.L., Vitányi, P.M.B.: The Google Similarity Distance. IEEE Trans. Knowledge and Data Engineering 19(3), 370–383 (2007)

Personalized Concept-Based Search and Exploration on the Web of Data Using Results Categorization

Melike Sah and Vincent Wade

Centre for Next Generation Localisation, KDEG, Trinity College Dublin, Dublin, Ireland
{Melike.Sah,Vincent.Wade}@scss.tcd.ie

Abstract. As the size of the Linked Open Data (LOD) increases, searching and exploring LOD becomes more challenging. To overcome this issue, we propose a novel personalized search and exploration mechanism for the Web of Data (WoD) based on concept-based results categorization. In our approach, search results (LOD resources) are conceptually categorized into UMBEL concepts to form *concept lenses*, which assist exploratory search and browsing. When the user selects a concept lens for exploration, results are immediately personalized. In particular, all concept lenses are personally re-organized according to their similarity to the selected concept lens using a similarity measure. Within the selected concept lens; more relevant results are included using results re-ranking and query expansion, as well as relevant concept lenses are suggested to support results exploration. This is an innovative feature offered by our approach since it allows dynamic adaptation of results to the user's local choices. We also support interactive personalization; when the user clicks on a result, within the interacted lens, relevant categories and results are included using results re-ranking and query expansion. Our personalization approach is non-intrusive, privacy preserving and scalable since it does not require login and implemented at the client-side. To evaluate efficacy of the proposed personalized search, a benchmark was created on a tourism domain. The results showed that the proposed approach performs significantly better than a non-adaptive baseline concept-based search and traditional ranked list presentation.

Keywords: Concept-based search, personalized search/exploration, linked open data, UMBEL, query expansion, results re-ranking, interactive personalization.

1 Introduction

With the adoption of the LOD by a wider Web community, large volumes of semantic data are being generated. The challenge now is finding and exploring relevant information on the WoD. This is crucial for the uptake of the LOD by applications in order to support both ordinary Web and Semantic Web users with innovative user interfaces. In this context, LOD search engines play a vital role for providing efficient access mechanisms. However, current approaches (e.g. Sindice [1], Watson [2]) adopt keyword-based search and ranked result lists presentation of traditional Information Retrieval (IR), which is not very efficient for large volumes of data [3]. In ranked lists,

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 532–547, 2013.

users cannot understand "what the resource is about" without opening and investigating the LOD resource itself. There is a need to investigate search problems on WoD.

Another search paradigm for the LOD is faceted search/browsing systems, which provide facets (categories) for interactive search and browsing [4]. Facets assist results filtering and exploration. However, the main limitation of faceted search is that facet creation depends on specific data and schema properties of underlying metadata and it can be difficult to generate useful facets to large and heterogeneous WoD [5]. Based on the existing work, it is evident that LOD search mechanisms need improvement, which is our main objective. This is crucial for exploration of WoD data/datasets and uptake of LOD by wider community not just Semantic Web experts.

Traditional IR has been investigating efficient search mechanisms for decades; results clustering and personalized search are two popular methods for enhancing search effectiveness. In clustering search, results are organized into categories for assisting users in results exploration and in disambiguation of the query (Snaket [6], Vivisimo.com, carrot2.org). For example, the query "tiger" may match to animal, computer or golf result categories. The user can disambiguate the query by selecting the correct category. Results categorization is used widely, such as Google categories, Yahoo Directories and Open Directory Project (ODP). Although clustering search and faceted search seems similar, the latter filters results based on schema/metadata, whereas the former clusters results based on their meaning (language model).

On the other hand, personalized search aims to improve retrieval efficiency by adapting results to context/interests of individual users; thus the user can explore personally relevant results. It is a popular research topic and commercial interest (i.e. Google). However, personalized search gained very little focus on the Semantic Web. This could be because of isolated and low volumes of metadata created in early linked data initiatives. As the size of LOD increases, personalized search and interactions become more important. We innovatively combined results categorization and personalized IR to introduce a novel personalized search and exploration mechanism.

1.1 Contributions

In our approach, users access to the WoD with (keyword or Uniform Resource Identifier (URI)) queries. UMBEL conceptual vocabulary (umbel.org) is used to categorize the retrieved LOD resources (search results) into concepts. UMBEL provides a hierarchy of ~25,000 broad concepts that are organized into 32 top-level supertype categories. UMBEL is also interconnected to linked datasets (i.e. DBpedia, GeoNames, Opencyc, schema.org), which can be used for results presentation. Results categorization is achieved by the proposed fuzzy retrieval model [8], which works on any linked dataset, scalable and reasonably accurate (~90% on 10,000 mappings). Alternatively, other methods can be utilized for categorization. For each query, our engine provides results and their UMBEL concepts. On the client-side, results with the same concepts are grouped to form *concept lenses.* Concept lenses favour results exploration and help to disambiguate the meaning of the query. In particular, concept lenses support informational queries (i.e. the intent is to acquire information). It is estimated that ~80% of Web queries are informational [10].

In our approach, personalization is applied in two phases: (i) When a user select a concept lens from the result lists for exploration, immediate personalization is applied; all concept lenses are re-organized according to their similarity to the selected concept lens using a similarity measure. This is a novel method, which allows re-organization of all results based on conceptual and syntactic similarity to a particular lens. In addition, within the selected concept lens, immediately more relevant results are included using results re-ranking and query expansion as well as relevant categories are suggested for results exploration. (ii) We also support interactive personalization. To achieve this, last *N* clicks of the users within a search session are monitored. When the user clicks on a result, within the interacted lens, immediately personalization is applied. Such as, relevant results and lenses are added by query expansion and results re-ranking. The adapted concept lenses are referred as *personal lenses*.

Our contributions are: (i) We propose a novel personalized concept-based search and exploration mechanism for the WoD. To the best of our knowledge, no such previous work exists. (ii) We suggest the use of results categorization as a tool for personalized concept lenses re-ranking, results re-ranking and query expansion. The evaluations have indicated that the use of these personalization and lenses approach greatly enhances retrieval precision. In particular, the key idea is that the user clicks on a concept lens that best suits his/her information needs. Given the selection, our approach personalizes the order of concept lenses. In addition, within the selected lens; the ranked list is personalized to push up the relevant results and the category label is used to generate an expanded query to retrieve more relevant results. We think that this is an innovative feature offered by our approach since it allows dynamically adaptation of results to the user's local choices. In addition, we support interactive personalization following user clicks onto results. (iii) Our personalization approach is non-intrusive, privacy preserving and scalable, since it does not require an explicit login by the user and the personalization is implemented completely at the client-side. (iv) Our approach is adaptable and can be plugged on top of any Linked Data search engine; in this paper, we use Sindice [1]. It only requires UMBEL categorizations, which can be achieved by number of methods such as the fuzzy retrieval model [8].

Section 2 discusses related work. In section 3, the system architecture is introduced. Section 4 introduces personalized concept-based search methods. Section 5 shows evaluations on a benchmark dataset. Section 6 provides conclusions and future work.

2 Related Work

2.1 Search Mechanisms - Clustering, Faceted and WoD Search Engines

Clustering or concept-based search (conceptual search) aims to improve retrieval effectiveness by organizing search results based on their meaning [6]. Open Directory Project and Yahoo Directory for instance use manual categorization, which is not scalable. Conversely, automatic clustering of results is scalable but challenging. Approaches usually use data mining, NLP and statistical techniques (e.g. k-means clustering) to calculate document similarities, form/label clusters and present flat or hierarchical result categories ([6], vivisomo.com, carrot2.org). In contrast to IR approaches, we use LOD

resources rather than documents; we extract semantic data from the context of resources for categorization in UMBEL concepts (see section 3 for details).

Faceted search [4] allows interactive filtering of results based on shared schema properties. Generally, faceted search uses labeled graph [18][19] or textual overviews (semantic properties as browsable facets). In both cases, usability/efficiency decreases as the complexity of information space increases. To increase usability of information visualization on huge repositories, [20] describes "*overview first, zoom and filter, then details on-demand*" fashion. [18] uses both statistical knowledge and graph structure (subject and broader topics) to estimate resource popularity for graph presentation in DBpedia. Whereas, [19] utilizes clustering and personalization in a multimedia domain to decrease visualization complexity. Faceted search is typically applied in closed domains since it requires high data completeness and consistent markup across the whole corpus. Considering the varying data quality and heterogeneous vocabularies of the WoD [5], it can be challenging to generate consistent facets for the whole LOD. Moreover, applying dynamic conjunctive clauses on large datasets significantly increases complexity of faceted search. Our approach works on open corpus of LOD resource thanks to the use of the proposed fuzzy retrieval and UMBEL [8].

Finally considering the large body of work on clustering or faceted search, current WoD search mechanisms (Sindice [1] and Watson [2]) utilize traditional full-text retrieval and ranked result lists, which are not focusing on data exploration problems. Users cannot understand "what the resource is about" without opening and investigating the LOD resource, since title/triples are not informative enough. Sig.ma [3] attempts to solve this issue using querying, rules, machine learning and user interaction. However, Sig.ma's focus is on data aggregation. Another relevant aspect of semantic search is the way users express their information needs. Keyword queries are the simplest and widely used approach [1][2][13]. Natural language queries increase expressiveness such as linguistic analysis can be applied to extract syntactic information [17]. Controlled natural language queries are also utilized, where query can be expressed by values/properties of an ontology [14]. Finally, the most formal systems use ontology query languages (i.e. SPARQL), which demands high expertise and impractical from usability point of view. A trade-off between expressivity and usability should be achieved. Compared to existing work, we propose a unique concept-based personalized search and exploration for the WoD. In our approach, we use keyword queries and results are categorized into concept lenses to support exploration. Categorization acts as a tool for personalized lenses/results re-ranking and query expansion.

2.2 Personalized Information Retrieval

Personalized IR is a popular topic in traditional Web. Generally, personalized IR comprises of: (1) User data gathering, (2) user profile representation and (3) personalization techniques. User profiles can be created from [11]: explicit/implicit user relevance feedback, desktop, social Web or user's context. In our work, we use user's context. The advantages and disadvantages are discussed further in section 4.1.

General user profile representation methods in personalized IR are: weighted keywords, semantic network of terms or semantic network of concepts [12]. The simplest

model is weighted vector of keywords. However, keyword-based representation does not capture semantics of related terms. Ontology-based profile representation techniques try to overcome this problem. [12] utilizes the entire ontology for representing user profiles. Extracted keywords from the browsed pages are matched to ontology concepts and concepts are represented as weighted vector of keywords. Generally user profiles are utilized for results re-ranking. In contrast to the general approach, our personalized search approach is driven by results categorization. We need to represent user interests as concept lenses for lenses re-organization and capture user's information needs from clicked results for results re-ranking. Lenses are re-organized based on user's local user choices, hence correct personalized re-ordering of categories significantly affects precision. For this purpose, we represent concept lenses with three rich sources of data for similarity comparison. First, all results within the concept lens are combined to create; a vector of UMBEL concepts (specific user interests); a vector of supertype concepts (broad user interests); and a vector of terms (for language comparison). In addition, we track last *N* user clicks within the current search session to represent user's interests for specific concepts, broad concepts and terms for results re-ranking. We represent user interests using combined ontology-based and keyword-based vectors. Usually either one of these representations is used.

Query disambiguation, query expansion, result re-ranking, results filtering, hybrid methods and collaborative adaptation are common personalized IR techniques [11]. Two popular techniques are query expansion and results re-ranking. Query expansion methods augment the query with terms that are extracted from interests/context of the user so that more personally relevant results can be retrieved. A general limitation is that if expansion terms are not selected carefully, it may degrade the retrieval performance. Conversely, in result re-ranking (rank biasing), the initial set of results are retrieved and the results are re-ranked based a user profile (i.e. profile similarity [12][13]). The aim is to push personally relevant results up in the result list.

In personalized IR, generally user's activities with the retrieval system are continuously monitored for results adaptation (i.e. Google, amazon). This approach requires explicit login by the user and storage of the user information at the server-side, which raises privacy issues. However, relying on all past user interests is tricky and often a correct subset of past interests needs to be identified for correct personalization based on the current information needs. Therefore, in our approach, we only use the current search context, hence it does not require user login. As a result, our approach provides personalization according to local choices of the user based on results categorization. A similar work is [6], which uses hierarchical page snippet clustering for personalized search. Results are categorized into hierarchical folders using gapped sentences and ODP. The user need to select a list of relevant labels related to his/her information needs. Then relevant results are filtered, and the query is expanded. In our approach, all relevant lenses are implicitly re-organized when the user selections a concept lens. This is different from the approach in [6], as it requires explicit selection of all relevant labels. In addition, we apply results re-ranking, query expansion and category suggestion within the selected concept lens as well as present the results using concept lenses rather than ranked lists of [6].

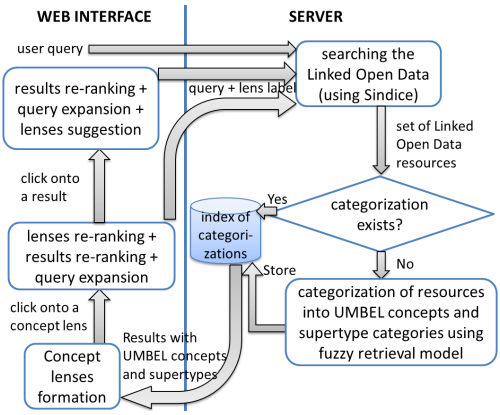
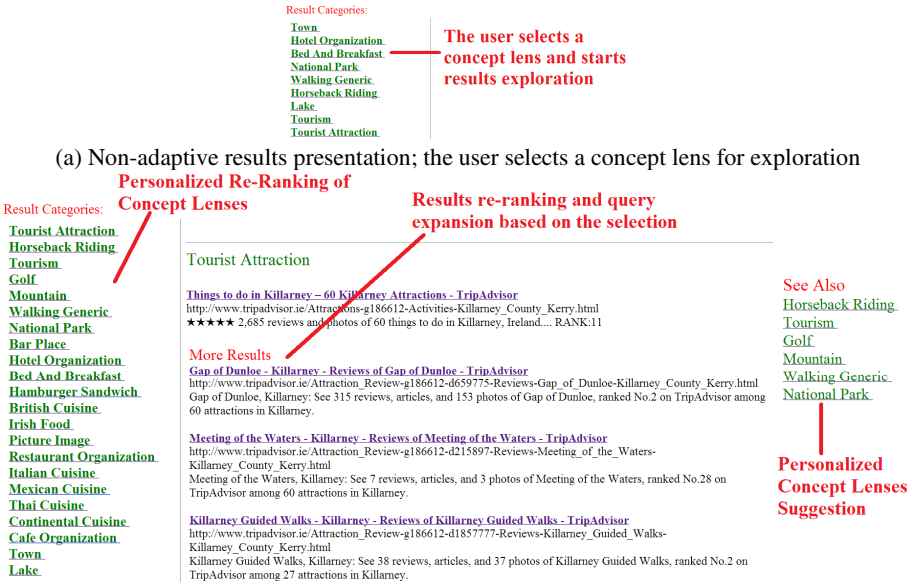


Fig. 1. The architecture and workflow

3 Proposed Personalized Concept-Based Search Framework

The proposed system architecture is shown in Figure 1. Users can provide keyword or URI queries to the system. Using the input queries, the WoD is searched. Our approach can be plugged on top of any LOD search engine (currently using Sindice search API). After receiving results, our system augments the results with UMBEL categorizations, which can be performed offline or dynamically [9]. For example, using a crawler and Sindice, LOD resources can be categorized offline by the proposed fuzzy retrieval model [8], or other clustering methods (also UMBEL linked data mappings can be used). New LOD resources are incrementally categorized and indexed at the server-side for a scalable performance [9]. In particular, we use the whole 5-depth UMBEL hierarchy (~25,000 concepts); a LOD resource may match to any concept, which is different than many personalized IR methods. Generally the top level or top 2 levels of the ontology (~200 categories of ODP) are used to represent search results. However, such an approach can only model general user interests. To achieve categorization, we extract various semantic information from context of LOD resources; type, subject, labels, property names and URL labels. Our experiments [8] on 10,000 mappings indicate that subject and type properties provide the best information for categorization, while property names add significant noise. Extracted semantic information is mapped to UMBEL concepts using a fuzzy retrieval model [8]. In order to utilize the semantic relationships and similarity present in the UMBEL vocabulary, we use the hierarchical relationships between concepts to form the vectors to represent the concepts. Vector space representation of concepts is an accepted method [13][14][16], which allows scalable performance. This provides a simple way of encoding key semantic knowledge into IR retrieval model. We use only hierarchical relationships in UMBEL as the ontology does not contain the semantic relatedness relationships between the concepts. We alleviate this issue to an extent by extracting data from subjects of LOD resources. For example, semantically related resources may share common subjects, e.g. Pope and Vatican might share Christianity and Catholic subjects. To include semantically related concepts into the categorization, we associate each LOD resources to 3 UMBEL concepts. We use the most confident

concept (categorization with the highest score) for lenses creation (e.g. Pope) and the rest for semantic similarity comparison. Moreover, textual content is extracted from abstract/labels of resources to generate term vectors for combined semantic and syntactic similarity. Combining semantic and syntactic similarity provides better results [17] when the data is incomplete or poor quality (i.e. varying LOD quality).



(a) Non-adaptive results presentation; the user selects a concept lens for exploration

(b) Results are immediately personalized; the order of lenses are adapted, more relevant results are added by results re-ranking and query expansion, also relevant concept lenses are suggested

Fig. 2. Personalized concept-based search for the query "killarney sightseeing" [7]

For a scalable performance, LOD resources' UMBEL concepts and supertypes are also indexed at the server-side as discuss in [9]. Subsequently, uncategorized LOD resources can be dynamically categorized using asynchronous parallel requests between the client and server [9]. Categorized LOD resources (results) are sent back to the client and the results with the same concepts are grouped to form *concept lenses.* Specifically, we only use the most confident categorization to form lenses.

The user is required to select a concept lens in order to start exploring results. When the user clicks onto a concept lens, all the concept lenses are personally reorganized based on conceptual and syntactic similarity to the selected concept lens. In addition, following a lens selection, more relevant results are included based on: (i) results re-ranking using concept similarity and (ii) query expansion using the concept lens label. Moreover, when the user interacts with the results, result re-ranking, query expansion and concept lenses suggestions are provided. With our approach, personalized and conceptual result exploration is supported. This is especially useful in complex information needs, such as information gathering in an unfamiliar domain.

The client side is written using Javascript and AJAX. To support non-intrusive user modeling and adaptation, personalization is completely implemented at the client-side. Thus, it does not require user login since only user's click data within the current

search session is used. Client-side personalization is also scalable and computationally efficient since the workload is distributed to the clients and network traffic is significantly reduced. We use Sindice Search API to search the WoD and Lucene for indexing/fuzzy retrieval model. The server side is implemented with Java Servlets and uses Jena. In Figure 2, our search interface is shown. A demo can also be found at [7].

4 User Data Gathering and Search Results Personalization

4.1 Context-Based User Data Gathering

User profiles can be generated from relevance feedback, implicit relevance feedback, desktop data, social Web or user's context. Generally the effectiveness of relevance feedback is limited since users are often reluctant to manually provide information. Implicit relevance feedback thus uses interactions with the system such as previous browsing activity, time spent on pages, etc. as an indication of implicit user interests. In both cases, the system needs time to gather enough information about user's all past interests. To overcome this issue, some approaches utilize desktop data or recently social web data [12], which often contains enough information about general user interests. However, relying on all past user interests is tricky and often a correct subset of past interests needs to be identified, which can be very challenging. This is because not all past interests may be important in the current context and an incorrect personalization may annoy the user experience, e.g. a user looking for hotels in Florence will not be interested to get Florence hotels in results after booking a hotel. Thus, approaches based on all past interests require fine-tuning, such as threshold selection for similarity/time decay, which may differ from various users or search scenarios. Moreover, in long-term user profiling, the extracted user interests are usually stored at the server side. This means users are required to register and login to get benefit of the personalization, which often raises privacy issues. An alternative to this approach is, no login/no storage or client storage. Client-side storage has its own issues; users may have multiple access mechanisms to the internet (especially with growing mobile access devices). Thus the user profile may be dislocated to multiple devices and the user may get different personalization experience based on the device s/he used. Finally, in the context-based user modeling, only the current available information within the current search context is utilized (i.e. query, query context, clicked results, etc.). A benefit is system only deals with few number of interests hence performance is scalable. The drawback is past user interests are lost but not all past interests are useful or identification of related interests can be challenging as we mentioned earlier.

In our approach, we use context-based user modeling rather than background knowledge. Only click data within the current search session is used to adapt to user's local choices. Search 'session start' is when the user opens the retrieval interface and 'session ends' when she closes it. The system is able to cope with changes of search domain from categorization. Suppose the user refined a query; it is probable that similar concepts/supertypes will occur in new search results. However, if the search topic changes completely, categorization in ~25,000 UMBEL concepts will not be the same, thanks to the use of whole concepts. Fortunately, supertypes can be used to understand user's general interests even if search topic change. This approach is com-

plemented by categorization and interactive personalization. Suppose the user is interested in concept *x* and clicked onto a promising result in this lens. After a quick investigation, she deems the result irrelevant. However, this negative feedback is still very valuable thanks to categorization. By analyzing the last *N* clicks of the user on concepts/supertype concepts and the system can find similar LOD resources that share related concepts. In addition, we developed an interactive personalization where any feedback can be used. On click to a concept lens or a result, immediate personalization is supported such as lenses re-ranking, results re-ranking and query expansion.

4.2 User Profile Representation

For profiling, we track: (i) click onto a concept lens and (ii) clicks onto last *N* results.

User Concept Lens Choices: When a user clicks onto a concept lens, the results are adapted based on user's local choices. Accurate personalized re-ordering of lenses significantly affects precision @Top *N* results. Thus, robust and efficient similarity measure is essential for personalized lenses re-ranking. Similarity measures play an important role in IR, such as measuring relevance between the user's keyword query and set of pages. A majority of these measures are statistical or linguistic models for unstructured text documents. With the Semantic Web, semantic similarity measures are proposed to compare concepts and/or concept instances. They can be classified into structure and information based approaches. The structure-based methods use ontology hierarchical structure, such as edge distance between concepts. Information based methods use the shared content between concept features, e.g. comparing concepts' textual data using cosine similarity. Hybrid approaches combine both methods. For semantics-based IR, appropriate similarity measures depend on many factors, such as concepts representation (e.g. bag of words, logic predicates, etc.), search context and concept expressivity [15]. Description logic based approaches allow full expressivity but complexity can be high [22]. Overall, similarity measures depend on the application area. In our approach, we use a hybrid similarity measure combining hierarchical structure of the UMBEL vocabulary and shared statistical data between resources. We adopted vector space representation of the ontology [13][14][16] that allows efficient and scalable similarity compared to more complex description logic-based approaches. Since, performance is vital for on-time personalization.

To represent user interests, first information about all results under a concept lens are used to represent concept lens with; (a) *vector of UMBEL concepts*, i.e. user interests to specific concepts in a 5-depth hierarchy of 25,000 concepts. Unlike general approach, we represent results with very specific UMBEL categorizations. (b) *vector of supertype categories*, i.e. top-level categories to represent broad user interests. (c) *vector of terms*, i.e. terms extracted from results snippets such as title, url keywords and descriptions of the concept lens. Stop words are removed and terms are stemmed for comparison. User's interest for a concept lens is represented with three vectors. Suppose search results contain *m* concept lenses, *l*. Each concept lens, *l*, contains *n* results, *r*. Each result is represented with up to *k* UMBEL concepts, *c*, and their associated supertypes, *sc* (*k*=3 in experiments). *Vector of concepts* is calculated as:

$$\sum_{z=1}^{m} l_z = \sum_{i=1}^{n} \sum_{j=1}^{k} r_i c_j \rightarrow Vc(\vec{l_z}) = (w(c_1, l_z), w(c_2, l_z), ..., w(c_t, l_z)) \quad (1)$$

where it is the sum of all the concepts that all results contain under a concept lens. Here, each dimension of $w(c_1, l_z)$ corresponds to a separate UMBEL concept or supertype category and its weight. A similar method is used for calculating *vector of supertypes*. We use concept/supertype frequency as weight, i.e. if a concept does not occur in the concept lens, the value is 0. Generally term frequency, inverse document frequency ($tf \times idf$) is used for weighting. However, our studies show that the frequency, tf , works better $tf \times idf$. idf weights rare terms (or concepts) higher. This works well for retrieval, but not for similarity comparison as we compute the shared information between the lenses. In the same manner, results' snippets are combined to form a *vector of terms* of the concept lenses (eq. 2). Each dimension corresponds to a unique term and its weight. Again we use the term frequency as weight.

$$\sum_{z=1}^{m} l_z = \sum_{i=1}^{n} r_i t \rightarrow \vec{Vt}(l_z) = (w(t_1, l_z), w(t_2, l_z), ..., w(t_s, l_z)) \quad (2)$$

User Interests: For interactive personalization of the results, we need to capture the user's information need from the clicked results, which are then used for results re-ranking and query expansion. For this purpose, we track last *M* clicks of the user, *u*, and generate three types of vectors to represent the user's information need: vector of concepts (specific interests), vector of supertypes (broad interests), vector of terms (language model). In this case, vectors are extracted from the last *M* results clicks;

4.3 Re-organization of Concept Lenses

Dynamic adaptation of results to the local user choices is the most innovative personalization provided by our system. This allows dynamic results adaptation to local user choices, which moves conceptually relevant concept lenses to the top of the list. To achieve this, we compare the similarity of the selected concept lens to other concept lenses using the cosine similarity of concept, supertype and term vectors:

$$sim(l_1, l_2) = \vec{V_1}(l_1).\vec{V_2}(l_2) / |\vec{V_1}(l_1)||\vec{V_2}(l_2)| \quad (3)$$

where $sim(l_1, l_2) \in [0,1]$, numerator is the inner product of the vectors and the denominator is the multiplication of the vector magnitudes. We generate three similarity scores for each concept lens, namely *c_sim*, *s_sim* and *t_sim* according to their similarity to the selected concept lens, l_s . The concept similarity (*c_sim*) compares similarity of shared specific concepts, i.e. if lenses share more specific concepts, it is more likely that they are relevant. The supertype similarity (*s_sim*) computes shared broad concepts. For example, "mountain" and "lake" concept lenses have the same supertype category and they broadly related. Finally, the term similarity (*t_sim*) allows comparing language models of the lenses. This information can be noisy since different resources may share similar meanings but may use different terms. However, still term similarities can be used to guarantee some level of similarity between lenses.

Our evaluations on the benchmark dataset showed that the concept vector similarity of lenses provided the best precision @top N concept lenses compared to supertype and term similarities (see section 5). In addition, when different similarity scores were combined, precision was improved. In particular, when the influence of the *c_sim* was weighted higher than the *s_sim* and *t_sim*, the best precision @top N concept lenses

was obtained. Especially the best results were obtained when $\alpha = 2, \beta = 1$ and $\delta = 1$. If $sim(l, l_s) > 0.2$, the concept lens is suggested for exploration as shown in Figure 2.

$$sim(l, l_s) = \left(\alpha * c\_sim(l, l_s) + \beta * s\_sim(l, l_s) + \delta * t\_sim(l, l_s)\right) / \left((\alpha + \beta + \delta)\right) \quad (4)$$

Finally, concept lenses are re-ranked in decreasing $sim(l, l_s)$ order. By default, the selected lens came on top of the list since cosine similarity of a vector to itself is 1.

4.4 Results Re-ranking and Concept Lenses Suggestion

For results re-ranking, each result is represented with a vector of concepts, supertypes and terms: $\vec{Vc}(r) = (w(c_1, r), ..., w(c_x, r))$, $\vec{Vsc}(r) = (w(sc_1, r), ..., w(sc_w, r))$, $\vec{Vt}(r) = (w(t_1, r), ..., w(t_y, r))$. We apply results re-ranking in two cases: (a) when the user selects a concept lens from the results list for exploration and (b) when the user clicks onto a result (LOD resource) within a concept lens. In both cases, the re-ranked results are included in the context of the interacted concept lens. This allows in context results exploration thanks to the use of concept lenses. In case (a), we compare concept vector, of the selected concept lens (l_s) with the top *K* results using eqs. (3), (5);

$$\sum_{i=1}^{K} sim(\vec{Vc}(r_i), \vec{Vc}(l_s)) \quad (5)$$

We compare concept vectors since results matching at specific UMBEL concepts are more likely to be relevant compared to supertype or term similarities (we only have user's interest for a concept lens). In our experiments, *K*=100, for a scalable performance. Results, where $sim(\vec{Vc}(r), \vec{Vc}(l_s)) > \alpha$, are re-ranked in decreasing order and added into the interacted concept lens. $\alpha = 0$; any match was considered because of specific concept vectors comparison. Later, α can be determined experimentally.

In case (b), we use the click history of the user within the current search session to re-rank results. In particular, from the last *M* results clicks of the user, user's specific concept, supertype and term interests are represented as vectors. These vectors are compared with top *K* result vectors using eqs. (3), (6);

$$\sum_{i=1}^{K} \frac{\alpha * sim(\vec{Vc}(r_i), \vec{Vc}(u)) + \beta * sim(\vec{Vsc}(r_i), \vec{Vsc}(u)) + \delta * sim(\vec{Vt}(r_i), \vec{Vt}(u))}{\alpha + \beta + \delta} \quad (6)$$

where three similarity scores are combined for re-ranking of the results in decreasing order. Especially, $\alpha = 2$, $\beta = 1$ and $\delta = 1$ gives better results. Again, a threshold can be used to select relevant results conservatively, i.e. higher thresholds. If a relevant result belongs to another concept, then the concept lens is suggested for exploration.

4.5 Query Refinement Using Concept Labels

Query adaptation is applied in two cases: (i) when the user selects a concept lens from the results list for exploration and (ii) when the last two consecutive result clicks

share the same concept. In both cases, we assume that the user is interested in this concept and we expand the original query with the concept label that the user is interested. It is a simple approach, but works well since UMBEL categorizations provide very specific concept names and it can be used to clarify the meaning of the query with the user feedback. In both cases, more results are included in the context of the interacted concept lens, so that the user can explore more relevant results in context.

5 Evaluations

In traditional IR, there are public benchmarks for standardized evaluation and comparison (i.e. TREC). However, there are no standard evaluation benchmarks for semantic search evaluations [13]. Current semantic search methods are based on user-centered evaluation, which tend to be high-cost, non-scalable and difficult to repeat. [13] proposes to use TREC for cross-comparison between IR and ontology-based search models. They annotate TREC collections with instances of ontology concepts. However, it was found that only 20% TREC search topics have semantic matches in 40 public ontologies. Thus it can be difficult to apply this technique in many topic domains. Although a similar approach of [14] can be adopted, they rely on semantic annotation of *documents*. In our approach, we focus on categorization of LOD resources as the basis of the personalization and visualization rather than semantic annotations of documents. Thus, we created a benchmark dataset using LOD resources, which is available online for validation and comparison [21]. We measured personalized search efficacy using precision @top *M* concept lenses and @top *N* results. We focused on precision since our aim is to improve precision on the top results/lenses.

Dataset. For the experiment, we selected tourism domain. Because our aim is not just providing direct answers to a search query but to support results exploration with categorization and personalization. The tourism domain suits such data gathering and informational queries, since the user has a vague idea about queries and gradually refines queries to gather/explore more information. This scenario is also fits for WoD search, since developers usually explore WoD to gather data about a specific domain.

Our dataset is about "tourism in Killarney Ireland" and it was created as follows: One option was to use Sindice for dynamic querying. However, Sindice search results may change due to dynamic indexing. Thus, we decided to index a particular dataset for stable and comparative evaluations. First, we investigated popular search queries about the domain from Google search trends. Then, these queries were used to query WoD with Sindice to gather data about available URIs. Particularly ~500 URIs from DBpedia, GeoNames, Trip Advisor and ookaboo domain were selected. RDF descriptions of the URIs, their UMBEL/supertype categorizations were indexed offline by the proposed fuzzy retrieval model [8] to carry the experiments. Then, we selected 20 queries, which do not have a direct answer, i.e. navigational queries were not selected, such as "Killarney Victoria Hotel". Although 20 queries is a small sample set, such sizes have been used before to determine indicative results in semantic search [13] [14]. Top concept lenses and results returned by the queries, were manually assigned relevant or irrelevant. For non-adaptive baseline systems, we used the same dataset.

5.1 Personalization Time Overhead

Results categorization is applied offline during the indexing of LOD resources. Thus, we computed average time required to generate personalized results (i.e. lenses re-organization, results re-ranking and query expansion following a lens selection). For each query, average of 5 runs used. Results showed that personalized results were obtained within an average of 0.26 seconds compared to 0.16 of non-adaptive case. Our personalization is scalable thanks to complete client-side implementation. The results were run on Windows 7 computer, 2.2GHz CPU and 7.90GB RAM.

5.2 Performance of Personalization Strategies

In the experiments, personalization was performed after the user's concept lens selection following a query. To evaluate the efficiency of personalized lenses re-ranking, precision at top M concepts was measured, which was adopted by [6]. Precision at top M concepts is: $P@M=R@M / M$, where $R@M$ is the number of concept lenses which have been manually tagged relevant among top M concept lenses. For ambiguous concept lenses, if the majority of results under the lens were relevant, then we judge relevant. We use P@1, P@3, P@5, P@10 and P@15, since lazy users browse top concept lenses. The results in Figure 3 (left) show that for lenses re-ranking, lenses' concept vector similary provided the best precision compared to supertype and term vector similarities. When various similarity scores were combined, the precision was improved (in the experiment, influence of concept similarity is higher than others, e.g. $\alpha = 2, \beta = 1$ and $\delta = 1$ in eq. (8)). The best personalized lenses re-ranking was obtained when concept, supertype and term vector similarities were combined.

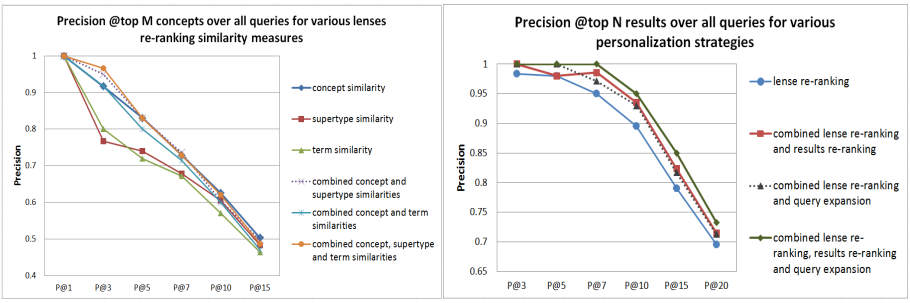


Fig. 3. Precision @top N concepts over all queries for lenses re-ranking similarity measures (left). Precision @top N results over all queries for various personalization strategies (right)

In a similar manner, we measured precision at top N results for different personalization strategies: $P@N = R@N / N$, where $R@N$ is the number of results which have been manually tagged relevant among top N results as shown in Figure 3 (right). The results showed that lenses re-ranking significantly improve precision @top N results. Combined lenses re-ranking with results re-ranking or query expansion improve lenses re-ranking performance. This also shows that personalized re-ranking of results

and query expansion with concept lens label work well. When all personalization strategies were combined, the best results were obtained, where 100% precision at P@3, P@5 and P@7 were obtained on the tested 20 queries.

5.3 Comparison with Non-adaptive Concept-Based Search and Ranked Lists

We compared personalized search performance against non-adaptive concept-based search and ranked list presentation. Here, the non-adaptive concept-based search present the results without adaptation to the user's selected concept lens, i.e. there is no lenses re-ordering, results re-ranking and query expansion. Whereas, the ranked result lists uses the original rank of the result and present it without categorization. First, we evaluated non-adaptive concept lenses ordering. Lenses can be ordered based on; (a) the minimum result rank within a lens, or (b) average of all results' ranks within it. Results in Figure 4 (left), show that both cases provide similar results. However, for the minimum rank order, P@1 is slightly better than the average rank. Thus, we used the minimum order for comparison with personalized lenses re-ranking. The personalized re-ordering of lenses significantly improved precision @top *M* concept lenses compared to the non-adaptive concept lenses as shown in Figure 4 (right). In a similar manner, we compared our personalized search on precision @top *N* results against non-adaptive concept-based search and traditional ranked list presentation (Figure 5). The results showed that our personalized search outperforms precision at all levels compared to non-adaptive concept-based search and traditional rank lists.

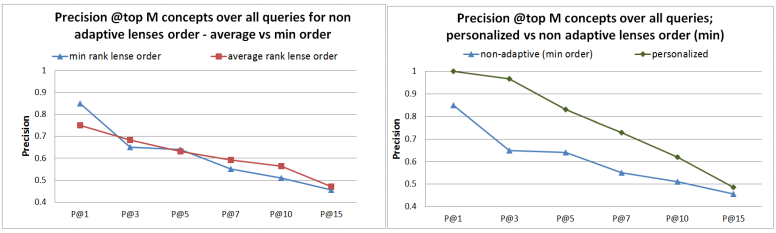


Fig. 4. Precision @top *M* concepts over all queries: Left; non-adaptive lenses ordering using minimum vs. average rank. Right; comparison of personalized vs. non-adaptive lenses ordering

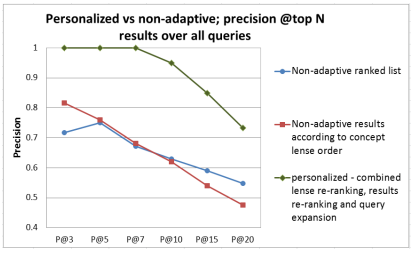


Fig. 5. Precision @top N results over all queries for personalized and non-adaptive search

6 Conclusions and Future Work

We introduced a novel personalized search and exploration mechanism for the Web of Data based on concept-based results categorization. In our approach, search results (LOD resources) are conceptually categorized to form *concept lenses*, which assist exploratory search/browsing. When the user selects a concept lens, results are immediately personalized; lenses are re-organized, more relevant results are included using results re-ranking and query expansion, as well as, relevant lenses are suggested for exploration. This is an innovative feature offered by our approach since it allows dynamic results adaptation to the user's local choices. Our personalization is privacy preserving, non-intrusive and scalable since it does not require user login and implemented at the client-side. Evaluations showed that the proposed approach significantly enhances precision compared to non-adaptive concept-based search and ranked list. In future, we will perform user studies to evaluate usability of our approach. In addition, data quality, trust and graph popularity can be considered in rankings.

Acknowledgments. This research is supported by the SFI part of EMPOWER research fellowship (07/CE/I1142) and part of the CNGL (www.cngl.ie) at University of Dublin.

References

1. Delbru, R., Campinas, S., Tummarello, G.: Searching Web Data: An Entity Retrieval and High-Performance Indexing Model. Journal of Web Semantics 10, 33–58 (2012)
2. D'Aquin, M., Motta, E., Sabou, M., Angeletou, S., Gridinoc, L., Lopez, V., Guidi, D.: Toward a New Generation of Semantic Web Applications. IEEE Intelligent Systems (2008)
3. Tummarello, G., Cyganiak, R., Catasta, M., Danielczyk, S., Delbru, R., Decker, S.: Sig.ma: Live views on the Web of Data. Journal of Web Semantics 8(4), 355–364 (2010)
4. Heim, P., Ertl, T., Ziegler, J.: Facet Graphs: Complex Semantic Querying Made Easy. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010, Part I. LNCS, vol. 6088, pp. 288–302. Springer, Heidelberg (2010)
5. Hogan, A., Umbrich, J., Harth, A., Cyganiak, R., Polleres, A., Decker, S.: An empirical survey of Linked Data conformance. Journal of Web Semantics 14, 14–44 (2012)
6. Ferragina, P., Gulli, A.: A Personalized Search Engine Based on Web-Snippet Hierarchical Clustering. In: International World Wide Web Conference (WWW), pp. 801–810 (2005)
7. Demo: `https://www.scss.tcd.ie/melike.sah/ESWC2013demo.swf`
8. Sah, M., Wade, V.: A Novel Concept-Based Search for the Web of Data Using UMBEL and a Fuzzy Retrieval Model. In: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 103–118. Springer, Heidelberg (2012)
9. Sah, M., Wade, V.: A Novel Concept-based Search for the Web of Data. In: I-SEMANTICS (2012)
10. Jansen, B.J., Booth, D.L., Spink, A.: Determining the User Intent of Web Search Engine Queries. In: International Conference on World Wide Web, pp. 1149–1150 (2007)

11. Ghorab, M.R., Zhou, D., O'Connor, A., Wade, V.: Personalised Information Retrieval: Survey and Classification. Journal of User Modeling and User-Adapted Interaction (to appear)
12. Sieg, Mobasher, B., Burke, R.: Web Search Personalization with Ontological User Profiles. In: International Conference on Information and Knowledge Management, CIKM (2007)
13. Fernandez, M., Lopez, V., Sabou, M., Uren, V., Vallet, D., Motta, E., Castells, P.: Using TREC for cross-comparison between classic IR and ontology-based search models at a Web scale. In: Workshop on Semantic Search (SemSearch 2009), WWW 2009 (2009)
14. Fernandez, M., Cantador, I., Lopez, V., Vallet, D., Castells, P., Motta, E.: Semantically enhanced Information Retrieval: An ontology-based approach. JWS 9(4), 434–452 (2011)
15. Janowicz, K., Raubal, M., Kuhn, W.: The semantics of similarity in geographic information retrieval. Journal of Spatial Information Sciences (2), 29–57 (2011)
16. Tous, R., Delgado, J.: A vector space model for semantic similarity calculation and OWL ontology alignment. In: Bressan, S., Küng, J., Wagner, R. (eds.) DEXA 2006. LNCS, vol. 4080, pp. 307–316. Springer, Heidelberg (2006)
17. Giunchiglia, F., Kharkevich, U., Zaihrayeu, I.: Concept Search. In: Aroyo, L., et al. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 429–444. Springer, Heidelberg (2009)
18. Mirizzi, R., Ragone, A., Di Noia, T., Di Sciascio, E.: Semantic wonder cloud: Exploratory search in DBpedia. In: Daniel, F., Facca, F.M. (eds.) ICWE 2010. LNCS, vol. 6385, pp. 138–149. Springer, Heidelberg (2010)
19. Tvarožek, M., Bieliková, M.: Factic: Personalized exploratory search in the semantic web. In: Benatallah, B., Casati, F., Kappel, G., Rossi, G. (eds.) ICWE 2010. LNCS, vol. 6189, pp. 527–530. Springer, Heidelberg (2010)
20. Shneiderman, B.: The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In: IEEE Symposium on Visual Languages (1996)
21. Benchmark: `https://www.scss.tcd.ie/melike.sah/tourismdataset.zip`
22. D'Amato, C., Fanizzi, N., Esposito, F.: A semantic similarity measure for expressive description logics. Convegno Italiano di Logica Computazionale, CILC (2005)

Combining a Co-occurrence-Based and a Semantic Measure for Entity Linking

Bernardo Pereira Nunes[1,2], Stefan Dietze[2], Marco Antonio Casanova[1], Ricardo Kawase[2], Besnik Fetahu[2], and Wolfgang Nejdl[2]

[1] Department of Informatics - PUC-Rio - Rio de Janeiro, RJ, Brazil
{bnunes,casanova}@inf.puc-rio.br
[2] L3S Research Center - Leibniz University Hannover - Hannover, Germany
{nunes,dietze,kawase,fetahu,nejdl}@l3s.de

Abstract. One key feature of the Semantic Web lies in the ability to link related Web resources. However, while relations within particular datasets are often well-defined, links between disparate datasets and corpora of Web resources are rare. The increasingly widespread use of cross-domain reference datasets, such as Freebase and DBpedia for annotating and enriching datasets as well as documents, opens up opportunities to exploit their inherent semantic relationships to align disparate Web resources. In this paper, we present a combined approach to uncover relationships between disparate entities which exploits (a) graph analysis of reference datasets together with (b) entity co-occurrence on the Web with the help of search engines. In (a), we introduce a novel approach adopted and applied from social network theory to measure the connectivity between given entities in reference datasets. The connectivity measures are used to identify connected Web resources. Finally, we present a thorough evaluation of our approach using a publicly available dataset and introduce a comparison with established measures in the field.

Keywords: Semantic connectivity, co-occurrence-based measure, linked data, data integration, link detection, semantic associations.

1 Introduction

The emergence of the Linked Data approach has led to the availability of a wide variety of structured datasets on the Web[1] which are exposed according to Linked Data principles [3]. However, while the central goal of the Linked Data effort is to create a well-interlinked graph of Web data, links are still comparatively sparse, often focusing on a few highly referenced datasets such as DBpedia[2], YAGO [28] and Freebase[3], while the majority of data exists in a rather isolated fashion. This is of particular concern for datasets which describe the same or potentially *related* resources or real-world *entities*. For instance, within the academic field, a wealth of potentially connected entities are

[1] http://lod-cloud.net/state
[2] http://dbpedia.org
[3] http://www.freebase.com

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 548–562, 2013.

described in bibliographic datasets and domain-specific vocabularies, while no explicit relationships are defined between equivalent, similar or connected resources [8].

Furthermore, knowledge extraction and Named Entity Recognition (NER) tools and environments such as GATE [5], DBpedia Spotlight[4], Alchemy[5], AIDA[6] or Apache Stanbol[7] are increasingly applied to automatically generate structured data (entities) from unstructured resources such as Web sites, documents or social media. For example, such automatically generated data may provide some initial classification and structure, such as the association of terms with entity types defined in a structured RDF schema (as in [22]). However, entities extracted via Natural Language Processing (NLP) techniques usually are noisy, ambiguous and lack sufficient semantics. Hence, identifying links between related entities within a particular dataset, as well as with pre-existing knowledge, serves three main purposes (a) enrichment, (b) disambiguation and (c) data consolidation. Often, dataset providers aim at *enriching* a particular dataset by adding links (*enrichments*) to comprehensive reference datasets. Current interlinking techniques usually resort to mapping entities which refer to the same resource or real-world entity, e.g., by creating `owl:sameAs` references between an extracted entity representing the city "Berlin" with the corresponding Freebase and Geonames[8] entries.

However, additional value lies in the detection of related entities within and across datasets, e.g., by creating `skos:related` or `so:related` references between entities that are to some degree connected [10,14]. In particular, the widespread adoption of reference datasets opens opportunities to analyse such reference graphs to detect the *connectivity*, i.e., the *semantic association* [2,26] between a given set of entities. However, uncovering these connections would require the assessment of very large data graphs in order to (a) identify the paths between given entities and (b) measure their meaning with respect to a definition of semantic connectivity.

In this paper, we present a general-purpose approach that combines a co-occurrence-based and a semantic measure to uncover relationships between entities within reference datasets in disparate datasets. Our novel semantic connectivity score is based on the Katz index [16], a score for measuring relatedness of actors in a social network, which has been adopted and expanded to take into account the semantics of data graphs, while the co-occurrence-based method relies on Web search results retrieved from search engines. Finally, we evaluate the approach using the publicly available USAToday corpus and compare our entity connectivity results with related measures.

The remainder of this paper is structured as follows. Section 2 discusses previous related work in the field. Section 3 presents the use case scenario that motivated our approach. Section 4 presents our entity connectivity approach. Section 5 and Section 6 show the evaluation strategies and their results. Finally, Section 7 summarizes our contributions and discusses future work.

[4] `http://dbpedia.org/spotlight`
[5] `http://www.alchemyapi.com`
[6] `http://adaptivedisclosure.org/aida/`
[7] `http://incubator.apache.org/stanbol`
[8] `http://www.geonames.org`

2 Related Work

Lehmann et al. [17] introduces RelFinder, which shows semantic associations between multiple entities from a RDF dataset, based on a breadth-first search algorithm, that is responsible for finding all related entities in the tripleset. Contrasting with RelFinder, Seo et al. [25] proposed the OntoRelFinder that uses a RDF Schema for finding semantic associations between two entities through its class links. Scarlet [23,24] is another approach that relies on different schemas to identify relationships between entities.

Han et al. [15] proposes a slightly different approach. Instead of finding connections between two given entities, they expect to find the entities that are most connected, with respect to a given relationship and entity. This approach is interesting since it throws another perspective on the problem that we consider. However, they look for connected entities by means of a known relationship, while we aspire to uncover such connections between known entities.

Anyanwu et al. [1] present the SemRank, a customizable query framework that allows different setups for ranking methods, resulting in different perspectives for the same query. Thus, given two entities, depending on the setup the search results vary from more traditional (e.g. common connections or closest paths between entities) to less traditional (e.g. longer paths). In our approach, we consider both short and long paths to determine connectivity between two entities and Web resources.

Work from Leskovec et al. [18] presents a technique suggesting positive and negative relationships between people in a social network. This notion is also addressed in our method, but we take into account the path length. The longer is the path, the smaller is its contribution to the score.

The problem of discovering relationships between entities was also addressed by Damljanovic et al. [6] in Open Innovation scenarios, where companies outsource tasks on a network of collaborators. Their approach exploits the links between entities extracted from both the user profiles and the task descriptions in order to match experts and tasks. For this task, they use reference datasets and distinguish between entities as hierarchical and transversal. Following her approach, we distinguish between both relations types, although we focus on transversal relations.

Related work in the field of recommender systems includes the work by Passant [20], which presents a linked data semantic distance measure (LDSD) for music recommendation, by taking mainly into account incoming and outgoing links as well as indirect links between resources (i.e., songs and singers) to determine a recommendation score, used for recommending both direct and lateral music. In later work [19], he introduces a filtering step, by removing properties between resources that are not meaningful in the music context. Work on movie recommendation by Souvik et al. [7] considers an approach based on object features in order to improve movie recommendation, by using several similarity functions that deal with nominal, boolean and numeric features. Furthermore, they also use a linear regression method to assign weights for each feature type. Although this method presents good results, they do not consider semantic connections to uncover latent features.

Fang et al. [9] introduces the REX system, which computes a ranked list of entity pairs to describe entity relationships. The graph structure is decomposed for an entity pair resulting in unique graph patterns and ranks, where these patterns are matched

according to a measure of interestingness, based on the traditional random walk algorithm and the patterns found between an entity pair.

Sieminski [27] presents a method to measure the semantic similarity between texts on the Web, which consists of a modified *tf-idf* model and semantic analysis that makes use of WordNet structure. However, unlike his work, we explore the connections given by transversal properties in order to uncover latent connections between texts, rather than to explore similarity between them.

From the approaches outlined, we combine different techniques to uncover connections between disparate entities, which allows us to exploit the relationships between entities to identify connected Web resources.

3 Motivation

In this section we describe an example originating from actual Web information integration problems to illustrate the motivation of our work on discovering *latent semantic relationships* through its *semantic relations*.

The example below shows two descriptions of documents extracted from the USAToday corpus. Note that, the underlined terms refer to the recognised entities in each document derived from an entity recognition and enrichment process.

(i) The Charlotte Bobcats could go from the NBA's worst team to its best bargain.

(ii) The New York Knicks got the big-game performances they desperately needed from Carmelo Anthony and Amar'e Stoudemire to beat the Miami Heat.

Although both documents are clearly related to Basketball/Sports topics, linguistic and statistical approach would struggle to point out that both documents are connected. First, both textual descriptions are rather short and lack sufficient contextual information what makes it harder for purely linguistic or statistical approaches to detect their connectivity. Second, in this particular case, there are no significant common words between the documents. Usually, statistical and linguistic approaches are particularly suitable for cases where large amounts of textual content is available to detect the relationships between Web resources. In particular, some common terminology is required for detecting similarities between Web resources.

On the other hand, these challenges can be partially overcome by taking advantage of structured background knowledge to disambiguate and enrich the unstructured textual information. The example shows two documents, each associated with a particular entity, where the term *Charlotte Bobcats* was enriched with the entity `http://dbpedia.org/resource/Charlotte_Bobcats` in the document (i) and the term *Carmelo Anthony* was enriched with the entity `http://dbpedia.org/resource/Carmelo_Anthony` in the document (ii). Thus, analysing the DBpedia graph uncovers a connection between *Charlotte Bobcats* and *Carmelo Anthony* (being a basketball team and player, respectively) and hence allows us to establish a connection between the entities and their connected Web resources. Specifically, both entities are connected through the path: *Charlotte Bobcats* ↔ *Eastern Conference (NBA)* ↔ *New York Knicks* ↔ *Carmelo Anthony*, where the intermediary entities uncover a connection between *Charlotte Bobcats* and *Carmelo Anthony*.

4 Approach

In this section, we introduce two novel measures for entity interlinking, a semantic graph-based connectivity score and one which utilises co-occurrence on the Web. Both detect complementary relationships between entities as results show in Section 6.

4.1 Semantic Connectivity Scores (SCS)

In this section, we define a semantic connectivity score between entities, based on a reference graph that describes entities and their relations. Similar to Damljanovic et al. [6], we distinguish between *hierarchical* and *transversal* relations in a given graph. Typical hierarchical properties in RDF graphs are, for instance, `rdfs:subclassOf`, `dcterms:subject` and `skos:broader`, and usually serve as an indicator for similarity between entities. In contrast, transversal properties do not indicate any classification or categorisation of entities, but describe non-hierarchical relations between entities which indicate a form of connectivity independent of their similarity.

To illustrate the semantic connectivity, we refer to the pair of entities "Jean Claude Trichet" and "European Central Bank", which have no equivalence or taxonomic relation, but have a high connectivity according to transversal properties. For example, the "European Central Bank" is linked to the entity "President of the European Central Bank" through the RDF property `http://dbpedia.org/property/leaderTitle` that, for its part, links to "Jean Claude Trichet" through the RDF property `http://dbpedia.org/property/title`.

Let R be a reference triple set and G be the associated undirected graph, in the sense that the nodes of G correspond to the individuals occurring in R and the edges of G correspond to the properties between individuals defined in R. From this point on, we will refer to the individuals occurring in R as *entities*.

We define the *semantic connectivity score* (SCS) between a pair of entities (e_1, e_2) in G as follows:

$$SCS(e_1, e_2) = \sum_{l=1}^{\tau} \beta^l \cdot |paths^{<l>}_{(e_1,e_2)}| \tag{1}$$

where $|paths^{<l>}_{(e_1,e_2)}|$ is the number of transversal paths between the entities e_1 and e_2 of length l, τ is the maximum length of paths considered (in our case $\tau = 4$, as explained in more details below), and $0 < \beta \leq 1$ is a positive damping factor. The damping factor β^l is responsible for exponentially penalizing longer paths. The smaller this factor, the smaller the contribution of longer paths to the final score. Obviously, if the damping factor is 1, all paths will have the same weight independently of length. In previous experiments, we observed that $\beta = 0.5$ presented better results in terms of precision [21].

The semantic connectivity score between entities is a variation of the Katz index [16] introduced to estimate the relatedness of actors in a social network. We introduced a number of derivations to improve its applicability to large graphs and to reflect the added semantics provided by labelled edges in RDF graphs, as opposed to the limited semantics of edges in a social network. A detailed discussion of the advantages and limitations of our approach is provided in Section 7.

As one main adaptation of Katz, we exploit the semantics of edges in a given data graph by excluding hierarchical properties from our connectivity score computation. As defined earlier, connectivity is indicated by transversal properties. Currently, no further distinction between property types has been introduced into our formula, though we explicitly envisage such an adaptation. However, given the vast amount of property types in datasets such as DBpedia, a distinction at the general and domain-independent level is computationally too expensive and therefore does not scale. Instead, we particularly suggest the adaptation of our formula to specific domains or entity types, which allows the consideration of more fine-grained semantics provided by distinct property types.

In addition, we opted for an undirected graph model in order to reduce computational complexity, since a property is often found in its inverse form (e.g. fatherOf/sonOf) [13]. While most current entity interlinking techniques apply their approaches to a restricted set of entity types to allow some sort of tailoring and, as consequence, more precise results, our experiments in Section 5 show that even our fairly generic score produces useful and promising results, which can be improved by means of domain-specific adaptations.

As the semantic score is based on the number of paths and distances (length of a path) between entities, SCS considers only paths with a maximum length ($\tau = 4$), as also adopted in [9]. This maximum length was identified by investigating the semantic score behaviour for edge distances ranging from 1 to 6, as detailed below.

In our experiments, we randomly selected 200 entity pairs and computed the semantic connectivity score (SCS) (see Eq. 1) for the aforementioned path length range (see Figure 1a). As expected, the average number of paths grows exponentially with the distance (i.e. the path length), see Figure 1a.

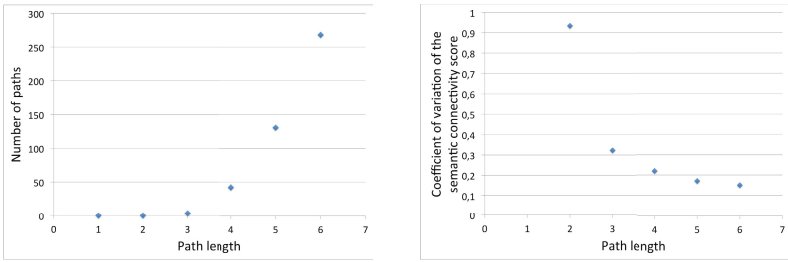
Thus, as in the small world assumption [29], beyond a certain path length, every node pair is likely to be connected. However, as opposed to the small world assumption that people are interlinked through a maximum distance of 6 connections, we found that for interlinking entities this number is lower, approximately by two degrees. This decision is backed up according to several experiments, detailed below.

After computing all entity pairs for different path lengths, we evaluated the coefficient of variation of the semantic score, $C_v = \sigma/\mu$, where, for a given length, σ is the standard deviation of the number of paths and μ is the mean number of paths. This coefficient is used to measure the spread of the semantic score distribution, taking into account an upper bound path length (see Figure 1b).

From the behaviour of the curve in Figure 1b, it is apparent that the contribution of paths with distances greater than 4 edges is low. Also as expected, the average running time to compute the path grows exponentially with the distance. Hence, including longer path lengths increases significantly the computational costs, while producing only minimal gains in performance. Thus, we obtain the best balance between performance and informational gain to the semantic score. That is, we minimise the path length considered, while maximise the contribution in the overall score.

4.2 Co-occurrence-Based Measure (CBM)

We introduce in this section a co-occurrence-based measure between entities that relies on an approximation of the number of existing Web pages that contain their labels.



(a) (b)

Fig. 1. Maximum path length analysis. Figure (a) shows the number of paths with respect to length and (b) shows the gain of information when considering different path lengths.

For example, we estimate the CBM score of a pair of entities by submitting queries (such as "Jean Claude Trichet" + "European Central Bank") to a search engine and retrieving the total number of search results that contain the entity labels in their text body. Thus, we define the CBM score of a pair of entities e_1 and e_2 as follows:

$$CBM(e_1, e_2) = \begin{cases} 0, & \text{if } count(e_1) = 0 \text{ or } count(e_2) = 0 \\ 1, & \text{if } count(e_1) = count(e_2) = count(e_1, e_2) = 1 \\ \dfrac{Log(count(e_1, e_2))}{Log(count(e_1))} \cdot \dfrac{Log(count(e_1, e_2))}{Log(count(e_2))}, & \text{otherwise} \end{cases} \tag{2}$$

where $count(e_i)$ is the number of Web pages that contain an occurrence of the label of entity e_i, and $count(e_1, e_2)$ is the number of Web pages that contain occurrences of the labels of both entities. Note that $count(e_1, e_2)$ is a non-negative integer always less than or equal to $count(e_i)$, for $i = 1, 2$. Hence, the final score is already normalised to $0 \leq CBM(e_1, e_2) \leq 1$.

There are other similar approaches to quantify the relation between entities, such as Pointwise Mutual Information (PMI)[4] and Normalised Google Distance (NGD)[12]. However, they take into account the joint distribution and the probability of their individual distributions, which requires to know a priori the total number of Web pages searched by a search engine.

To illustrate the co-occurrence-based score (CBM), consider the values $count(e_1) = count(e_2) = count(e_1, e_2)$, meaning that all occurrences of e_1 and e_2 appear together. In this case, the resulting co-occurrence-based score is 1, disregarding the number of search results.

For example, having $count(e_1) = count(e_2) = count(e_1, e_2) = 10$ or $count(e_3) = count(e_4) = count(e_3, e_4) = 1000$, would result in the same score. Evidently, if we would consider the probabilities, as in PMI or NGD, the latter case would get a higher score. Nevertheless, since we are not interested in disjoint comparisons, e.g., $CBM(e_1, e_2)$ against $CBM(e_3, e_4)$, we do not need to estimate the total number of pages, neither include it in the formula.

4.3 Towards a Combined Measure

As shown in previous work [21], although there is an overlap between the semantic and co-occurrence based approaches, some relationships cannot be uncovered by co-occurrence methods or by semantic methods alone. Thus, given that the results from SCS and CBM are seen as complementary, one conclusion is to combine them, which provides the advantage of scalability at discovering entity connections, where CBM would be used as a default approach, and SCS could be employed as an extensive search for finding latent connections in the resulting set of entity pairs deemed unconnected according to CBM, see Eq. 3.

$$\alpha_{CBM+SCS}(e_i, e_j) = \begin{cases} CBM(e_i, e_j), & \text{if } CBM(e_i, e_j) > 0 \\ SCS(e_i, e_j), & \text{otherwise} \end{cases} \tag{3}$$

where e_i and e_j are entities and $i \neq j$.

5 Evaluation Method

5.1 Dataset

The dataset for assessing entity connectivity consists of a set of 40,000 document pairs randomly selected from the USAToday news Website[9], where each document contains a title and a summary as textual content. The summary of each document has on average 200 characters. The corpus was annotated using DBpedia Spotlight[10] which resulted in approximately 80,000 entity pairs.

5.2 Gold Standard

Given the lack of benchmarks for validating latent relationships between entities, we created a gold standard using CrowdFlower[11], a crowdsourcing platform. To ensure a sufficient quality of the results, we required each user to pass through a set of tests where correct answers were known already, what allowed us to filter out poor assessors. In this way, we were able to avoid relevance judgements from untrusted workers. Moreover, as our corpus is focused on American news, we restrict the assessment only to workers located in the United States.

Thus, in order to construct the gold standard, we randomly selected 1000 entity pairs and 600 document pairs to be evaluated. The evaluation process consisted of a questionnaire in a 5-point Likert scale model where participants are asked to rate their agreement of the suggested semantic connection between a given entity pair. Additionally, we inspected participants' expectations regarding declared connected entities. In this case, presenting two entities deemed to be connected, we asked participants if such connections were expected (from *extremely unexpected* to *extremely expected* in the Likert scale).

[9] http://www.usatoday.com
[10] http://spotlight.dbpedia.org/
[11] https://www.crowdflower.com/

The collected judgements provided a gold standard for the analysis of our techniques. Note that in the case of this work, additional challenges are posed with respect to the gold standard, because our semantic connectivity score is aimed at detecting possibly unexpected relationships which are not always obvious to the user. To this end, a gold standard created by humans provides an indication of the performance of our approach with respect to precision and recall, but it may lack appreciation of some of our found relationships (see Section 6.2 for a detailed discussion).

5.3 Evaluation Methods

We also present a comparison of our approach against competing methods which measure connectivity via co-occurrence-based metrics to detect entity connectivity. In this evaluation we compared the performance of CBM against SCS and a third method (Explicit Semantic Analysis (ESA)) that is based on statistical and semantic methods.

Specifically, ESA [11] measures the relatedness between Wikipedia concepts by using a vector space model representation, where each vector entry is assigned using the *tf-idf* weight between the entities and its occurrence in the corresponding Wikipedia article. The final score is given by the cosine similarity between the weighted vectors. Note that ESA can be applied to measure any kind of corpora, not just Wikipedia concepts.

5.4 Evaluation Metrics

We measure the performance of the entity connectivity using the standard metrics of precision (P), recall (R) and $F1$ measure. Note that in these metrics, as relevant entity pairs, we consider those that were marked in the gold standard (gs) as connected according to the 5-point Likert Scale (*Strongly Agree* & *Agree*).

(P) is defined as the ratio of the set of retrieved entity pairs that have relevant uncovered connections over the set of entity pairs that have connections, see Eq. (4).

$$P = \frac{|\mu^{\tau}_{retrieved} \cap \mu_{relevant}|}{|\mu^{\tau}_{retrieved}|} \tag{4}$$

where $\mu_{relevant}$ is the set of retrieved entity pairs that are relevant and $\mu^{\tau}_{retrieved}$ is the set of retrieved connections that has a semantic connectivity score greater than a given threshold (τ). The threshold used in our experiments is shown in Section 6).

The recall measure is the ratio of the set of the retrieved entity pairs (R) that have relevant uncovered connections over all relevant connected entity pairs according to the gold standard, see Eq. (5).

$$R = \frac{|\mu^{\tau}_{retrieved} \cap \mu_{relevant}|}{|\mu_{relevant_{(gs)}}|} \tag{5}$$

where $\mu_{relevant_{(gs)}}$ is the set of all relevant entity pairs.

Finally, $F1$ measure shows the balance between precision and recall, and is computed as $F1 = 2 \cdot \frac{P \cdot R}{P+R}$.

6 Results

For each method described in the Sections 4 and 5, we present the results on their ability to discover latent connections over the entities. Furthermore, we also present an in depth-analysis of their shortcomings and advantages for discovering connections between entities.

6.1 Entity Connectivity Results

Table 1 shows the results obtained by the questionnaire and used as gold standard for the entity connectivity. The results are presented in a 5-point Likert scale of agreement ranging from *Strongly Agree* to *Strongly disagree*.

Table 1. Number of entity-pairs in each category (5-point Likert scale) in gold standard

Strongly Agree	Agree	Undecided	Disagree	Strongly Disagree
63	178	127	227	217

In Figure 2, we report the performance for the co-occurrence-based score (CBM), Explicit Semantic Analysis (ESA) and our proposed adaptation of the Katz score (SCS). We considered as relevant all the entity pairs which had relevance judgements as *Strongly Agree* and *Agree*, and scores greater than a threshold. Since our task is to uncover latent relationships between entities rather than ranking them, we set the threshold to 0 (i.e. we include all results), but for some tasks we might want to raise this, e.g. for ranking or recommending.

According to Figure 2, SCS performs better in terms of precision whereas CBM achieves highest recall value. SCS and CBM present only minimal differences with respect to precision and recall, while ESA has the lowest values for all metrics.

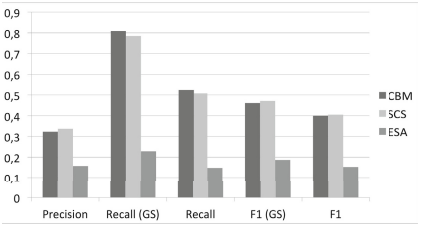


Fig. 2. P/R/F1 measure according to the gold standard (GS) amongst methods

In addition to performance, we are also interested in the agreement between the methods. Identifying missed and detected relationships amongst all measures provides an indicator of their complementarity. In Table 2 we present a pairwise comparison of methods where we show the ratio of connections that are found by one method and missed by another. It is notable that CBM and SCS capture most of the connections, even though CBM misses 3.1% and 11.2%, and SCS misses 9.5% and 12.3% for *Strongly Agree* and *Agree* respectively.

Table 2. Ratio of connections detected by each method, according to the gold standard

	CBM (*not* in SCS)	CBM (*not* in ESA)	SCS (*not* in CBM)	SCS (*not* in ESA)	ESA (*not* in CBM)	ESA (*not* in SCS)
Strongly Agree	**9.5%**	**76%**	**3.1%**	**71%**	**7.9%**	**9.5%**
Agree	**12.3%**	**63.4%**	**11.2%**	**60.1%**	**8.9%**	**6.7%**
Undecided	9.4%	60.6%	6.3%	59.8%	5.5%	7.9%
Disagree	15.0%	63.0%	7.1%	53.3%	7.1%	5.3%
Strongly Disagree	18.4%	63.1%	51.6%	4.6%	4.6%	6.9%

Besides the missed connections, we also take into account the expectedness of a connection between entity pairs. The expectedness shows how well established the connection is: an unexpected connection would be a relevant inferred indirect link between the entities. Thus, unexpectedness can be interpreted as a creation of novel links between entities. We see that SCS uncovers 25% of the unexpected connections, while CBM uncovers 16%. For this task, ESA was not able to uncover any new connections.

6.2 Results Analysis

In this section, we provide a detailed analysis of the results. The analysis is guided by the initial aims of our work on discovering latent connections between entities within a data graph (at varying path lengths), rather than competing with well established methods such as co-occurrence-based approaches widely deployed by search engines. To this end, the results of the listed approaches are complementary, where each of the approaches is able to establish unique entity connections.

In Figure 3, we show the agreement of entity pair ranking retrieved by SCS compared with CBM. The entity pair ranking follows an expected decline, where most connections are found at high ranks, whereas only a few are found at very low ranks.

As we can see in Figure 3, for the topmost rank of co-occurrence-based entity pairs, 225 of them have a semantic connection. Ideally, since these pairs are ranked in the top position, we expected to find a semantic connection between all of them. Arguably,

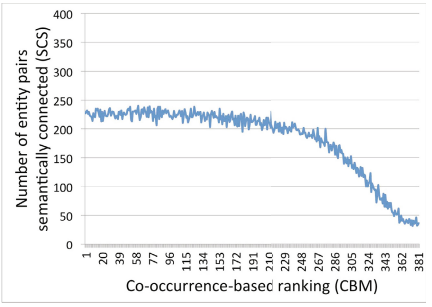


Fig. 3. The *x*-axis represents the ranking position *x* of entity pairs according to the CBM rankings. The *y*-axis represent the number of entity pairs ranked at *xth* position that have a semantic connection according to our connectivity threshold.

Table 3. Kendall tau and Jaccard-index between SCS and CBM entity rankings

	k@2		**k@5**		**k@10**	
Dataset	Kendall tau	Jaccard-index	Kendall tau	Jaccard-index	Kendall tau	Jaccard-index
USAToday	0.40	0.09	0.47	0.19	0.52	0.21

the dependency rank-position to semantic connection should follow the trend where the lower the rank position, the higher the number of semantic connected entity pairs. In this sense, we can estimate which items have some missing relations. This is the first step in the task of actually discovering the missing relations. By observing the missing semantic ranked pairs on the *x*-axis, we can identify which entities miss some connection induced by the co-occurrence-based score (the problem introduced on Section 3). It is worth noting that, after the 260th rank position in the *x*-axis, the behaviour of the curve is in line with our expectations, i.e., the lower the correlation induced by the co-occurrence-based score, the lower that induced by the semantic connectivity score.

To show the complementarity between CBM and SCS, we used the Kendall tau rank correlation coefficient to assess the agreement of the entity ranks induced by the semantic connectivity score based on the DBpedia graph against the entity ranks induced by CBM. Table 3 shows the results.

As we can see from Table 3, the overlap between the rankings is not high. However, as our previous evaluation with the gold standard shows, this indicates that the scores induce different relationships between entities. The CBM score induces a relationship that reflects the overall co-occurrence of entities in the Web, whereas the semantic connectivity score mirrors the DBpedia graph.

Thus, as shown in Table 4, the CBM+SCS is the best performing approach compared to the other methods for the task of entity connectivity. Moreover, when comparing the F1 results from the CBM+SCS and SCS, we achieve significantly different results for p-value = 0.04 with 95% confidence.

Table 4. P/R/F1 measures according to gold-standard and amongst methods

	CBM	SCS	ESA	CBM+SCS
Precision	0.32	0.34	0.16	0.34
Recall (GS)	0.81	0.78	0.23	0.90
Recall	0.52	0.51	0.15	0.58
F1 (GS)	0.46	0.47	0.19	0.50
F1	0.40	0.41	0.15	0.43

We would also like to point out the challenges posed by our approach on creating a gold standard. As mentioned previously, while our work aims at detecting semantic entity connections beyond traditional co-occurrences, this results in connections which might be to some extent unexpected yet correct, according to background knowledge (such as DBpedia in our case). Hence, using a manually created gold standard, though being the only viable option, necessarily impacts the precision values for our work in a negative way, as correct connections might have been missed by the evaluators. This has been partially confirmed by the large number of detected co-occurrences which

were marked as *undecided* by the users, where manual inspection of samples in fact confirmed a positive connection. This confirms that in a number of cases, connections were not necessarily incorrect but simply unknown to the users. Thus, we believe that a more thorough evaluation providing the evaluators with information on how a connection emerged, by showing all properties and entities that are part of a path greater than one, would give us more reliable judgements.

An example found in our evaluation is between the politicians "Barack Obama" and "Olympia Snowe", where the first is the current US president and the latter is one of the current senior US senators. Although the evaluators did not identify a connection between them, our semantic connectivity approach found several paths with length 2 or more. Additionally, they are related via several topics in real life, which confirms the validity of the paths found by our approach. For instance, this information could be exploited by news Websites for improving the user experience on finding related topics or news.

7 Discussion and Outlook

We have presented a general-purpose approach to discover relationships between entities, utilising structured background knowledge from reference graphs as well as co-occurrence of entities on the Web. To compute entity connectivity, we first introduced a semantic-based entity connectivity approach (SCS), which adapts a measure from social network theory (Katz) to data graphs, in particular Linked Data. We were able to uncover 14.3% entity connections not found by the state of the art method described here as CBM. While using a combination of CBM+SCS, we achieved a F1 measure of 43% for entity connectivity.

Our experiments show that SCS enables the detection of entity relationships that a priori linguistic and co-occurrence approaches would not reveal. Contrary to the latter, SCS relies on semantic relations between entities as represented in structured background knowledge, captured in reference datasets.

While both approaches (CBM and SCS) produce fairly good indicators for entity and document connectivity, an evaluation based on Kendall's tau rank correlation showed that the approaches differ in the relationships they uncover [21]. A comparison of agreement and disagreement between different methods revealed that both approaches are complementary and produce particularly good results in combination with each other. The semantic approach is able to find connections between entities that do not necessarily co-occur in documents (found on the Web), while the CBM tends to emphasise entity connections between entities that are not necessarily strongly connected in reference datasets. Thus, a combination of our semantic approach and traditional co-occurrence-based measures provide promising results for detecting related entities.

Despite the encouraging results, one of the key limitations of our Katz-based measure is the limited consideration of edge semantics in its current form. At the moment, property types are distinguished only at a very abstract level, while valuable semantics about the meaning of each edge (i.e., each property) is left unconsidered during the connectivity computation. We are currently investigating approaches to take better advantage of the semantics of properties in data graphs.

Another issue faced during the experimental work is related to the high computational demands when applying our approach to large-scale data, which restricted our experiments to a limited dataset. In particular, the combination of traditional measures with our approach could help in improving performance, for instance, by computing our semantic connectivity only between entity pairs deemed unconnected by traditional measures. In addition, reducing the gathering of paths to a limited set of nodes ("hub nodes") might help in further improving scalability.

References

1. Anyanwu, K., Maduko, A., Sheth, A.: Semrank: Ranking complex relationship search results on the semantic web. In: Proceedings of the 14th International Conference on World Wide Web, WWW 2005, pp. 117–127. ACM, New York (2005)
2. Anyanwu, K., Sheth, A.: p-queries: Enabling querying for semantic associations on the semantic web. In: Proceedings of the 12th International Conference on World Wide Web, Budapest, Hungary, pp. 690–699. ACM Press, New York (2003)
3. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. Int. J. Semantic Web Inf. Syst. 5(3), 1–22 (2009)
4. Church, K.W., Hanks, P.: Word association norms, mutual information, and lexicography. Comput. Linguist. 16(1), 22–29 (1990)
5. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In: Proc. of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL 2002), Philadelphia, pp. 168–175 (July 2002)
6. Damljanovic, D., Stankovic, M., Laublet, P.: Linked data-based concept recommendation: Comparison of different methods in open innovation scenario. In: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 24–38. Springer, Heidelberg (2012)
7. Debnath, S., Ganguly, N., Mitra, P.: Feature weighting in content based recommendation system using social network analysis. In: Proceedings of the 17th International Conference on World Wide Web, WWW 2008, pp. 1041–1042. ACM, New York (2008)
8. Dietze, S., Yu, H.Q., Giordano, D., Kaldoudi, E., Dovrolis, N., Taibi, D.: Linked education: Interlinking educational resources and the web of data. In: Ossowski, S., Lecca, P. (eds.) SAC, pp. 366–371. ACM (2012)
9. Fang, L., Sarma, A.D., Yu, C., Bohannon, P.: Rex: Explaining relationships between entity pairs. Proc. VLDB Endow. 5(3), 241–252 (2011)
10. Ferrara, A., Nikolov, A., Scharffe, F.: Data linking for the semantic web. Int. J. Semantic Web Inf. Syst. 7(3), 46–76 (2011)
11. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: Proceedings of the 20th International Joint Conference on Artifical Intelligence, IJCAI 2007, pp. 1606–1611. Morgan Kaufmann Publishers Inc., San Francisco (2007)
12. Gligorov, R., ten Kate, W., Aleksovski, Z., van Harmelen, F.: Using google distance to weight approximate ontology matches. In: Proceedings of the 16th International Conference on World Wide Web, WWW 2007, pp. 767–776. ACM, New York (2007)
13. Graves, A., Adali, S., Hendler, J.: A method to rank nodes in an rdf graph. In: Bizer, C., Joshi, A. (eds.) Proceedings of the Poster and Demonstration Session at the 7th International Semantic Web Conference (ISWC 2008), Karlsruhe, Germany, October 28. CEUR Workshop Proceedings, vol. 401. CEUR-WS.org (2008)

14. Halpin, H., Hayes, P.J., McCusker, J.P., McGuinness, D.L., Thompson, H.S.: When owl:sameAs isn't the same: An analysis of identity in linked data. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010, Part I. LNCS, vol. 6496, pp. 305–320. Springer, Heidelberg (2010)
15. Han, Y.-J., Park, S.-B., Lee, S.-J., Park, S.Y., Kim, K.Y.: Ranking entities similar to an entity for a given relationship. In: Zhang, B.-T., Orgun, M.A. (eds.) PRICAI 2010. LNCS, vol. 6230, pp. 409–420. Springer, Heidelberg (2010)
16. Katz, L.: A new status index derived from sociometric analysis. Psychometrika 18(1), 39–43 (1953)
17. Lehmann, J., Schüppel, J., Auer, S.: Discovering unknown connections - the dbpedia relationship finder. In: Auer, S., Bizer, C., Müller, C., Zhdanova, A.V. (eds.) CSSW. LNI, vol. 113, pp. 99–110. GI (2007)
18. Leskovec, J., Huttenlocher, D., Kleinberg, J.: Predicting positive and negative links in online social networks. In: Proceedings of the 19th International Conference on World Wide Web, WWW 2010, pp. 641–650. ACM, New York (2010)
19. Passant, A.: dbrec — music recommendations using DBpedia. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010, Part II. LNCS, vol. 6497, pp. 209–224. Springer, Heidelberg (2010)
20. Passant, A.: Measuring semantic distance on linking data and using it for resources recommendations. In: AAAI Spring Symposium: Linked Data Meets AI. AAAI (2010)
21. Pereira Nunes, B., Kawase, R., Dietze, S., Taibi, D., Casanova, M.A., Nejdl, W.: Can entities be friends? In: Rizzo, G., Mendes, P., Charton, E., Hellmann, S., Kalyanpur, A. (eds.) Proceedings of the WoLE Workshop in Conjuction with the 11th International Semantic Web Conference, vol. 906, pp. 45–57. CEUR-WS.org (November 2012)
22. Risse, T., Dietze, S., Peters, W., Doka, K., Stavrakas, Y., Senellart, P.: Exploiting the social and semantic web for guided web archiving. In: Zaphiris, P., Buchanan, G., Rasmussen, E., Loizides, F. (eds.) TPDL 2012. LNCS, vol. 7489, pp. 426–432. Springer, Heidelberg (2012)
23. Sabou, M., d'Aquin, M., Motta, E.: Exploring the semantic web as background knowledge for ontology matching. In: Spaccapietra, S., Pan, J.Z., Thiran, P., Halpin, T., Staab, S., Svatek, V., Shvaiko, P., Roddick, J. (eds.) Journal on Data Semantics XI. LNCS, vol. 5383, pp. 156–190. Springer, Heidelberg (2008)
24. Sabou, M., d'Aquin, M., Motta, E.: Relation discovery from the semantic web. In: Bizer, C., Joshi, A. (eds.) Proceedings of the Poster and Demonstration Session at the 7th International Semantic Web Conference (ISWC 2008), Karlsruhe, Germany, October 28. CEUR Workshop Proceedings, vol. 401. CEUR-WS.org (2008)
25. Seo, D., Koo, H.K., Lee, S., Kim, P., Jung, H., Sung, W.-K.: Efficient finding relationship between individuals in a mass ontology database. In: Kim, T.-H., Adeli, H., Ma, J., Fang, W.-C., Kang, B.-H., Park, B., Sandnes, F.E., Lee, K.C. (eds.) UNESST 2011. CCIS, vol. 264, pp. 281–286. Springer, Heidelberg (2011)
26. Sheth, A.P., Ramakrishnan, C.: Relationship web: Blazing semantic trails between web resources. IEEE Internet Computing 11(4), 77–81 (2007)
27. Sieminski, A.: Fast algorithm for assessing semantic similarity of texts. IJIIDS 6(5), 495–512 (2012)
28. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proceedings of the 16th International Conference on World Wide Web, WWW 2007, pp. 697–706. ACM, New York (2007)
29. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature 393(6684), 440–442 (1998)

Publishing Bibliographic Records on the Web of Data: Opportunities for the BnF (French National Library)

Agnés Simon[1], Romain Wenz[1], Vincent Michel[2], and Adrien Di Mascio[2]

[1] Bibliothéque nationale de France, Paris, France
{agnes.simon,romain.wenz}@bnf.fr
[2] Logilab, Paris, France
{adrien.dimascio,vincent.michel}@logilab.fr

Abstract. Linked open data tools have been implemented through `data.bnf.fr`, a project which aims at making the BnF data more useful on the Web. `data.bnf.fr` gathers data automatically from different databases on pages about authors, works and themes. Online since July 2011, it is still under development and has feedbacks from several users, already.

First the article will present the issues linked to our data and stress the importance of useful links and of persistency for archival purposes. We will discuss our solution and methodology, showing their strengths and weaknesses, to create new services for the library. An insight on the ontology and vocabularies will be given, with a "business" view of the interaction between rich RDF ontologies and light HTML embedded data such as `schema.org`. The broader question of Libraries on the Semantic Web will be addressed so as to help specify similar projects.

Keywords: Libraries, Open data, Culture, Project management, Entity linking, Relational database, CubicWeb, Encoded Archival Description, Marc formats, Open Archive Initiative, Text Encoding Initiative.

1 Introduction

The BnF (French national library) sees Semantic Web technologies as an opportunity to weave its data into the Web and to bring structure and reliability to existing information. The BnF is one of the most important heritage institutions in France, with a history going back to the 14th century and millions of documents, including a large variety of hand-written, printed and digital material, through millions of bibliographic records. Linked Open Data tools have been implemented through `data.bnf.fr`, a project which aims at making the BnF data more useful on the Web.

data.bnf.fr publishes data automatically merged from different in-house databases describing authors, works and themes. These concepts are given persistent URIs, as they are the nodal points to our resources and services. We provide **different views of the same information**: HTML and PDF views for humans and raw data in RDF and JSON for machines. This data is freely reusable under an **Open License**. The site, powered by the **open source platform** CubicWeb, queries a **relational database** to generate both HTML and RDF data. Available online since July 2011, this service is under continuous development with several releases per year. After having gathered

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 563–577, 2013.

feedback from the public and users, we are now in a position to report on this use of Semantic Web technologies.

We want to show how we transform a mass of bibliographical data from different databases, to display structured and reliable data in liked data: what were the difficulties and the solutions? What are the impacts in terms of services for libraries and for the wider community of the Web?

2 Context and Goals

2.1 Strength and Weakness of our Data

The BnF (French national Library) took a first step towards the Web with the digital library Gallica (`http://gallica.bnf.fr`)[1], which offers over 2 million documents like books, reviews, images, objects, and scores. Yet when it comes to our data, it sometimes remains hard to find it especially as users have new expectations and habits on the Web. They need to reach digital collections and references to physical documents through simple keyword searches via search engines and "following their nose" from one link to another. As we cannot ask an always broader audience to become familiar with our various catalogues, we have to help them getting oriented in this mass of data. Indeed the BnF holds millions of documents and descriptive data, especially:

- from the "Catalogue général" (`http://catalogue.bnf.fr/`), which is the main catalogue with about 11 millions of bibliographical data including all the French Legal Deposit,
- from the Archives and Manuscripts database (`http://archivesetmanuscrits.bnf.fr/`), with around 150 000 records,
- from the authority files, with more than 2 million authority records on persons, organizations, works or subjects, and structured repositories, such as the list of roles (`http://data.bnf.fr/vocabulary/roles`) that have a value on the Web.

For machines this data is hard to handle: it is hidden in the deep Web, unstructured, and stored in relational databases. The information has originally been produced to manage our collections, before Web standards even existed. Besides, descriptions are maintained in disparate BnF catalogues, reflecting the methods and technologies used for their descriptions. They have been produced in different formats, according to the type of document that is described. For instance, a collection of archives and manuscripts needs a hierarchical structure, to describe documents together, as they were produced and received during the activities of a person. Therefore archives are described in EAD-XML formats, adapted to a hierarchical "tree structure", whereas books and reviews from the main catalogue are described in a MARC format, created in the 1960's for the librarian community and displaying a flat series of records [2][3][4].

Nevertheless libraries have been playing a major role in normalizing data and respecting cataloguing codes, norms and formats. A first step has been taken by adopting XML (TEI [5], EAD-XML [6], Dublin Core [7] for instance) formats to create or exchange data. We also have been using permanent and reliable "ARK" identifiers [8], to identify catalogue records, archival resources, digital objects from Gallica, and authority records, but also for quoting these resources, with a common "resolver". For instance, the digital object `http://gallica.bnf.fr/ark:/12148/`

bpt6k134521m is also accessible with the persistent link http://ark.bnf.fr/ark:/12148/bpt6k134521m. Furthermore these identifiers have been used inside the library to link our bibliographical records from the main catalogue and our archival finding aids in XML to the authority records. Some of these links are already "typed". For instance, the link between a book and its author or contributor is usually specified by a role code, listed and controlled in our repositories. Charles Baudelaire is the translator of this edition of Dix contes by Edgar Poe: the record http://catalogue.bnf.fr/ark:/12148/cb311263053 is linked to the author with role "translator", expressed with the code 0680.

Thus the work on standards and identifiers has made it possible for libraries to become part of the semantic Web (http://www.ifla.org/about-swsig). To do so, library data has to become both linked, and open.

2.2 Business Issues: Libraries in Linked Open Data

This project takes part of the international experimentations of "Linked Open Data" (LOD), that have popped up among national libraries. The Library of Congress, the Deutsche Nationalbibliothek or the British Library display all their bibliographical records in RDF. Libraries across the world show interest on these topics, in a passionate and sometimes controversial way [9], recommending to identify sets of data as "possible candidates for early exposure as Linked Data and foster a discussion about Open Data". It is also an incentive for others to use this material and to give access to culture. That way, the BnF is taking part of the "Open data" movement, to give access to the information to the broader public, by using the most recent technologies. That is why we chose the "Open License" that allows any commercial use under the condition of quoting the source "Bibliothèque nationale de France".

Yet the BnF had specific goals and issues. First we had to deal with different databases, to link metadata of paper documents with its digitized version, or to gather archives with published documents. We had to transform data from non-interoperable databases into structured and exchangeable data compatible with Semantic Web standards. The workflow was to take our data as it is, with its faults and assets, to keep the global data producing process and our existing catalogues. We chose to keep separate the archival base from the bibliographical base, "upstream", and to display data on the Web with common vocabularies, "downstream".

Secondly, data.bnf.fr also builds *pages for humans*, whereas most big libraries display their bibliographic data in triple stores as another kind of bibliographic product for libraries. The quality of the data being irregular as a result of the long history of our catalogue, this data is displayed gradually on the Web.

Finally, as the main purpose of the library is to give access to documents for patrons, the HTML publication had to be coherent with the RDF publication, the data in RDF being just a different view from the same data that is in the HTML page. The URIs displayed in the RDF give actual links to relevant and existing information, which makes the issue on identifiers so important. Besides the RDF and the HTML data model are similar. We chose basic concepts that are relevant for creating a Web page: authors, works, and subjects. It happened to be an opportunity to implement the FRBR (fundamental requirements for bibliographic records) model [10] which is mainly based on

three entities (author, work, and subject) and on the difference between the work, as an intellectual creation, the versions of this work like a translation or an illustrated edition ("expression") and the publication of this creation ("manifestation"). For instance, in data.bnf.fr, you find pages about the work Tales of the grotesque and arabesque (`http://data.bnf.fr/11943795/edgar_allan_poe_histoires_extraordinaires/`) where we gather the different editions of this work. On the page about the author *"Charles Baudelaire"* (`http://data.bnf.fr/11890582/charles_baudelaire/`), we gather links to his works such as `"Le Spleen de Paris"` and to his specific contributions on publications as translator, illustrator, or dedicator..., at the good level of the model. Considering our needs and issues and from a business side, we finally chose to rely on CubicWeb on the following grounds: it is an open source software; it can extract data from different databases, match them and gather them from different databases; it can publish the same information in different views (Web pages for humans, as well as structured data for computers).

2.3 Technical Considerations

Since all the workflow would rely on documenting provenance and merging information, the use of a triple store was not obvious, and did not appear as the only possible option. Alignments had to be made between various sources: several datasets had to be matched and linked. But most of them were already exposed on the "Linked Open Data cloud", with reliable and efficient links. Whenever it was possible, we used existing matching « Hubs » like the Virtual International Authority File (VIAF), now available in Open Data or DBpedia, as go-between to link to other sets, so as to get the best results with the available time and effort. VIAF is an international project to federate the different authority files from many national libraries. We plan to keep using DBpedia in the future to link to other datasets, that are linked to it. Consuming information which is available with an open license is one of the key issues for using existing matchings. We are also creating new matchings, for instance for geographic entities, which imply making an alignment between our geographic subject headings, Geonames, and our "Rameau" subject headings which is used in the library records.

The BnF chose CubicWeb among other software solutions (including triple Stores), because it had good references, a cost-efficient development, and an ability to publish data with Semantic Web standards (RDF, SPARQL, HTML5, CSS3, Responsive Design) and to cope with our data in several formats. It appeared as one of the most advanced open source Python frameworks for data management. In the next section, we will explain this choice by detailing the advantages and drawbacks of this approach, which reflects the common questions that appear when Semantic Web projects enter a production environment and have to become "business as usual".

3 CubicWeb in a Nutshell

CubicWeb is a Semantic Web application framework, licensed under the LGPL. It relies on different widely-used and well established technologies (see Fig.1 for the global architecture of CubicWeb):

– SQL frameworks for the databases (e.g. sqlite, MySql, PostgreSql),
– Python for the core code and the web server,
– Javascript for the client-side logic.

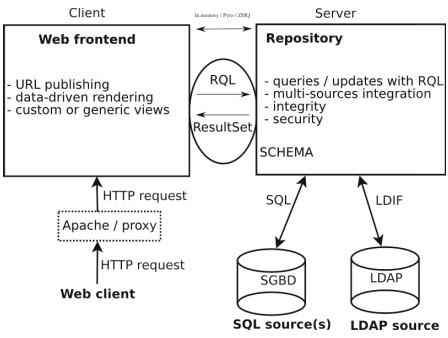


Fig. 1. Overview of CubicWeb architecture

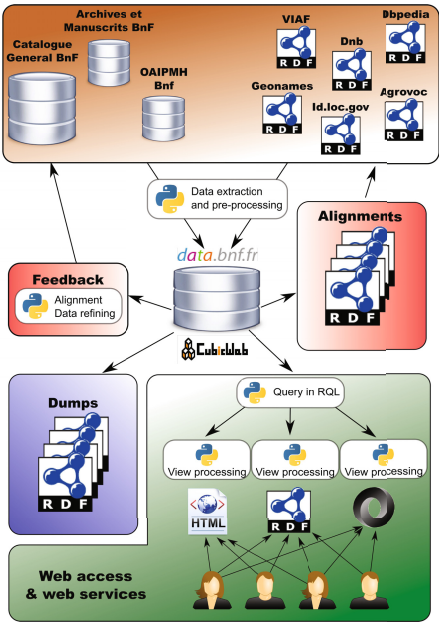


Fig. 2. Global architecture of Data.bnf.fr

3.1 Schema, Views and Entities

CubicWeb applications are structured in three main parts:

– a *schema*, i.e. data model,
– some *views*, i.e information publishing (HTML, PDF, RDF, etc.),
– some *entity classes*, i.e. business logic.

The *schema* defines the data model in terms of attributes/relations/constraints. It is written in Python, and makes the description of the data very simple. Here is an example of the data.bnf.fr schema for an Author (cf. Snippet 1).

With CubicWeb, the result of queries is presented by applying functions named "views". A top-level view can generate a Web page, but also generate a PDF or a JSON file. This is a key distinction when comparing CubicWeb with Web frameworks that are centered on Web pages and not on data. Frameworks centered on Web pages use templates to introduce dynamic content, with a template language that usually becomes cumbersome if one needs more than loops and tests. With CubicWeb, a Web page is a call to the top-level view that calls other views, each of these views call other views, down to the basic text properties of objects. Using templates is possible, but requires

having a piece of template for each function/view. We tried to use templates, but it proved more efficient and readable not to split things into two and to directly emit HTML, XML, text or binary data directly to the output stream. *Views* define the different (fine-grained) ways of displaying data. These views could write (chunk of) HTML pages, but also RDF, CSV, JSON, XML, PDF, etc... (cf. Snippet 2) Each output representation therefore uses the same data. There are no static dumps of data, and every visualization are always up to date. These views may be also used to visualize the same result set in different ways based on the actual context: a result set of works will be displayed differently if we are in the page of their author, or if we are on the page of listing of all the works in the database.

Entity classes define business logic. Some logic could be added to entities, by adding python functions building attributes or relations.

Snippet 1. This defines a *Person*, with a *gender* and a *birthplace*, a link to a *PersonDefinition* called *preferred_form*, and multiple *other_forms*. The *PersonDefinition* is an entity with two attributes *name* and *surname*.

```
class Person(EntityType):
    preferred_form = SubjectRelation('PersonDefinition', cardinality='1?')
    other_forms = SubjectRelation('PersonDefinition', cardinality='1*')
    gender = String(vocabulary=(_('M'), _('F')))
    birthplace = String(maxsize=128)
    # ...

class PersonDefinition(EntityType):
    surname = String(maxsize=256)
    firstname = String(maxsize=128)
    # ...
```

3.2 Relation Query Language

CubicWeb uses a homemade query language, called RQL, which is similar to W3C's query language SPARQL ahd that has been developed since 2001. This language is closely related to the underlying data model, and is used to query the data. It is based on syntax analysis of the query, and can infer information such as entity type from the query. Development of RQL started in 2001. When the normalization process that led to SPARQL started at W3C, Logilab (the company behind CubicWeb) had not enough manpower to participate. Therefore, RQL has been developed in parallel to SPARQL. The two languages share common goals and focus on relationships. Yet there are a few interesting differences between them:

- RQL syntax is often simpler with less punctuation signs and braces,
- RQL has always allowed INSERT, SET and DELETE operations,
- RQL is easily extended with functions, for example, using PostGIS (GIS system based on Postgresql) one can use in RQL the INTERSECTS function defined with the PL/PGSQL procedural language,
- Experience shows that RQL is quickly adopted by power-users, especially with a directive usable in restructured text wiki-like pages `:rql:'<some_query>:<vid>'`

Here are a few RQL queries that are used in data.bnf.fr :

- `Any X WHERE X notice_id 12345678` will return any object in the database that has this notice id, whether it be a work, an author, etc.
- `Any R WHERE X is Rameau, X eid 1234, X broader_concept R` will return all Rameau objects that are broader concepts of the Rameau with internal id 1234,
- `Any X WHERE X is Person, X preferred_form P, P surname ILIKE "A%"` will return all persons whose preferred form's surname starts with 'A' or 'a',

This language is the only way to talk to the database, it supports the four LMD basic operations *Any* (read), *INSERT* (create), *SET* (update), *DELETE* (delete), and also subqueries, orderby, aggregation, functions.

Snippet 2. This view generates the *Other resources* part of an author or work page

```
class OtherRessourcesView(EntityView):
    __regid__ =  'other-ressources'
    __select__ = EntityView.__select__ & is_instance(*DU_ETYPES)

    def cell_call(self, row, col):
        # get the current entity (author or work)
        entity = self.cw_rset.get_entity(row, col)
        if entity.has_other_resources:
            self.w(u'<div class="section" id="other-ressources">')
            self.w(u'<h2>%s</h2>' % self._cw._("other ressources").capitalize())
            # display virtual exhibitions
            self.w(entity.view('virtual-exhibitions'))
            # display BnF's aligned bookmarks
            self.w(entity.view('bnf-bookmarks'))
            # If an author is aligned on dbpedia, display its short abstract
            self.w(entity.view('dbpedia'))
            self.w(u'</div>')
```

3.3 Security

The permission definition is an integral part of the data model definition in a CubicWeb application (cf. Snippet 3). More specifically, the permission model is very simple in data.bnf.fr since nearly everything is readable by anyone. A simple workflow is attached to each kind of entity in the internal data model (e.g. *Person*, *Work*, etc.). Those entities can be temporarily unpublished by an agent using the administration Web interface (which is actually the very same Web application as the official Web site).

With this permissions, the query `Any X WHERE X is Person` will be executed as is if the user is in the *managers* or *users* group but will otherwise be transformed into `Any X WHERE X is Person, X visible TRUE`

This is a real time saver since, as a programmer, you actually don't have to worry about permissions when writing queries, the repository will never return something that the connected user is not allowed to read. The same logic applies for *add*, *update* or *delete* queries.

Snippet 3. Permissions used on main entities: to be able to read an entity, the user issuing the query must either be in one the *managers* or *users* groups. Otherwise, the CubicWeb repository will inject the RQL expression `X visible TRUE` in the original query to make sure that only entities matching this condition will be returned

```
__permissions__ = {
    'read': ('managers', 'users', ERQLExpression('X visible TRUE')),
    'update': ('managers',
               ERQLExpression('U in_group G, G name "users", X in_state S, '
                              'S name IN "temporary-unpublished"')),
    'delete': ('managers',),
    'add': ('managers',)}
```

4 Semantic Point of View

The heart of a CubicWeb application is the data model. Once this data model is defined, the framework is able to generate a database and a Web application instance to add, store, browse and query data fulfilling this data model (see Fig.2 for the global architecture of data.bnf.fr).

4.1 Using Relational Databases

CubicWeb applications have been deployed, used and maintained for 10 years, a time period where quite a few Semantic Web standards were still emerging. For this reason, we decided to stick to well established standards and used SQL relational databases: the knowledge base is huge, every system administration team knows how to deal with them, how to optimize them, how to replicate them, etc. Major Websites can be built upon a triple store, but the SQL relation databases have, in our opinion, a bit more feedbacks from industrial use and make it easier to interact with existing teams who work on relational databases (library catalogues) inside the library. From the library's point of view, it is also an opportunity to keep the producing formats (EAD, MARC...) and workflows, as they are. Furthermore, while RDF is the *de facto* standard in the Semantic Web world for data input/output, Semantic Web applications don't need to rely on a triple-store for internal data management.

In our case, we need to absorb different kind of data, structured or not (Marc XML data, RDF-NT, RDF-XML, CSV files or dumps of relational databases), and therefore using SQL database(s) as a pivot for melting all these data may be interesting. Where triplestores are the natural choice for storing and querying RDF data, we need in our case to serve thousands of daily views, for more than 200.000 web pages, in a rich variety of formats (RDF/JSON/CSV/HTML). This implies a strong structuration and control of the data we put in, and a better integration in a complete Web application. Relational databases and the underlying relational algebra field have been studied for years and have reached both theoretical and practical maturity needed for such applications. Furthermore, with open source SQL backends, we benefit from a huge knowledge base, large communities, and a lot of cookbooks for deployments, optimizations, debugging, etc.

One Model Definition, Several Ontologies Used in Published Data. Using a SQL database is a good way to store the data independently of the different ontologies that may be used to publish them. Indeed, it was easier for us to sketch a data model to store all required information (we knew we had to manipulate authors, books, etc.) but the exact definition of the exposed data model was a delicate issue. For instance, we internally define the notion of `Person` (e.g. *Victor Hugo*), which is later exposed as a *skos:Concept* and as a *foaf:Person* which share common properties but also have specificities. Besides a potential problem of data duplication, enforcing this duality in the data model would complicate the application code and logic since we nearly never have to make this distinction. Furthermore, both ontologies require different granularity of information. *foaf:Person* will need *foaf:name* and *foaf:surname* properties whereas *skos:Concept* will expose a concatenation of those properties in a *dc:title* field.

For those reasons, using a simple and strongly typed data model and storing data efficiently in a SQL database, allows us to program very easily with standard software components and libraries, and to publish data in whichever format is required (several ontologies, several output formats such as JSON, PDF, etc.). Of course, the internal data model changes regularly but CubicWeb provides helpers to do it very smoothly.

Avoiding Duplications. As stated above, another interesting aspect is that we avoid information duplication (which is still important, especially with millions of entities). Indeed, in the previous example, the same SQL records (author name and surname) are used for generating several RDF triples. The same thing is useful for works for example: works have a title that may be represented by a *dc:title* or a *skos:prefLabel*. Using an underlying SQL database avoid data duplication, as the two RDF triples are generated from only one SQL record.

Inner Model Can Be More Stable than Published Ontologies. Keeping the structured information in a SQL database, it is very easy to generate new RDF triples and push them into the graph. Moreover, changes in ontologies are easily handled by regenerating the RDF triples according to the new versions of the ontologies without using more complex tools of ontology evolution.

For example, let's consider the publication of the *BnF ontology* for the authors' roles. An author is related to a document (*Manifestation*) with a given role (*writer*, *scientific editor*, *trompetist*, ...). In the first versions of data.bnf.fr, these roles were published in the RDF using the *id.loc.gov* role referential, whereas in the HTML pages, the BnF's own roles referential was used, with a granularity that better fits its data. In recent versions of the Web site, the BnF referential has been published in RDF and is now used in the generated triples. The current architecture allows us to display the same information with a different granularity between the views.

All Internal Data Doesn't Have to Be Published. Every bit of data in the application is defined according to the internal data model, including *statistics*, *authentication data* from the LDAP directory, *title sort keys*. This information is needed to have a fully functional website but it doesn't make sense to publish them in RDF. Gathering all this

data at the same place is definitely not necessary but it eases development and allows us to build simple query. E.g "Give me the 10 most visited documents, negotiated in RDF, in the last 5 days, sorted by number of visits, then by alphabetical order", and then apply the same `list` view that we can find on standard pages, in two lines of python code:

```
Any X, C ORDERBY C DESC, T LIMIT 10 WHERE H stats_about X, X title_sort_key T,
      H hit_type "rdf", H count C, H period P, P start > TODAY - 5
```

RQL Is Not SPARQL. SPARQL is a great query language that has become the standard in the Semantic Web community. CubicWeb provides a simple SPARQL to RQL translator that transforms a standard CubicWeb application into a SPARQL endpoint. Unfortunately, only a subset of SPARQL is usable and only a subset of the internal data is queryable. This is partly because semantics of both languages differ a bit, but mostly because it requires an automatic mapping of the internal data model (defined in `Yams`, queryable in RQL) to the published data model, which is sometimes just not possible. For simple cases, CubicWeb uses a simple API to define equivalences or transformations between the internal `Yams` datamodel and the published RDF data:

```
# a Person should be translated into a foaf:Person
xy.add_equivalence('Person', 'foaf:Person')
# the surname property is transformed into foaf:familyName
xy.add_equivalence('Person surname', 'foaf:Person foaf:familyName')
# the birthplace is transformed into the placeOfBirth property
xy.add_equivalence('Person birthplace', 'RDAgroup2elements:placeOfBirth')
```

A very simple alternative would be generate all the rdf triples from the internal SQL database, push them in a triplestore and use CubicWeb `hooks` to keep the data up-to-date.

4.2 A General Overview of Involved Datasets

To interlink the data to other datasets, we use the fact that many semantic data are also open, and allow us to avoid to restart all the alignments from zero. The high quality of the alignments and the large number of authors make VIAF a crucial referential to be aligned with. It is also aligned on Wikipedia, and provides bridge between other reference databases. Using the VIAF's dump, we create **15937 exact matches** with the authors in our database. Moreover, this referential database is also a good way to benchmark the alignment tools used or developed in this project (see the *Data Alignment* section below). The matching are based on already existing alignments (e.g. VIAF, Geonames), and were derived using the URIs of the entities. Thus the disambiguation issue was considered as already solved in these dumps.

Thus, appart from the internal databases of the BnF, we use different external open databases and referentials:

`Dbpedia` **5488 exact matches** on Dbpedia, **3947 exact matches** on the french Wikipedia; `DnB` (German National Library) **26088 close matches**; `Geonames` **7038 close matches**, **22951 exact matches**; `Agrovoc` **685 exact matches**; `LCSH` **82937 close matches**; `Sudoc` **3318 exact matches**; `Thesaurus W` **66 close matches**, **979 exact matches**.

By aggregating different RDF dumps (nt, rdf/xml, CSV), and by performing simple string matching, we manage to create more **169290 (close/exact) matches** between the

presented database and more than **8 different referential datasets**. Moreover, these databases allow a rapid and easy increase of the interlinking of our data, as they already present alignments to other database (e.g. Dbpedia is also aligned with `Freebase`, `Project Gutenberg`, `New York Times` ...)

4.3 Data Alignments

Many documents in the original catalog were not aligned on a *FRBR Work*: therefore we had to build such links. For example, the *FRBR Manifestation "Les Misérables, by Victor Hugo"*, should be aligned on the *FRBR Work "Les Misérables"* by the author *"Victor Hugo"*. The first approach is a *Naive alignment*. This alignment strategy is based on basic string matching, with few normalization pre-processings. It basically checks if two strings start similarly, while removing some common stopwords (e.g. *Le*, *Les*, ...):

- *"Misérables, Les, par Victor Hugo"* is aligned to the FRBR Work *"Les Misérables"*.
- *"Les Misérables, édition de 1890"* is aligned to the FRBR Work *" Misérables"*.

However, some cases are far more difficult and are not covered by the previously described business logic, e.g. *"La véritable histoire des Misérables, par Victor Hugo"*. For such cases, we develop a machine-learning based alignment that works on a bag-of-words representation of the *FRBR Manifestations* to be aligned, and on the *FRBR Works* of the author. Basically, we build a one-versus-all classification scheme based on Logistic Regression using the `scikit-learn`, for each of the *FRBR Work* of the author, in order to predict if a new *FRBR Manifestation* may be considered to be a representation of the Work or if it is not close enough compared to the other works. This approach allows to perform the following alignments:

- Multiple references to works, e.g. *"Les Misérables et Notre-Dame de Paris, Victor Hugo"*,
- Deletion/insertion of words, e.g. *"Notre-Dame, 1890"*, Les [1892] Misérables,
- Different words order/mispelling, e.g. *"Notre Dam de Paris"*, *"Paris, Notre-Dame de"*.

Finally, other kinds of data have to be aligned. For example, the Rameau subject *Nice* for the French city in the south of France should be aligned with the heading describing *Nice* and with some external referential such as Geonames. For such large-scale alignment (> 100.000 elements by corpus), we use the `Nazca` (`live demo` python library that provides a high-level API for data alignment, with SPARQL/RQL utilities.

4.4 URIs and URLs

In this project, we face different technological issues. One of them is the requirement of unique ids and stable URIs.

Ids Dequirements and Stable URIs. One of the input database of the project is the *BnF Archives et Manuscripts* database. However, as opposed to the main *BnF Catalogue General database*, we do not have unique ids/URIs to refer to those documents. We decided to use the *ARK* specifications [8] to automatically assign an id to the documents, based on the archives number.

We had to build URIs that are both stable (Semantic Web requirement) and human readable (so that an URI can be clearly related to the concept behind). The main difficulty here relies on the fact that the label used to describe the different concepts may change. Indeed, as a reference authority, the BnF chose a preferred way to label an author. However, this label may change, and using the label in the URI is thus conflictual. To solve this, we build URI of the form *http://data.bnf.fr/ark*

URL Redirection / Content Negotiation. We use the *ark* identifier system to identify each entity in a stable way, and to build the, *resource identifier URI* (following the cool URI conventions [11]) `http://data.bnf.fr/<ark-of-the-entity>`, that redirects to the *document resource URI* `http://data.bnf.fr/<notice-id>/<human-readable-title>`

The `notice-id` part of the URL is an internal, stable and unique identifier used at the BnF to index notices. This identifier is used as a seed to build the final `ark` identifier. The actual content delivered when asked for the *document resource URI* depends on the content negotiation step. Negotiable content types are `RDF` (`nt`, `n3` or `xml`), `PDF` and `HTML`. `Content-Location` header will be set accordingly.

For instance, the following HTTP request:

```
GET http://data.bnf.fr/ark:/12148/cb11928669t
     Accept:text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
     Accept-Charset:ISO-8859-1,utf-8;q=0.7,*;q=0.3
     Accept-Encoding:gzip,deflate,sdch
     Accept-Language:fr-FR,fr;q=0.8,en-US;q=0.6,en;q=0.4
```

will redirect to `http://data.bnf.fr/11928669/voltaire/`:

```
GET http://data.bnf.fr/11928669/voltaire/
     Accept:text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
     Accept-Charset:ISO-8859-1,utf-8;q=0.7,*;q=0.3
     Accept-Encoding:gzip,deflate,sdch
     Accept-Language:fr-FR,fr;q=0.8,en-US;q=0.6,en;q=0.4
```

that will in turn answer:

```
    Content-Encoding:gzip
    Content-Language:fr
    Content-Location:http://data.bnf.fr/11928669/voltaire/fr.html
    Content-Type:text/html;charset=UTF-8
```

5 Discussion and Applications

5.1 Building a Domain-Specific RDF Model with Standard Vocabularies

Libraries have a strong tradition of data exchange and interoperability, with the use of the *Marc* formats since the early seventies. But these library-specific formats are obviously limited to library communities, and hard to use for developers out of the "library world". By moving them to RDF we meant to facilitate new and unexpected uses of our data from different communities. That is why we chose common, simple vocabularies that are widely used and tested on the Web: `skos` [12], to describe concepts; `foaf` [13] for persons and organizations; existing library vocabularies such as `Dublin Core` [7] for bibliographic information. We also used the ontology `bnf-onto` only for classes and properties that were not expressed anywhere else:

http://data.bnf.fr/ontology/bnf-onto/. All these library-specific terms were declared as sub-properties or sub-classes of existing vocabularies, for those who want broader information. For example http://data.bnf.fr/ontology/bnf-onto/ouvrageJeunesse to sort adapted editions of a work for the younger public, and which is a subclass of DCMIterms: text.

Though we tried to have a simple model for end users, we also experiment on the data models that are currently being discussed in the library community. We use the *Functional Requirements for Bibliographic Records* (FRBR model) to express precise and relevant links between our data. This model is specified at http://data.bnf.fr/semanticweb (French) and http://data.bnf.fr/semanticweb-en. We also used and published BnF specific vocabularies (http://data.bnf.fr/vocabulary) that are matched to the Library of Congress: Country codes list, Relator codeslist, Types of subject headings.

5.2 Retrieving Data: Simple or Extensive Re-use

Together with the serializations that we provide, namely RDF/XML, NT, N3, several users asked for a simplified view with the main concepts, the links to the digitized documents, without the whole descriptions of every document. Therefore we offer a JSON view with a simplified model and less resources. There already are developers of small applications, who build timelines for research purposes- or for smartphone applications.

We wanted our data not only to be used on the Web, but also to be visible, in order to reach a new public that do not know about the BnF collections. As *author*, *work* and *subject* pages are open on the Web and can be reached by search engines, we provide HTML embedded data from Schema.org. These elements are used by search engines to identify, disambiguate terms, and, above all, to put forward digital documents. As a consequence, Gallica pages can be easier to find when they are in data.bnf.fr. We also integrated Opengraph Protocol (OG) metadata, so that the pages can be represented in social networks. Adopting these vocabularies answers to a different logic than displaying our data in RDF: as schema.org vocabularies have been created by and for search engines, they are simple and have a high level granularity. The library follows the evolutions of Schema.org for libraries (http://www.w3.org/community/schemabibex/).

Users can download data in RDF-XML, NT or N3, either for each page, through content negotiation or by clicking on the RDF logo, or get a bulk download of all data. As the volume of data is progressively increasing, the site is going through performance difficulties: the RDF pages as well as the RDF dump are hard to generate. So we decided to lighten the RDF of the pages, to attribute actionable URI to manifestations and to split our dump, according to the different uses of our data: lighter dumps for authors, works, subjects, a dump with complete detailed manifestations, and one for external links. To go further, users want to pick and choose the data they need without necessarily downloading everything: for example, taking only the main information about the author (his different names, his dates for instance), and leaving aside the documents to which he contributed. Therefore the next step could be to offer a SPARQL endpoint, to enable a dynamic interrogation and retrieval of our data to the external user, but also

for the BnF operators to have a better knowledge of its own data. This idea of getting to know our collections better through Semantic Web technologies is very important, because it makes the improvements important on the business side, not only in terms of services for the end-users, but also in terms of curation of our collections, in a long-term perspective.

5.3 Enhancing Services in the Long Term

Providing structured data in Open data enables to create new links and new interfaces. We had feedbacks from end-users, mainly directly on the interface, and through statistics on use (search by title of a work for instance). Since all the raw data is available with an Open License, we had comments from developers, or instance on the properties and vocabularies used, or to ask for a SPARQLendpoint. We take these remarks into account as the site is being developed. As the volume of the base is increasing, we may provide new services such as a SPARQL endpoint. We can also have an idea of the kind of content that is being used and how. Some of end-users are re-distributing the dataset and referencing it for others to re-use, starting with `data.gouv.fr`, the official Open data portal of the French State, but also other sites such as CKAN, OKF and Open data directory. Other users are data specialists from the cultural sector, who use a part of the data for specific purposes in their local applications, such as the `Institut français`. This broad range of uses of the "raw data" shows us that library information can be useful for broader communities. Some users now can avoid duplication of data when indexing resources. For example, displaying authority data, such the thesaurus RAMEAU in SKOS, with over 160 000 subjects, was very much expected by users. The project *MACS* (Multilingual access to subjects) [14] is a good use case in that matter. It matches subject authority from the BnF (RAMEAU), the Deutsche Nationalbibliothek, the Library of Congress and the National Library of Switzerland so that users can put the keywords they chose in their own language.

The BnF also tries to enhance new uses of its data. RDF allows the library to create new services, by following the links in RDF graph. For instance, you can provide all the different editions of a work in a digitized version. These links can be used in apparently simple functionalities, such as: finding other editions of a book, digital versions of it, other works by a writer, and so on.

Finally, combining Semantic Web tools with matching techniques, we answer to a great demand inside the library to improve the catalogues at the source without increasing the cataloguers' work. It is of course useful inside the application. But it can also be used in the original library catalogues. Thus, if data.bnf.fr may lead to new services for the external users, it is also a way to improve and correct automatically or semi automatically our own data, in the long term. That is why we try to build routines and mechanisms that can be used inside the original catalogues. Step by step, inferences on our data are integrated to the original sources. A specific interface has been developed on the business side, so that librarians can validate the automatically generated matchings. We can automatically generate *Work* pages inside our authority files or create new links between bibliographic records and the work authority file, following the FRBR principles. The aim is to reduce cataloguing tasks as much as possible, and to create new links, in a way that can be immediately useful for the end-user.

6 Conclusion

Semantic Web tools made our data visible and exchangeable on the Web. As we are manipulating data and not bibliographic records, we could imagine new ways of organising the information, such as pages for a date (e.g. `http://data.bnf.fr/what-happened/date-1515`) or for a role (example: all the authors who have been making coins, such as *Louis XIV*, `http://data.bnf.fr/vocabulary/roles/r370`). The software may also display graphic visualization of the data, such as maps, diagrams or time-lines, and bring new opportunities, about the use of geographical data for instance.

Today (March 2013), `http://data.bnf.fr` displays millions of RDF triples, corresponding to 20 % of the BnF main catalogue. The next step is to increase gradually the volume to include the whole main catalogue, with its 1.6 million authors, in the long term. This will imply performance issues, but also a real opportunity to bring valuable and massive data on the Web. Correlated with matching techniques and data mining, Semantic Web is a condition and an opportunity to create new links and new services in interfaces that have to remain easy to use and quick to understand.

References

1. Martin, F.: Gallica, bibliothèque et plate-forme numériques, 12e Journées des Pôles associés et de la Coopération (2009)
2. Bermès, E.: Des identifiants pérennes pour les ressources numériques. I' expérience de la bnf (2006)
3. Crawford, W.: MARC for library use: Understanding integrated USMARC. Professional librarian series. G.K. Hall (1989)
4. Taylor, A., Miller, D.: Introduction to Cataloging and Classification. Library and Information Science Text Series. Libraries Unltd Incorporated (2006)
5. Stührenberg, M.: The tei and current standards for structuring linguistic data an overview. Journal of the Text Encoding Initiative (3), 1–14
6. Pitti, D.V.: Encoded archival description: An introduction and overview. D-Lib Magazine 5(11) (1999)
7. Dekkers, M., Weibel, S.: Dublin core metadata initiative progress report and workplan for 2002. D-Lib Magazine 8(2) (2002)
8. Kunze, J.A.: Towards electronic persistence using ark identifiers, ark motivation and overview (2003)
9. Baker, T., Bermès, E., Coyle, K., Dunsire, G., Isaac, A., Murray, P., Panzer, M., Schneider, J., Singer, R., Summers, E., Waites, W., Young, J., Zeng, M.: Library linked data incubator group final report: W3C incubator group report. Technical report (October 25, 2011)
10. On the Functional Requirements for Bibliographic Records, IFLA Study Group: Functional requirements for bibliographic records: final report (2009)
11. Sauermann, L., Cyganiak, R., Völkel, M.: Cool uris for the semantic web. Technical Memo TM-07-01, Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (2007)
12. W3C: SKOS Simple Knowledge Organization System Reference (2009)
13. Brickley, D., Miller, L.: FOAF Vocabulary Specification 0.97. Namespace document (2010)
14. Landry, P.: Multilingual subject access: the linking approach of MACS. Cataloging & Classification Quarterly 37(3/4), 177–191 (2004)

Hafslund Sesam – An Archive on Semantics

Lars Marius Garshol and Axel Borge

Bouvet ASA, Oslo, Norway
{larsga,axel.borge}@bouvet.no

Abstract. Sesam is an archive system developed for Hafslund, a Norwegian energy company. It achieves the often-sought but rarely-achieved goal of automatically enriching metadata by using semantic technologies to extract and integrate business data from business applications. The extracted data is also indexed with a search engine together with the archived documents, allowing true enterprise search.

1 Introduction

Every enterprise has a number of different IT systems, each of which maintains an incomplete picture of the enterprise. The full picture is nowhere to be found, because information is not connected across the different systems. Solving this is non-trivial, as traditional systems can only store data which fits their schema, and a single system for the entire enterprise is unrealistic.

We have developed a system called Sesam for Norwegian energy company Hafslund, which collects information from different IT systems and integrates it into a meaningful whole. This allows users to search and browse data across system borders. The system avoids the schema problem by using RDF to store the integrated data.

Sesam is actually Hafslund's internal document archive, but an archive built in an unusual way. Documents are tagged with URIs from the triple store, and these URIs connect the document metadata with enterprise data extracted from backend systems. Having the enterprise data available also allows metadata to be automatically enriched by traversing the data in the triple store.

The system thus improves metadata quality while at the same time reducing the need for manual metadata input by users. In addition, it is used by customer service representatives to find information relevant to callers.

An overview of the system architecture is shown in figure 1 on the facing page.

1.1 User Interface

The user interface to the system is an application built on a search engine, which has indexed both documents and the structured RDF data. The application presents a faceted search interface with entity pages (pages that show all data about one entity), and the ability to navigate from one entity to related entities.

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 578–592, 2013.

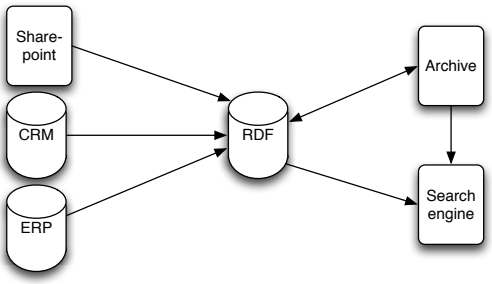


Fig. 1. System architecture

The interface also provides type-ahead functionality to help users understand what they can search for.

In the user interface system users can navigate from a customer in the CRM system to the same customer in the ERP system, and from the ERP customer to connected equipment in the ERP system, and so on.

The user interface is deliberately kept generic, the display logic for RDF data being a direct translation from the structure of the RDF data. Thus new properties and classes can be added to the RDF data and be displayed without modifying the user interface.

1.2 Collecting Information

All source systems are integrated in the same way: a wrapper is added to expose an SDShare server interface. SDShare is a specification for synchronizing RDF data using Atom feeds [SDShare]. Once a source system exposes a set of SDShare feeds the integration is complete, as an SDShare client can then pull the data into the triple store.

The SDShare client is a generic implementation of the SDShare protocol, which periodically polls each data source for new data, and automatically transferring any new data to the triple store, keeping the triple store in sync with sources. At the moment the client polls most sources every 5 minutes, which is more than sufficient for an archive system. Some sources are polled more often, and some as rarely as once an hour.

Data from each source system is kept in a separate graph in the triple store, allowing the source of each statement to be tracked. This also provides a partitioning of the data that is useful for maintenance purposes.

1.3 Archiving

Sesam exposes a web service interface for archiving based on the CMIS standard [CMIS], to allow applications to add support for archiving directly from the application. Thus users can do their archiving from the context of the end-user

application they are working in, without having to turn to a separate archiving tool, and without requiring manual double entry by archivists. This has obvious usability and cost benefits.

Each archiving source gathers as much metadata about the document as it can, and represents it using its own vocabulary. The document is then posted to the CMIS interface, where the CMIS server translates the metadata to the vocabulary used by the archive.

In addition, the metadata is automatically enriched. For example, if the document is tagged with the URI of an electricity meter, the CMIS server will automatically add the URI of the customer currently owning that meter. The metadata translation and enrichment is configured using an RDF vocabulary annotating the CMIS metadata vocabularies. The enrichment code is thus entirely generic, and has no built-in knowledge of the various metadata vocabularies. It also makes the archive clients truly independent of the model used by the archive.

1.4 Ontology

The core of the ontology is at the moment drawn from the ERP system, and contains typical ERP entities like employee, customer, project, and equipment. The ontology is expressed in RDFS, and uses only a few very basic OWL constructs. No reasoning is done using the ontology.

A simplified view of the ontology is shown in figure 2. The full ontology is considerably larger, and changes as new sources and data are added to the system.

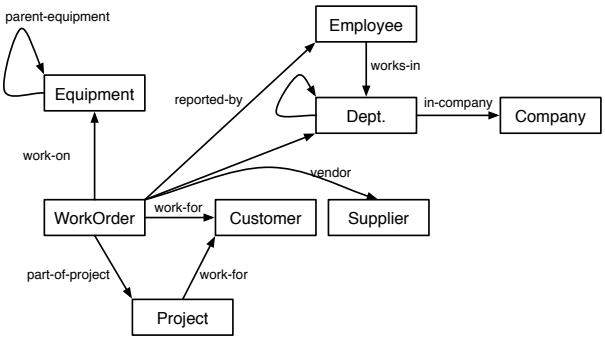


Fig. 2. System ontology

Note that in addition to this core ontology, there are separate ontologies for each source system, subclassed from the core ontology where possible.

2 Principles and Requirements

The architecture of the system has been guided by a few simple principles and requirements described in this section.

2.1 Principles

The interfaces between components should be standards-based, allowing individual components to be changed or replaced without affecting other components. This also enables the use of existing open source or commercial components which support the standards.

The system should be driven by configurations and annotations of the source data, rather than logic defined in code. A corollary is that all mappings should reside in data, not code. Similarly, code should be generic and handle new schema elements correctly without having to be modified. This makes the system much more flexible, and limits the amount of code.

Further, configuration and annotations should be stored in the triple store, rather than in peripheral systems. This makes it easier for developers to make changes without having to be intimately familiar with every component.

Finally, source data should be extracted as-is, and not transformed into a canonical data model for the entire enterprise. Not transforming data dramatically simplifies integrations, and avoids having to "dumb down" the data to the lowest common denominator. Normalization to a common representation can be implemented where necessary as a feedback loop reading source data from the triple store and writing back normalized data.

2.2 Requirements

Archiving, while important and in some cases a legal requirement, is seen by employees essentially as a distraction from their real jobs. It follows that the process must be as simple as possible, and not require users to enter large amounts of metadata.

The system must handle 1000 users, although not necessarily simultaneously.

Initial calculations of data size assumed 1.4 million customers and 1 million electric meters with 30-50 properties each. Including various other data gave a rough estimate on the order of 100 million statements.

The archive must be able to receive up to 2 documents per second over an interval of many hours, in order to handle about 100,000 documents a day during peak periods. The documents would mostly be paper forms recording electric meter readings.

To inherit metadata tags automatically requires running queries to achieve transitive closure. Assuming on average 10 queries for each document, the system must be able to handle 20 queries per second on 100 million statements.

2.3 Technology Choices

To write generic code we must use a schemaless data representation, which must also be standards-based. The only candidates were Topic Maps [ISO13250-2] and RDF. The available Topic Maps implementations would not be able to handle the query throughput at the data sizes required. Testing of the Virtuoso triple store indicated that it could handle the workload just fine. RDF thus appeared to be the only suitable technology.

The canonical approach to RDF data integration is currently query federation of SPARQL queries against a set of heterogeneous data sources, often using R2RML. Given the size of the data set, the generic nature of the transitive closure queries, and the number of data sources to be supported, we considered achieving 20 queries per second with query federation unrealistic.

We therefore had to transfer data from the data sources into the triple store and keep it in sync with changes. Of the open specifications for this SDShare was considered the most suitable.

We chose to use a search engine as the front-end as we considered it better at handling full-text searches of documents, many concurrent user searches, and filtering of search results by access control rules.

2.4 Data Integration

The heart of the data integration is the triple store, in our case Virtuoso. All data in the system, except actual documents and their metadata, is stored in the triple store. In order to reduce the coupling with the triple store product, we only interact with the triple store using SPARQL and SPARQL Update, sent using the SPARQL Protocol. This should theoretically allow us to change triple store without anything more than minor configuration changes in other components.

The data flows between components are implemented using the SDShare protocol.

2.5 The SDShare Protocol

SDShare servers expose data collections, where a collection is a data set defined by the server. It could be an RDF graph internally on the server, but doesn't have to be. The top level of the SDShare interface is the overview feed, which is an Atom feed providing a link to the collection feed for each collection, as shown in figure 3 on the facing page.

The collection feed is the entry point for each collection, and provides two links: one to the snapshot feed for the collection, and one to the fragment feed for the collection. Subscribers to a collection generally record the URL of the collection feed in their configurations.

The snapshot feed contains a list of links to actual snapshots. A snapshot is a representation of the entire collection in some RDF format. Many implementations offer just a single snapshot, which is a service providing a live export of the entire collection to RDF.

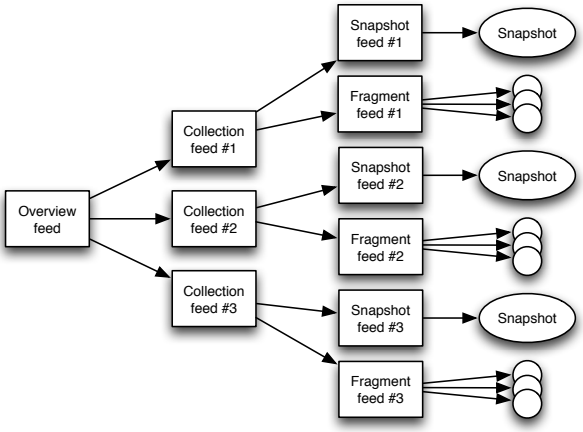


Fig. 3. SDShare server structure

Snapshots serve two purposes: they allow clients to make a local copy before starting to synchronize, and they allow clients to reset their local copy in case there are problems with it.

The fragment feed contains a list of links to fragments. A fragment is a small subset of a collection, typically just all statements where a particular resource is the subject. The fragment feed contains a list of fragments which have changed. By subscribing to it, clients can replicate those changes in their local copies.

The protocol does not inform clients of exactly which triples have changed. Once a resource has changed, the fragment for that resource shows up in the fragment feed, and the client downloads a complete copy of the fragment. The fragment is applied on the client side by simply deleting all statements about the resource in the target graph, and then inserting the new fragment.

The fragment feed often grows very large. In order to avoid having to download the entire feed each time the client polls, a `since` parameter can be added to the request for the fragment feed. The parameter specifies that the client only wishes to see changes after the given time (typically the time of the last change the client has seen).

In addition, servers may page the feed. That is, the fragment feed may be broken into pages, each page providing a `next` link to the next page. Thus clients avoid having to download very large Atom feeds in a single request in cases where there are a large number of changes.

2.6 The SDShare Client

We have developed a generic SDShare client which can be configured with *(Collection feed, SPARQL endpoint)* URI pairs. The client can download a snapshot from the SDShare collection and feed it into the SPARQL endpoint using SPARQL Update. After that, it polls the fragment feed at set intervals for new

fragments. These are also applied to the SPARQL endpoint using SPARQL Update statements.

Adding a new data source thus requires no more than implementing an SDShare server wrapper around the data source, and then adding a new endpoint pair to the configuration. The configuration provides a URI identifying the collection, and this URI is used as the URI of a graph in the triple store. Thus different collections can provide data about the same resources without conflict, as SPARQL Update statements can be used to update the resource in G without modifying it in G'.

The client has support for pluggable backends, and another backend uses a trivially simple HTTP protocol to POST fragments to recipients. This backend is used for recipients which are not triple stores.

2.7 SDShare from the Triple Store

In order to make the RDF data in the triple store available to clients, we expose SDShare feeds from the triple store. This is implemented using an SDShare server framework implemented in Java, which uses SPARQL queries to produce the feeds with `select` queries and the actual snapshots and fragments with `construct` queries. The queries are configurable.

In order to do change data capture we initially added triggers to Virtuoso's RDF data table. These triggers updated a custom table containing the changelog, which was mapped to a virtual graph using Virtuoso's pre-R2RML mapping mechanism, and could thus be queried with SPARQL.

As the system grew, we experienced performance issues with this approach, and so changed the system so that all clients making updates must insert timestamp triples in the triple store when making changes.

2.8 The ERP System

Hafslund uses the IFS ERP system, which is based on an Oracle database. In general, implementing the snapshot part of SDShare on top of Oracle is near-trivial. The difficult part is being able to do change data capture, in order to implement the fragment feed. However, IFS has a history table tracking changes in the database, and the administrator interface can be used to configure which parts of the database have changes tracked. The fragment feed is thus easily implementable through queries against the history table.

This integration has been through a number of iterations, but at the moment it is implemented using the BrightstarDB SDShare server. This is a commercial product which, given a configuration from the relational schema, produces SDShare feeds. It can do change data capture in a number of different ways, including using SQL queries against the change log.

2.9 The CRM Systems

Hafslund at the moment uses two CRM systems: Siebel and Tieto CAB. Here, too, the integration is done at the database level, using the BrightstarDB

SDShare server. The data sets are somewhat larger than from the ERP system, and the databases provide only partial changelogs. This has required using the "last modified" column in cases where data is never deleted. For the remaining cases the BrightstarDB product can compare hashes of database rows against previously stored hashes to see which rows have changed.

The integration with Siebel was completed in just a couple of days. The CAB integration took longer, but in this case the problem was to get the necessary data into the CAB database (as conversion from the system CAB replaces was still in process), and being allowed to access necessary data. That is, the problems were organizational, not technical.

2.10 Sharepoint

The Sharepoint integration reads two SDShare feeds from the triple store, and writes their contents into Sharepoint's taxonomy component (TermStore). The data mapping is a very simple mapping from one RDF property to the TermStore hierarchy, and another to the term labels in TermStore.

The integration keeps copies of the Hafslund organization structure and a hierarchical classification scheme up to date in the TermStore, allowing Sharepoint content to be tagged with these concepts. A separate integration reads the TermStore contents back out as an SDShare feed, so that the internal Sharepoint identifiers for these terms are available in the triple store with sameAs-mappings to the original resources.

2.11 To the Archive

The actual archive system used at Hafslund is Public 360, which takes care of handling basic document metadata, content, versioning, access control, and so on. Public 360 also provides compliance with the Norwegian NOARK archive standard [NOARK5], which is a legal requirement for parts of the Hafslund group.

In order for key metadata required by NOARK to be present in the archive, documents need to be tagged with what Public 360 calls "contacts". These exist in the triple store as employees, customers, and suppliers, and so must be imported into the archive. This was done by configuring a special SDShare feed from the triple store containing only resources of these classes.

For flexibility we wanted to avoid hard-wiring the mappings from the data in the triple store to the Public 360 data model. A mapping vocabulary describing mappings from the RDF data to arbitrary RDF was developed, and is applied by the SPARQL queries used to set up the SDShare feeds. These mappings also filter out data that should not be included.

The Public 360 integration code is thus completely generic, and has no knowledge of the mapping. Instead, RDF statements are mechanically translated into the Public 360 data model, using introspection of the URIs in the RDF to determine which classes and fields in Public 360 to write data to.

2.12 From the Archive

Contacts in the archive are pulled into the triple store, and as their URIs have been stored in the archive, `owl:sameAs` statements to the original resources are included. This allows contact information on archived documents to be translated from the ERP/CRM identifiers for contacts to the archive identifiers for the same contacts.

The SDShare wrapper is implemented against the Public 360 web service API, which provides a log of changes.

3 System Components

3.1 The Search Engine

The search engine used is Recommind. The vendor has added an SDShare connector, allowing Recommind to crawl the SDShare feeds provided by the triple store and the archive to index the entire data set. Which RDF properties to index and display are configured using the Recommind administration GUI.

The user interface application is actually the default search interface of Recommind, heavily customized using JavaScript and CSS. This approach, rather than building a custom application, was chosen in order to save time and cost.

3.2 Archiving

The search engine interface has now been integrated in a number of applications, allowing users to see data from the search engine directly in the application. The integration is done by embedding a generic browser component in the client application.

The integrations also make use of the application context, so that when browsing a particular object in the client application, the web interface displays the entity page for that particular object in Sesam. Thus, when working with a particular customer in a CRM system, the user can switch from the CRM view to the Sesam view to see all relevant information about the customer, including links to duplicates of the customer and information about the same customer in other applications.

At the moment such integrations are provided in IFS, Public 360, Sharepoint, CAB, Siebel, and GeoNIS.

In addition, integrations have been developed that make it possible to send documents directly to the archive from IFS, CAB, and Sharepoint. These work by exploiting functionality for storing documents that's already present in these applications, and picks up the documents for forwarding to the archive. The documents are passed on with their metadata in the source application (including references to related objects in the source application) to the CMIS server.

3.3 The CMIS Server

The CMIS server was implemented using Apache Chemistry OpenCMIS, where we plugged in an implementation of the `createDocument` method. This implementation receives documents, translates metadata to the Public 360 metadata vocabulary, and automatically enriches metadata that's already present.

A mapping vocabulary for CMIS metadata was developed, allowing us to configure things like:

- Mappings from one CMIS property to another.
- Static properties to be inherited from existing values (for example, documents in archive X must have property Y set to Z).
- RDF properties to traverse along to collect additional tags.

In addition, URIs in the metadata identifying resources in the triple store are translated into the URIs for the corresponding resources in the target graph. Incoming metadata may well contain a reference to a customer using its URI from the ERP system, which may need to be translated to the URI of the contact in the archive. This is easily done using SPARQL queries that traverse `owl:sameAs` statements to resources defined in the archive graph (which is the target graph in this context).

In order to inherit metadata by traversing all annotated RDF properties from given tags, repeated SPARQL queries are run to produce transitive closure. Thus, a large number of queries must be run for each archived document.

An example may serve to make this clearer. A subset of the source CMIS metadata might look as shown below. Please note that this is CMIS metadata, represented in the CMIS protocol, and not RDF (CMIS allows URIs as the names of properties).

```
http://.../hummingbird/document-number=3483122
http://.../hummingbird/title=Complaint letter of 2012-07-10
http://.../hummingbird/creation-date=2012-07-13
http://.../hummingbird/references=http://.../ifs/work-order/201013
```

After processing through the CMIS server, it might look as follows:

```
http://.../360/external-id=3483122
http://.../360/archive=3
http://.../360/title=Complaint letter of 2012-07-10
http://.../360/document-date=2012-07-13
http://.../360/tags=http://.../ifs/work-order/201013
http://.../360/tags=http://.../360/project/4882
http://.../360/tags=http://.../360/contact/35823
```

Here we have translated the metadata to the fields used by Public 360, added a static value, and traversed outwards from work order 201013 to find related objects. These related objects have URIs in the IFS graph, but have been translated to the corresponding URIs in the 360 graph, via `owl:sameAs` statements.

3.4 Access Control

There are strict access control rules on many of the documents in the archive, as some contain personal information about individuals and others contain confidential commercial information.

Each individual application has its own acccess control implementation, including the archive system. Access control information is extracted from each system together with the other enterprise data, so that the triple store contains the access control group memberships and settings.

When a user logs in to the search engine their access group memberships are loaded from the triple store. Once the groups have been loaded, Recommind automatically performs real-time filtering of search results based on the user's group memberships.

3.5 Deduplication

Data quality analysis of the ERP system quickly showed that it contains many records representing the same real-world entities (duplicates). This is caused by a number of factors, one being the design of the relational schema, which has one table each for Hafslund group companies, employees, customers, and suppliers. Unfortunately, these four categories overlap considerably, which forces data duplication.

There is also much duplication internally within the customer and supplier tables. This seems to be partly caused by limitations on how payment information is attached to these entities, and partly by careless data entry by users.

The consequences for information retrieval are serious, however. Imagine wanting to find a document about customer when the customer is registered 10 times. To find the document the user is forced to repeat the search for each customer copy. It's clear that this is going to be a problem in practice.

To solve this problem we turned to record linkage techniques [Winkler06]. A quick review of existing software found many tools, but none that seemed to meet our requirements for such a tool, which would have to support:

- Receiving data via SDShare.
- Storing the links found in a database.
- Continuously receiving new data and updating the link database.

In the end we implemented our own record linkage engine, known as Duke [Duke], which solved the problem. Both precision and recall of the deduplication done by this engine appears to be satisfactory for user purposes.

Duke maintains a single table of links in an Oracle database, with time stamps in the table, allowing us to easily expose the links in an SDShare feed. Links are expressed as `owl:sameAs` and `haf:possiblySameAs` statements. The SDShare client thus pulls the discovered links back into the triple store, where they are stored in a separate Duke graph, and displayed by the search engine application.

4 Evaluation

The project has been through a pilot phase, and the implementation phase started in late 2010. The system went into production in the autumn of 2011.

4.1 Performance and Scalability

Triple Store. To give an impression of the scale of the system, table 1 contains an overview of the size of the main graphs in the development environment. Ontology and mapping graphs as well as some graphs with reference data are omitted. The total number of statements in the system is around 630 million, and growing daily. (Hummingbird is the old archive, now replaced by Sesam.)

Table 1. Graph sizes

Graph	Statements
IFS data	5,417,260
Public 360 data	3,725,963
GeoNIS data	44,242
Tieto CAB data	138,521,810
Hummingbird data 1	32,619,140
Hummingbird data 2	165,671,179
Hummingbird data 3	192,930,188
Hummingbird data 4	48,623,178
Address data	2,415,315
Siebel data	36,117,786
Duke links	4,858

Virtuoso has held up to these data sizes very well, running in a 2-node cluster in order to provide failover. It's possible to write queries that run slowly, obviously, but generally performance is good. Virtuoso used to freeze for a few minutes when doing checkpoints, but a configuration change fixed this. As end-users only interact with the search engine the consequences were in any case limited.

Search Engine. Initially, the Recommind search engine was too slow. Searches generally took on the order of 5-10 seconds. The cause was that each RDF property in the triple store was a separate facet, and Recommind scaled poorly with the number of facets. By collapsing these properties into a smaller number of semantically equivalent facets, search times were reduced to less than a second.

However, Recommind cannot index and search at the same time, so searches used to hang for 30 seconds after each indexing. This is a serious problem when indexing runs once every five minutes. Tuning has reduced this issue.

SDShare Synchronization. Synchronization via SDShare has performed very well. Generally, importing a snapshot is faster than transferring the same amount of data via fragments. The performance also varies with the source and sink involved. For our purposes, performance has been adequate. The average time to process a fragment varies from 50 to 450 milliseconds, depending on the source/sink combination.

The SDShare client has been optimized somewhat from the original, naive implementation, to a multi-threaded design where different transfer jobs can run in parallel. In addition, the frontends and backends now use persistent HTTP connections, in order to avoid having to open and close three TCP connections per fragment, as was previously the case.

4.2 Architectural Properties

The system has a number of relatively unusual architectural properties, which in our opinion has contributed greatly to the success of the project:

- Generally, the data integrations are nearly stateless, since the integrations only expose Atom feeds. This greatly simplifies the integrations, and also means they can be deployed on any number of nodes. The SDShare client has a minimal amount of state per integration: the timestamp of the last change.
- The application of SDShare fragments is idempotent, so fragments can be processed more than once with no adverse effects.
- If necessary, we can delete the entire contents of the triple store, and reload everything from the source.
- Uniformity. All data integrations follow the exact same approach.
- Simplicity. Most components in the system (except the CMIS server) are simple, and easy to understand.

4.3 Architectural Flexibility

The architectural flexibility of the system has been proved several times over, in our view.

Changing Components. Perhaps the best example is the extraction of data from the ERP system. Originally, this was implemented using the Ontopia Topic Maps engine, which used a DB2TM component to map the relational data to Topic Maps, and then used the built-in SDShare server to expose SDShare feeds. This worked fairly well, but was a bit slow, and required a big, separate component to be set up and maintained.

Eventually, Ontopia was replaced in favour of Virtuoso's built-in virtual graph mechanism, using the pre-R2RML functionality. The Oracle tables of the ERP system were linked in and mapped, and then queried with SPARQL to produce SDShare feeds using the existing SPARQL-to-SDShare server. This had excellent

performance and worked very well, until we needed to UTF-8 encode URIs to handle primary keys in the ERP system containing non-ASCII characters. After that change performance degraded, and we were not able to fix it.

Finally, we switched to the BrightstarDB SDShare server, which is the component currently used.

In each of these cases, the change had no effect on any other component, except that the SDShare endpoint URI in the SDShare client changed.

Handling Duplicates. When the problem with duplicate resources was discovered we quickly came up with the solution of one SDShare feed into the deduplicator, and another SDShare feed going back. To create the first SDShare feed required no more than a few SPARQL queries in the configuration. The second was likewise trivial to set up.

The entire problem was solved simply by adding a new component, and wrapping it with already existing components. It's difficult to imagine any comparably simple solution with traditional technologies.

4.4 Ease of Development

When the project started, only a few of the developers were familiar with RDF, SPARQL, and SDShare. Generally, this has not been a problem, but for some developers writing generic code that does not have hard-wired data binding has been a bit of a challenge.

4.5 Stability

The stability of the tools throughout has generally been excellent, with the exception of the Public 360 archive system.

Synchronization via SDShare has worked well and been mostly stable. If an SDShare sync process stops for some reason, the only real consequence is that data does not get updated. Once the problem is resolved by admins the data flow starts again, catching up with changes that had not been applied.

It is worth contrasting this with the query federation approach, where the failure of either a data source or the mapping to it risks making the entire system fail or causes part of the data to disappear until the problem is resolved.

4.6 Usability

An interview with a project participant representing the users indicated that users were satisfied with the ability to find content, describing it as "good". The benefits from being able to navigate across system boundaries were then not yet realized, as only the ERP integration was in production at the time.

The users complained about "instability", meaning the indexing problem described in 4.1), and the user interface having some issues with handling of context based on cookies.

The user interface also lacks some functionality users want, such as the ability to attach documents to emails directly from search results.

4.7 Other Aspects

The project won the "Archive of the Year 2012" prize from the Norwegian Archive Council. The rationale was "innovative and strategic use of technology" to "improve data gathering and simplify the use of metadata".

The customer has stated that while the project was expensive, the project has paid for itself through cost savings at the document center [Pretorius2012].

5 Conclusion

Overall, we not only consider this project a success, but have reused the general architecture in other projects with excellent results. Three projects for other customers have already used the same technology, and we expect many more to follow.

Our experience is that using RDF greatly simplifies information integration compared to traditional technologies. We also consider SDShare a key enabler, as it greatly contributes to the simplicity and flexibility of the architecture.

References

[CMIS] Content Management Interoperability Services (CMIS) Version 1.0; OASIS Standard (May 01, 2010), http://docs.oasis-open.org/cmis/CMIS/v1.0/os/cmis-spec-v1.0.pdf

[Duke] Duke; open source software, http://code.google.com/p/duke/

[Winkler06] William, W.E.: Overview of Record Linkage and Current Research Directions. Research report series (Statistics #2006-2) (February 08, 2006); U.S. Census Bureau, http://www.census.gov/srd/papers/pdf/rrs2006-02.pdf

[ISO13250-2] ISO 13250-3: Topic Maps – Data Model; International Organization for Standardization; Geneva, http://www.isotopicmaps.org/sam/sam-model/

[NOARK5] Noark 5 – Standard for elektronisk arkiv; Arkivverket; versjon 3.0 (March 01, 2011), http://www.arkivverket.no/arkivverket/Offentlig-forvaltning/Noark/Noark-5/

[Pretorius2012] Pretorius, J.A.: Enkel arkivering, sikker gjenfinning, og deling av virksomhetskritisk informasjon i Hafslund; oral presentation (November 21, 2012), Video: https://new.livestream.com/accounts/233730/events/1689454

[SDShare] Moore, G., Garshol, L.M.: SDShare - A Protocol for the Syndication of Resource Descriptions; version 1.0 draft (July 10, 2012), http://www.sdshare.org/spec/sdshare-current.html

Connecting the Smithsonian American Art Museum to the Linked Data Cloud

Pedro Szekely[1], Craig A. Knoblock[1], Fengyu Yang[2], Xuming Zhu[1], Eleanor E. Fink[1], Rachel Allen[3], and Georgina Goodlander[3]

[1] University of Southern California, Los Angeles, California, USA
{pszekely,knoblock}@isi.edu, xumingzh@usc.edu, efink@ifc.org
[2] Nanchang Hangkong University, Nanchang, China
frueyang@gmail.com
[3] Smithsonian American Art Museum, Washington, DC, USA
{rallen,goodlanderg}@si.edu

Abstract. Museums around the world have built databases with metadata about millions of objects, their history, the people who created them, and the entities they represent. This data is stored in proprietary databases and is not readily available for use. Recently, museums embraced the Semantic Web as a means to make this data available to the world, but the experience so far shows that publishing museum data to the linked data cloud is difficult: the databases are large and complex, the information is richly structured and varies from museum to museum, and it is difficult to link the data to other datasets. This paper describes the process and lessons learned in publishing the data from the Smithsonian American Art Museum (SAAM). We highlight complexities of the database-to-RDF mapping process, discuss our experience linking the SAAM dataset to hub datasets such as DBpedia and the Getty Vocabularies, and present our experience in allowing SAAM personnel to review the information to verify that it meets the high standards of the Smithsonian. Using our tools, we helped SAAM publish high-quality linked data of their complete holdings (41,000 objects and 8,000 artists).

1 Introduction

Recently, there have been a number of efforts to publish metadata about the objects in museums as Linked Open Data (LOD). Some notable efforts include the Euopeana project [7], which published data on 1,500 of Europe's museums, libraries, and archives, the Amsterdam Museum[3], which published data on 73,000 objects, and the LODAC Museum [11], which published data from 114 museums in Japan. Despite the many recent efforts, there are still significant challenges in publishing data about artwork to the linked data cloud. Mapping the data of a museum to linked data involves three steps:

1. **Map the Data to RDF.** The first step is to map the metadata about works of art into RDF. This involves selecting or writing a domain ontology with standard terminology for works of art and converting the data to RDF

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 593–607, 2013.

according to this ontology. De Boer et al. [3] note that the process is complicated because many museums have rich, hierarchical or graph-structured data. The data often includes attributes that are unique to a particular museum, and the data is often inconsistent and noisy because it has been maintained over a long period of time by many individuals. In past work, the mapping is typically defined using manually written rules or programs.

2. **Link to External Sources.** Once the data is in RDF, the next step is to find the links from the metadata to other repositories, such as DBpedia or GeoNames. In previous work, this has been done by defining a set of rules for performing the mapping. Because the problem is difficult, the number of links in past work is actually quite small as a percentage of the total set of objects that have been published.
3. **Curate the Linked Data.** The third step is to curate the data to ensure that both the published information and its links to other sources within the LOD are accurate. Because curation is so labor intensive, this step has been largely ignored in previous work and as a result links are often inaccurate.

Our goal is to develop technology to allow museums to map their own data to LOD. The contributions of this paper are an end-to-end approach that maps museum source data into high quality linked data and the corresponding lessons learned in performing this mapping. In particular, we describe the process and the lessons learned in mapping the metadata that describes the 41,000 objects of the Smithsonian American Art Museum (SAAM). This work builds on our previous work on a system called Karma for mapping structured sources to RDF. However, in the real-world data provided by the Smithsonian, we discovered that there were complex structures that required new capabilities in Karma. In terms of linking, we found that mapping the entities, such as artist names, to DBpedia could not be easily or accurately performed using existing tools, so we developed a specialized mapping approach to achieve high accuracy matches. Finally, to ensure that the Smithsonian publishes high quality linked data, we developed a curation tool that allows the museum staff to easily review and correct any errors in the automatically generated links to other sources.

In the remainder of this paper, we describe our approach and present the lessons learned in mapping (Section 2), linking (Section 3), and curating (Section 4) the SAAM data. For each of these topics, we describe our approach, present lessons learned, and evaluate the effectiveness of our approach. We then compare our work to previous work (Section 5) and conclude with a discussion of the contributions and future work (Section 6).

2 Mapping the Data to RDF

2.1 The SAAM Database

SAAM stores collection metadata in a relational database managed by TMS[1], a comprehensive data management system for museums. The SAAM deployment of TMS consists of over 100 tables, containing significant amounts of data

[1] `http://gallerysystems.com/tms`

that needs to remain private (e.g., financial information). In order to avoid issues about private data, we only use the tables that the museum uses to populate their Web site. All the information in these eight tables already appears on the museum Web site, so the museum is amenable to publishing it as linked data. The structure and format of these data are tailored to the needs of the Web site and some fields need to be decoded to produce appropriate RDF. For example, descriptive terms are encoded in text such as "Authorities\Attributes\Objects\Folk Art". The database includes data about 41,267 objects and the 8,261 artists who created them.

For objects, the database contains the regular tombstone information including classification (e.g., sculpture, miniature), their role (e.g., artist, printer), available images, terms (e.g., Portrait Female – Turner, Tina). For artists, the database contains names, including multiple variants (e.g., married name, birth or maiden name), title and suffixes, biographical information and geographical information including city, county, state and country of relevant places (e.g., birth and death place, last known residence) and citation information.

Lesson 1: Satisfy the Legal Department First. Much of the data in museums is proprietary and getting approval from the legal department can be challenging. We use the data that drives the Web site; it is not the raw data, but adequate and circumvents issues that could have stopped or delayed the project.

2.2 Europeana Data Model (EDM)

The Europeana Data Model (EDM[2]) is the metamodel used in the Europeana project[3] to represent data from Europe's cultural heritage institutions. EDM is a comprehensive OWL ontology that reuses terminology from several widely-used ontologies: *SKOS* [4] for the classification of artworks, artist and place names; *Dublin Core* [5] for the tombstone data; *FOAF* [6] and *RDA Group 2 Elements* [7] to represent biographical information; *ORE* [8] from the Open Archives Initiative, used by EDM to aggregate data about objects.

The SAAM ontology[9] (Figure 1) extends EDM with subclasses and subproperties to represent attributes unique to SAAM (e.g., identifiers of objects) and incorporates classes and properties from *schema.org* [10] to represent geographical data (city, state, country). We chose to extend EDM because this maximizes compatibility with a large number of existing museum LOD datasets.

[2] `http://www.europeana.eu/schemas/edm/`
[3] `http://europeana.eu`
[4] `http://www.w3.org/2004/02/skos/`
[5] `http://purl.org/dc/elements/1.1/` and `http://purl.org/dc/terms/`
[6] `http://xmlns.com/foaf/0.1/`
[7] `http://rdvocab.info/ElementsGr2`
[8] `http://www.openarchives.org/ore/terms/`
[9] `http://americanart.si/linkeddata/schema/`
[10] `http://schema.org/`

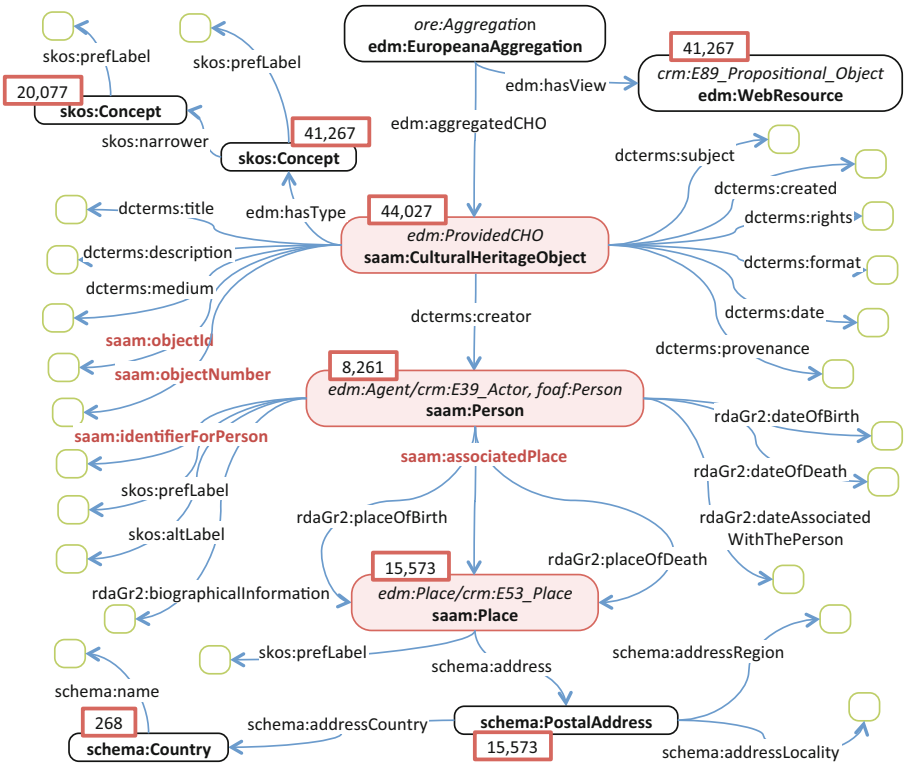


Fig. 1. The SAAM ontology. Named ovals represent classes, un-named green ovals represent literals, arcs represent properties, boxes contain the number of instances generated in the SAAM dataset, italicized text shows superclasses, all properties in the saam namespace are subproperties of properties in standard vocabularies.

One of the most challenging tasks in the project was selecting and extending the ontologies. We considered EDM and CIDOC CRM[11]; both are large and complex ontologies, but neither fully covers the data that we need to publish. We needed vocabularies to represent biographical and geographical information, and there are many to choose from. Following the lead of the Amsterdam Museum [3], we used RDA Group 2 Elements for the biographical information. We didn't find guidance for representing the geographical information in the cultural heritage community so we selected schema.org as it is a widely used vocabulary. Our extensions (shown in boldface/red in Figure 1) are subclasses or subproperties of entities in the ontologies we reuse.

Lesson 2: A Library of Ontologies for Cultural Heritage Is Desperately Needed. While EDM represents an excellent starting point for modeling cultural heritage data, the community can benefit from guidance on vocabularies to represent data not covered by EDM and an integrated library with the recommended ontologies.

[11] `http://www.cidoc-crm.org`

2.3 Using KARMA to Map the SAAM Data to RDF

Prior Work. In previous work [9], we developed KARMA, a tool to map structured data to RDF according to an ontology of the user's choice. The goal is to enable data-savvy users (e.g., spreadsheet users) to do the mapping, shielding them from the complexities of the underlying technologies (SQL, SPARQL, graph patterns, XSLT, XPath, etc). KARMA addresses this goal by automating significant parts of the process, by providing a visual interface (Figures 2 to 4) where users see the KARMA-proposed mappings and can adjust them if necessary, and by enabling users to work with example data rather than just schemas and ontologies. The KARMA approach to map data to ontologies involves two interleaved steps: one, assignment of *semantic types* to data columns and two, specification of the relationships between the semantic types.

A semantic type can be either an OWL class or the range of a data property (which we represent by the pair consisting of a data property and its domain). KARMA uses a conditional random field (CRF) [10] model to learn the assignment of semantic types to columns of data from user-provided assignments [5]. KARMA uses the CRF model to automatically suggest semantic types for unassigned data columns (Figure 2). When the desired semantic type is not among the suggested types, users can browse the ontology to find the appropriate type. KARMA automatically re-trains the CRF model after these manual assignments.

The relationships between semantic types are specified using paths of object properties. Given the ontologies and the assigned semantic types, KARMA creates a graph that defines the space of all possible mappings between the data source and the ontologies [9]. The nodes in this graph represent classes in the ontology, and the edges represent properties. KARMA then computes the minimal tree that connects all the semantic types, as this tree corresponds to the most concise model that relates all the columns in a data source, and it is a good starting point for refining the model (Figure 3). Sometimes, multiple minimal trees exist, or the correct interpretation of the data is defined by a non-minimal tree. For these cases, KARMA provides an easy-to-use GUI to let users select a desired relationship (an edge in the graph). KARMA then computes a new minimal tree that incorporates the user-specified relationships.

Challenge 1: Data Preparation. We encountered multiple situations where we had to filter and transform data prior to modeling it and converting it to RDF. The following are the the types of data preparation tasks we encountered: *Filtering tables:* for example, the SAAM tables represent constituents, which includes both people and organizations. The ontologies for people and organizations are different so we defined database views to filter the tables accordingly. *Data extraction:* for example, the keywords associated with the art objects need to be extracted from text such as "Authorities\Attributes\Objects\Subject Specific\Animal\bird\owl". *Concatenating and formatting columns:* the SAAM tables represent people names, dates and places in a structured form (e.g., year, month and date in separate columns). We needed to concatenate these fields to construct values for single properties (e.g., dateOfBirth), taking care to insert separators and leading zeroes to format them appropriately.

We addressed these data preparation tasks before modeling the data in KARMA by defining views and stored procedures in the database. We then loaded the new tables and views in KARMA to model them. While data preparation is routine in database applications and powerful tools are available to support them, RDF mapping tools (including KARMA) lack the needed expressivity. Tools like ClioPatria [3] allow users to define expressions in a full programming language (Prolog in the case of ClioPatria) and invoking them within their mapping rules. Our approach is to enable users to use whatever tools they are familiar with in a prior data preparation step.

Lesson 3: The Data Preparation/Data Mapping Split Is Effective. The range of data preparation tasks is open-ended and ad hoc. It is wise to acknowledge this and to design a data mapping architecture that is compatible with traditional data preparation tools. This allows the data mapping language to remain relatively simple. KARMA integrates with a data preparation step by providing the ability to specify many aspects of the mapping in the data tables themselves (discussed below). We did the data preparation primarily using SQL views, other users of KARMA have used Google Refine[12].

Challenge 2: Mapping Columns to Classes. Mapping columns to the ontology is challenging because in the complete SAAM ontology there are 407 classes and 105 data properties to choose from. KARMA addresses this problem by learning the assignment of semantic types to columns. Figure 2 shows how users define the semantic types for the constituentid (people or organizations) and place columns in one of the SAAM tables. The figure shows a situation where KARMA had learned many semantic types. The left part shows the suggestions for constituentid. The SAAM database uses sequential numbers to identify both constituents and objects. This makes them indistinguishable, so KARMA offers both as suggestions, and does not offer other irrelevant and incorrect suggestions. The second example illustrates the suggestions for the place column and shows how users can edit the suggestions when they are incorrect.

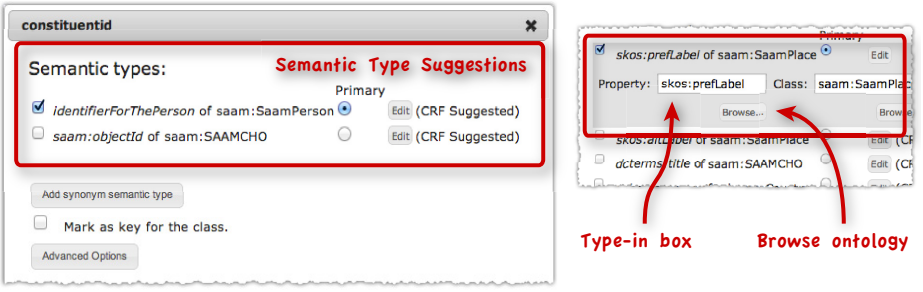


Fig. 2. Semantic types map data columns to classes and properties in an ontology. Left: KARMA suggestions to model the constituentid column in a SAAM table (the first choice is correct). Right: user interface for editing incorrect suggestions.

[12] http://code.google.com/p/google-refine/

Challenge 3: Connecting the Classes. This is also challenging because there are 229 object properties in the ontology to choose from. Figure 3 illustrates how KARMA automatically connects the semantic types for columns as users define them. In the first screen the user assigns a semantic type for consitutentid. In the second screen, the user assigns a semantic type for place, and KARMA automatically adds to the model the associatedPlace object property to connect the newly added SaamPlace to the pre-existing SaamPerson. Similarly, when the user specifies the semantic type for column city, KARMA automatically adds the address object property.

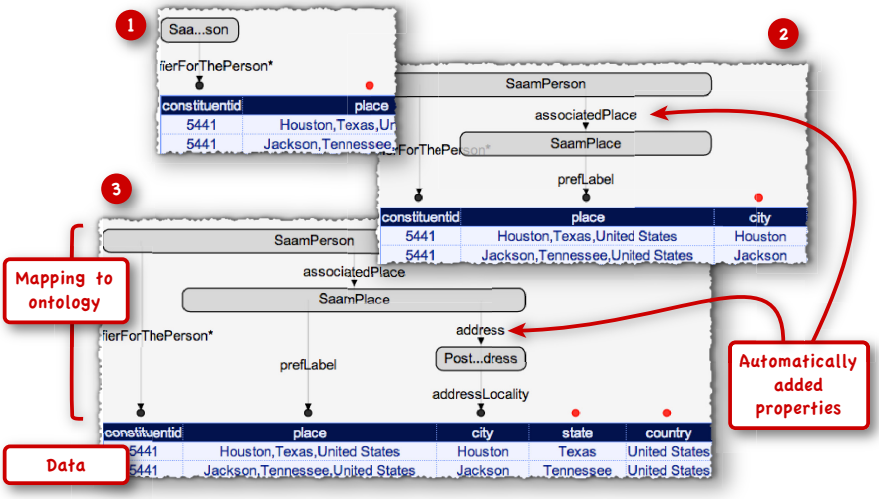


Fig. 3. Each time the user adds new semantic types to the model, KARMA connects them to the classes already in the model

Each time users model the semantic type of a column, KARMA connects it to the rest of the model. In the examples, the connections use a single property, but KARMA searches the whole graph and finds longer paths when appropriate. In addition, weights in the graph [9] bias the algorithm to prefer specific properties rather than general properties inherited from superclasses. Sometimes, multiple lowest-cost models exist, or the appropriate model is not the lowest-cost model. Users can easily adjust the proposed model by clicking on an incorrect property and selecting the appropriate one from a menu of all compatible properties.

Lesson 4: Property Domain and Range Definitions Are Important. KARMA leverages domains and ranges to automate modeling the relationships between the columns in the tables, often selecting the correct property. When KARMA proposed non-sensical, complicated paths to connect classes (e.g., subclass path via Thing), it was often because properties lacked domain or range information or because the classes we defined had not been defined as subclasses of the appropriate classes. This feedback helped us to integrate the SAAM-specific ontology with the large complex ontologies we are reusing.

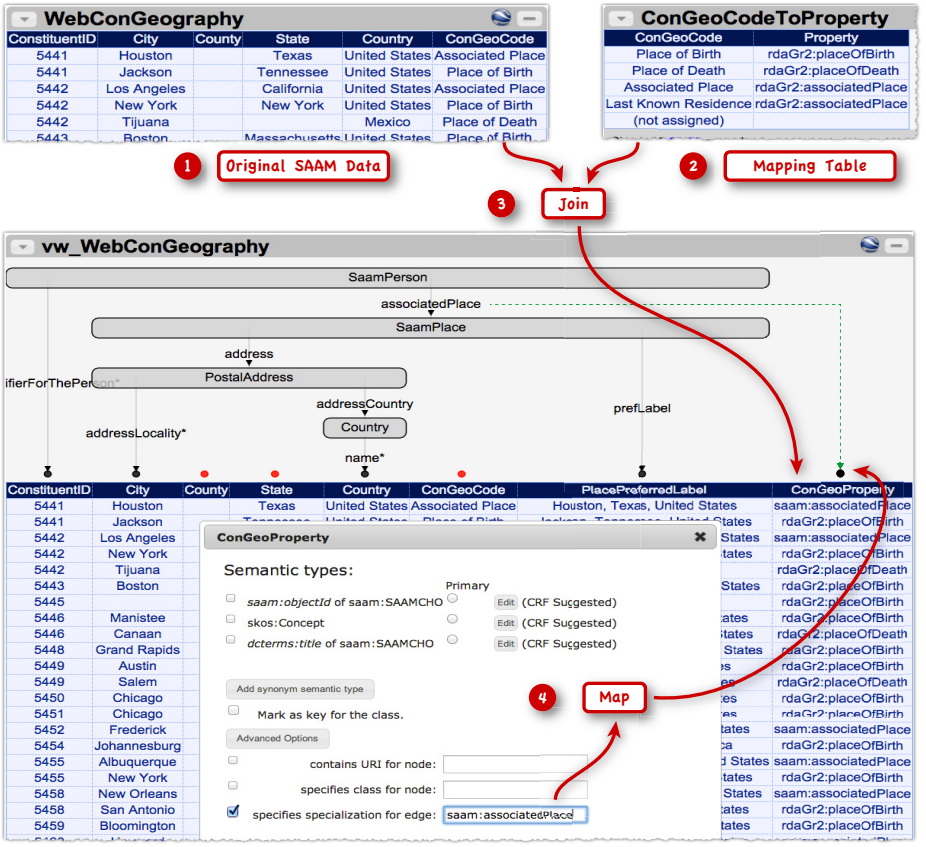


Fig. 4. Mapping columns where different rows must be mapped using different properties: 1) the data table; 2) a table to translate ConGeoCode to ontology properties; 3) join to add a column with the property URIs; 4) map values in the ConGeoProperty column to the associatedPlace property.

Challenge 4: Mapping Depends on Field Values. Figure 4 illustrates a situation where the mapping from a table to the *desired* ontology cannot be specified at the schema level. The WebConGeography table contains information associated with people. Each row represents a place association: the first column (ConstituentID) represents the person identifier, the middle columns represent the place and the last column (ConGeoCode) represents the meaning of the place. The SAAM ontology defines a generic property associatedPlace to represent the relationship between a person and a place. This general property is appropriate for the first and third rows, but not for the others (e.g., the second row should use the more specific property rdaGr2:placeOfBirth).

To model these situations, users add a column that contains the required data. In the particular case illustrated in Figure 4, the user can define a table that maps the ConGeoCodes to the appropriate properties (step 2) and then do a join to add

the new column (step 3). Finally, when defining the semantic type for the new column (step 4), users can specify that the values in the column specialize the associatedPlace property. Analogous situations arise in the SAAM tables that represent data about dates associated with people and variant people names. This type of mapping can be defined in tools such as D2RQ[13], but requires an expert user to define multiple, complex conditional mapping rules. The ability to easily define these data-specific mappings is new since our prior work in KARMA.

Lesson 5: Supporting Row-Level Metadata Solves Many Complex Mapping Problems. The same mechanism we used to model row-specific properties can be used to model row-specific classes, URIs and language tags. It enables users to invoke arbitrary computation using their favorite tools to define data-dependent aspects of the mapping that cannot be cleanly represented in declarative representations. Other approaches such as D2RQ offer a limited set of built-in functions (e.g., concatenation, regular expression) that can be extended by writing Java classes. Our approach enables users to use whatever tools they are comfortable using.

Evaluation. We evaluated the effectiveness of KARMA by mapping 8 tables (29 columns) to the SAAM ontology (Table 1). We performed the mapping twice: in *Run 1*, we started with no learned semantic types, and in *Run 2* we ran KARMA using the semantic types learned in the first run. The author of the paper that designed the ontology performed the evaluation. Even though he knows which properties and classes to use, when KARMA didn't suggest them he used the browse capability to find them in the ontology instead of typing them in. It took him 18 minutes to map all the tables to RDF, even in the first run, when KARMA's semantic type suggestions contained the correct semantic type 24% of the time. The second run shows that the time goes down sharply when users don't need to browse the ontology to find the appropriate properties and classes. The evaluation also shows that KARMA's algorithm for assigning relationships among classes is very effective (85% and 91% correct in *Run 1* and *Run 2*).

Lesson 6: Ontology Design Is the Hard Part. Even though it takes about 8 to 18 minutes to map all the tables using KARMA, it took about 2 weeks after the initial design of the ontology to map all the tables. We spent the time designing and redesigning the ontology. During that period, we mapped the tables many times to slightly different ontologies. So, in Table 1 *Run 2* is typical as we spent significant type rerunning KARMA after slight changes to the ontology.

Table 1. Effectiveness of KARMA's automation capabilities

	# of times KARMA's top 4 suggestions contain the correct semantic type	*# of times KARMA correctly assigns relationships among classes*	*Time (minutes)*
Run 1	7 out of 29 (24%)	30 out of 35 (85%)	18
Run 2	27 out of 29 (93%)	32 out of 35 (91%)	8

[13] http://d2rq.org

3 Linking to External Resources

The RDF data will benefit the Smithsonian museum and the community if it is linked to useful datasets. We focused on linking SAAM artists to DBpedia[14] as it provides a gateway to other linked data resources and it is a focus for innovative applications. We also linked the SAAM artists to the Getty Union List of Artist Names (ULAN®) and to the artists in the Rijksmuseum dataset.

Museums pride themselves in publishing authoritative data, so SAAM personnel manually verified all proposed links before they became part of the dataset. To make the verification process manageable, we sought high-precision algorithms. We matched people using their names, including variants, and their birth dates and death dates. The task is challenging because people's names are recorded in many different ways, multiple people can have the same name, and birth dates and death dates are often missing or incorrect. For technical reasons we did not use other information, although we plan to do so in the future.

Our approach involves estimating the ratio of people in DBpedia having each possible value for the properties we use for matching (e.g., ratio of people born in 1879). For attributes compared using equality (birth/death years), we scan all people in DBpedia counting the number that have each specific value. For dependent attributes such as birth and death year, we also compute the ratios for pairs of values. We compare names using the Jaro-Winkler string metric [4], and for them compute the ratios as follows: we divide the interval $[0, 1]$ in bins of size ϵ, and for each bin we estimate the number of pairs of people whose names differ by a Jaro-Winkler score less than ϵ. Empirically, we determined that $\epsilon = 0.01$ and 10 million samples yield good results in our ground truth dataset.

The matching algorithm is simple. Given a SAAM and a DBpedia person, their matching score is $s = 1 - d * n$ where d is the *date score* and n is the *name score*. If the dates match exactly, d is the fraction of people in DBpedia with those dates. Otherwise, d is the sum of the fractions for all the intervening years. n is the fraction of people in DBpedia whose Jaro-Winkler score is within ϵ from the score between the given pair of people, computed using the estimates discussed above. We use a simple blocking scheme based on the first letter of the name. To increase efficiency, we discard pairs whose birth or death years differ by more than 10 and whose Jaro-Winkler score is less than 0.8.

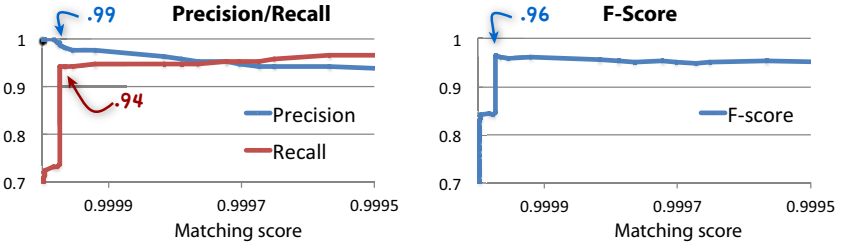


Fig. 5. Precision/Recall and F-score as a function of our algorithm's matching score

[14] http://dbpedia.org

Evaluation. To evaluate our algorithm we constructed ground truth for a dataset of 535 people in the SAAM database (those whose name starts with A). We manually searched in Wikipedia using all variant names and verified the matches using the text of the article and all fields in the SAAM record, including the biography. We found 176 matches in DBpedia.

Figure 5 shows the evaluation results on the ground truth (note that the matching score s decreases from left to right). The highest F-score .96 achieves a precision of .99 and a recall of .94 (166 correct results, 1 incorrect result). At this threshold s^* all names and the years for all but 5 people match exactly. The incorrect result is one where neither date matches, but interestingly, there are 4 results where the years are present but not equal. The sharp increase in recall comes at a score $s > s^*$ where suddenly 37 results for people missing a single date are above threshold (all these are correct results). The next interesting threshold is $\hat{s} = 0.9995$. Between s^* and $\hat{s}$ are 13 results; of these, 4 are correct (2 with non matching names) and 9 incorrect, yielding .938 precision and .966 recall. For $s < \hat{s}$, the density of correct results decreases sharply, containing only 4 correct results in the next 286 candidates. Based on these results, we used $\hat{s}$ as the threshold to match the SAAM data against all DBpedia people (2,807 results), the Getty ULAN (1,759 results) and the Rijksmuseum (321 results).

4 Curating the Linked Data

Museums need the ability to ensure that the linked data they publish are of high quality. The first aspect of the curation process is to ensure that the RDF is correct. Museum personnel can easily browse individual RDF records on the Web, but without understanding the relationship between an RDF record and the underlying database records, it is hard to assess whether the RDF is correct. Karma helps museum personnel understand these relationships at the schema level by graphically showing how database columns map to classes and properties in the ontology (e.g., Figures 3 and 4). Karma also lets users click on individual worksheet cells to inspect the RDF generated for it, helping them understand the relationships at the data level. These graphical views also enabled SAAM personnel and the Semantic Web researchers to communicate effectively while refining the ontology and the mappings. Our goal by the end of the project is that SAAM personnel will use Karma to refine the mappings on their own.

The second aspect of the curation process is to ensure that links to external sources are correct. Our approach is to 1) record the full provenance of each link so that users (and machines) can record links and inspect them when the data sources or the algorithm change, and 2) make it easy for users to review the results of the linking algorithm. We use the PROV ontology[15] to represent provenance data for every link including revisions, matching scores, creation times, author (human or system/version) and data used to produce the link. Users review the links using the Web interface depicted in Figure 6. The interface is a visualization and editor of the underlying PROV RDF records. Each row

[15] `http://www.w3.org/TR/prov-o/`

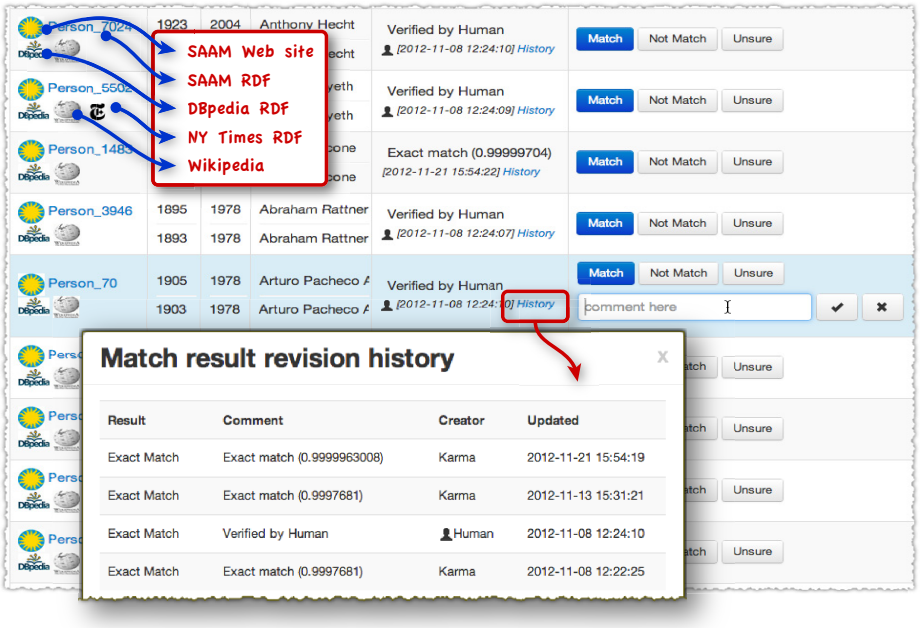


Fig. 6. The KARMA interface enables users to review the results of linking

represents a link. The first cell shows the records being linked: the top part shows links to information about the SAAM record and the bottom part shows links to information for a record in an external source. The next columns show the data values that were used to create the link and information about its revision history. The last column shows buttons to enable users to revise links and provide comments (recorded as PROV records). SAAM personnel used this interface to verify all 2,807 links to DBpedia.

Lesson 7: PROV Is a Suitable Technology for Curating the Links. In addition to supporting the user interface for human verification of links, the PROV representation affords other benefits. We can use SPARQL statements to construct a dataset of owl:same-as triples containing only those links that have been verified by a human (suitable for publication on the Smithsonian Web site) or a dataset containing containing all links with a matching score above a given threshold (suitable for other applications). Similarly, when the underlying data changes (e.g., there is a new version of DBpedia) or a new version of the matching software becomes available, it is possible to retrieve the affected links.

5 Related Work

There has been much recent interest in publishing museum data as Linked Open Data. Europeana[7], one of the most ambitious efforts, published the metadata on 17 million items from 1,500 cultural institutions. This project developed a

comprehensive ontology, called the Europeana Data Model (EDM) and used it to standardize the data that each organization contributes. This standard ontology enables Europeana to aggregate data from such a large number of cultural institutions. The focus of that effort was on developing a comprehensive data model and mapping all of the data to that model. Several smaller efforts focused on mapping rich metadata into RDF while preserving the full content of the original data. This includes the MuseumFinland, which published the metadata on 4,000 cultural artifacts[8] and the Amsterdam Museum [3], which published the metadata on 73,000 objects. In both of these efforts the data is first mapped directly from the raw source into RDF and then complex mapping rules transform the RDF into an RDF expressed in terms of their chosen ontology. The actual mapping process requires using Prolog rules for some of the more complicated cases. Finally, the LODAC Museum published metadata from 114 museums and research institutes in Japan. They defined a relatively simple ontology that consists of objects, artists, and institutions to simplify the mapping process.

In our work on mapping the 41,000 objects from SAAM, we went beyond the previous work in several important ways. First, we developed an approach that supports the mapping of complex sources (both relational and hierarchical) into a rich domain ontology [9]. This approach is in contrast to previous work, which first maps the data directly into RDF [1] and then aligns the RDF with the domain ontology [2]. As described earlier, we build on the EDM ontology, a rich and easily extensible domain ontology. Our approach makes it possible to preserve the richness of the original metadata sources, but unlike the MuseumFinland and the Amsterdam Museum projects, a user does not need to learn a complex rule language and only needs to do a data preparation step to define database views using SQL statements and simple stored procedures.

Second, we performed significantly more data linking than these previous efforts. There is significant prior work on linking data across sources and the most closely related is the work on Silk [14] and the work on entity coreference in RDF graphs [13]. Silk provides a nice framework that allows a user to define a set of matching rules and weights that determine whether two entities should be matched. We tried to use Silk on this project, but we found it extremely difficult to write a set of matching rules that produced high quality matches. The difficulty was due to a combination of missing data and the variation in the discriminability of different data values. The approach that we used in the end was inspired by the work on entity coreference by Song and Heflin [13], which deals well with missing values and takes into account the discriminability of the attribute values in making a determination of the likelihood of a match.

Third, because of the importance to the Smithsonian of producing a high-quality linked data, we developed a curation tool that allows an expert from the museum to review and approve or reject the links produced automatically by our system. Previous work has largely ignored the issue of link quality (Halpin et al. [6] reported that in one evaluation roughly 51% of the same-as links were found to be correct). The exception to this is the effort by the NY Times to map all of their metadata to linked data through a process of manual curation.

In order to support a careful evaluation of the links produced by our system, we developed the linking approach that allows a link reviewer to see the data that is the basis for the link and to be able to drill down into the individual sources to evaluate a link.

6 Conclusions and Future Work

In this paper we described our work on mapping the data of the Smithsonian American Art Museum to Linked Open Data. We presented the end-to-end process of mapping this data, which includes the selection of the domain ontologies, the mapping of the database tables into RDF, the linking of the data to other related sources, and the curation of the resulting data to ensure high-quality data. This initial work provided us with a much deeper understanding of the real-world challenges in creating high-quality link data.

For the Smithsonian, the linked data provides access to information that was not previously available. The Museum currently has 1,123 artist biographies that it makes available on its website; through the linked data, we identified 2,807 links to people records in DBpedia, which SAAM personnel verified. The Smithsonian can now link to the corresponding Wikipedia biographies, increasing the biographies they offer by 60%. Via the links to DBpedia, they now have links to the New York Times, which includes obituaries, exhibition and publication reviews, auction results, and more. They can embed this additional rich information into their records, including 1,759 Getty ULAN® identifiers, to benefit their scholarly and public constituents.

The larger goal of this project is not just to map the SAAM data to Linked Open Data, but rather to develop the tools that will enable any museum or other organization to map their data to linked data themselves. We have already developed the Karma integration tool, which greatly simplifies the problem of mapping structured data into RDF, a high-accuracy approach to linking datasets, and a new curation tool that allows an expert to review the links across data sources. Beyond these techniques and tools, there is much more work to be done. First, we plan to continue to refine and extend the ontologies to support a wide range of museum-related data. Second, we plan to continue to develop and refine the capabilities for data preparation and source modeling in Karma to support the rapid conversion of raw source data into RDF. Third, we plan to generalize our initial work on linking data and integrate a general linking capability into Karma that allows a user to create high-accuracy linking rules and to do so by example rather than having to write the rules by hand.

We also plan to explore new ways to use the linked data to create compelling applications for museums. A tool for finding relationships, like EverythingIsConnected.be [12], has great potential. We can imagine a relationship finder application that allows a museum to develop curated experiences, linking artworks and other concepts to present a guided story. The Museum could offer pre-built curated experiences or the application could be used by students, teachers, and others to create their own self-curated experiences.

Acknowledgements. This research was funded by the Smithsonian American Art Museum. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Smithsonian Institution.

References

1. Bizer, C., Cyganiak, R.: D2R Server–publishing relational databases on the semantic web. Poster at the 5th International Semantic Web Conference (2006)
2. Bizer, C., Schultz, A.: The R2R Framework: Publishing and Discovering Mappings on the Web. In: 1st International Workshop on Consuming Linked Data, Shanghai (2010)
3. de Boer, V., Wielemaker, J., van Gent, J., Hildebrand, M., Isaac, A., van Ossenbruggen, J., Schreiber, G.: Supporting Linked Data Production for Cultural Heritage Institutes: The Amsterdam Museum Case Study. In: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 733–747. Springer, Heidelberg (2012)
4. Cohen, W.W., Ravikumar, P., Fienberg, S.E., et al.: A comparison of string distance metrics for name-matching tasks. In: Proceedings of the IJCAI 2003 Workshop on Information Integration on the Web (IIWeb 2003), pp. 73–78 (2003)
5. Goel, A., Knoblock, C.A., Lerman, K.: Exploiting Structure within Data for Accurate Labeling Using Conditional Random Fields. In: Proceedings of the 14th International Conference on Artificial Intelligence, ICAI (2012)
6. Halpin, H., Hayes, P.J., McCusker, J.P., McGuinness, D.L., Thompson, H.S.: When owl:sameAs isn't the same: An analysis of identity in linked data. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010, Part I. LNCS, vol. 6496, pp. 305–320. Springer, Heidelberg (2010)
7. Haslhofer, B., Isaac, A.: data.europeana.eu - The Europeana Linked Open Data Pilot. In: Multiple Values Selected, The Hague, The Netherlands (July 2011)
8. Hyvonen, E., Makela, E., Salminen, M., Valo, A., Viljanen, K., Saarela, S., Junnila, M., Kettula, S.: MuseumFinland - Finnish museums on the semantic web. Web Semantics: Science, Services and Agents on the World Wide Web 3(2-3) (2005)
9. Knoblock, C.A., Szekely, P., Ambite, J.L., Goel, A., Gupta, S., Lerman, K., Muslea, M., Taheriyan, M., Mallick, P.: Semi-automatically mapping structured sources into the semantic web. In: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 375–390. Springer, Heidelberg (2012)
10. Lafferty, J., McCallum, A., Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: Proceedings of the International Conference on Machine Learning (2001)
11. Matsumura, F., Kobayashi, I., Kato, F., Kamura, T., Ohmukai, I., Takeda, H.: Producing and Consuming Linked Open Data on Art with a Local Community. In: Proceedings of the Third International Workshop on Consuming Linked Data (COLD 2012). CEUR Workshop Proceedings (2012)
12. Sande, M.V., Verborgh, R., Coppens, S., Nies, T.D., Debevere, P., Vocht, L.D., Potter, P.D., Deursen, D.V., Mannens, E., Walle, R.: Everything is Connected. In: Proceedings of the 11th International Semantic Web Conference, ISWC (2012)
13. Song, D., Heflin, J.: Domain-independent entity coreference for linking ontology instances. ACM Journal of Data and Information Quality, ACM JDIQ (2012)
14. Volz, J., Bizer, C., Gaedke, M., Kobilarov, G.: Silk–a link discovery framework for the web of data. In: Proceedings of the 2nd Linked Data on the Web Workshop, pp. 559–572 (2009)

Guiding the Evolution of a Multilingual Ontology in a Concrete Setting

Mauro Dragoni[1], Chiara Di Francescomarino[1], Chiara Ghidini[1], Julia Clemente[2], and Salvador Sánchez Alonso[2]

[1] FBK–IRST, Trento, Italy
[2] Universidad de Alcalá, Alcalá de Henares, Spain
{dragoni,dfmchiara,ghidini}@fbk.eu, julia@aut.uah.es, salvador.sanchez@uah.es

Abstract. Evolving complex artifacts as multilingual ontologies is a difficult activity demanding for the involvement of different roles and for guidelines to drive and coordinate them. We present the methodology and the underlying tool that have been used in the context of the Organic.Lingua project for the collaborative evolution of the multilingual Organic Agriculture ontology. Findings gathered from a quantitative and a qualitative evaluation of the experience are reported, revealing the usefulness of the methodology used in synergy with the tool.

1 Introduction

Ontologies are dynamic entities that evolve over time because they need to reflect changes in the domain they describe, in their conceptualization, or in their specification. As stated in [1], ontology evolution can be defined as "the timely adaptation of an ontology to the arisen changes and the consistent propagation of these changes to dependent artifacts.".

Managing the evolution of ontologies is a complex problem, well known to the Semantic Web community, and a number of efforts are devoted to tackle different aspects of the problem, as described in [2]. The problem becomes even more complex when, besides the evolution of the ontology entities, the changes are related also to the multilingual aspects of the ontology itself. Indeed, in the last years, the construction of multilingual ontologies has become an important objective for organizations working in multilingual environments. Examples are companies which need to apply ontologies to information retrieval on a mass of resources written in different languages, or international bodies using ontologies as a way of describing terminological standards in a particular field (e.g., food, diseases, agriculture, and so on). As described in [3], obtaining multilingual ontologies is a complex activity which requires to tackle a number of problems spanning from the translation of labels and description associated to a given ontology entity to the adaptation of the ontology to a concrete language and cultural community.

These requirements lead to the necessity of guiding the evolution of an ontology according to its final use; moreover, this necessity become stronger when ontologies are used not only for modeling a domain, but also for describing and accessing resources in knowledge repositories or web portals.

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 608–622, 2013.

In this paper we present the methodological and technical solution used for facing the problem of the multilingual evolution of an ontology for the organic agriculture (the *Organic Agriculture* ontology) in the context of the Organic.Lingua project.

The main contributions of the paper are: (i) the definition of a methodology for guiding the evolution in a collaborative environment and its usage in a real setting; and (ii) a quantitative and a qualitative evaluation about the synergistic usage of a semantic web tool and the defined methodology.

2 The Organic.Lingua Project

Organic.Lingua (http://www.organic-lingua.eu) is an EU-funded project that aims at providing automated multilingual services and tools facilitating the discovery, retrieval, exploitation and extension of digital educational content related to Organic Agriculture and AgroEcology. More in concrete, the project aims at providing, on top of a web portal, cross-lingual facility services enabling users to (i) find resources in languages different from the ones in which the query has been formulated and/or the resource described (e.g., providing services for the cross-lingual retrieval); (ii) manage meta-data information for resources in different languages (e.g., offering automated meta-data translation services); and (iii) contribute to evolve the content (e.g., providing services supporting the users in the content generation).

The accomplishment of these objectives is reached in the Organic.Lingua project by means of two components: on the one hand, a web portal offering software components and linguistic resources able to provide multilingual services and, on the other hand, a conceptual model (formalized in the *Organic Agriculture* ontology) used for managing information associated with the resources provided to the final users and shared with other components deployed on the Organic.Lingua platform. In a nutshell, the usage of the *Organic Agriculture* ontology is twofold:

- Resource annotation: each time a content provider inserts a resource in the repository, the resource is annotated with one or more concepts extracted from the ontology. The list of available concepts is retrieved by using an ontology service deployed in the ontology management component (shown in Section 4). Then, this list is exploited for annotating the learning resources published on the Web portal.
- Resource retrieval: when web users perform queries on the system, the ontology is used, by the back-end information retrieval system, to perform advanced searches based on semantic techniques. Moreover, the ontology is used also by the Cross-Language Information Retrieval component for query expansion purposes.

Due to this intensive use of the ontology in the entire Organic.Lingua portal, evolving both the knowledge represented by the artifact, as well as, the linguistic layer, requires a precise methodology, and dedicated tools, for avoiding the loss of effectiveness of the components deployed on the platform. In the next two sections we describe the methodology that we have defined in the context of the Organic.Lingua project and that we propose as a general best practice for the evolution of complex artifacts as the ones used in this project (Section 3) as well as the dedicated collaborative tool that supports the proposed methodology (Section 4).

3 Guiding Evolution with a Scenario-Based Methodology

Collaboratively building, maintaining and evolving multilingual ontologies is not a trivial task: multilinguality, in fact, adds the linguistic problem to classical problems of collaborative modeling such as the background, skills and role differences that may exist between e.g., domain and knowledge experts that collaboratively model the ontology. Facing all these issues together requires appropriate methodologies (and tools) to support the work of the involved experts.

In the Organic.Lingua project, a multi-role scenario-based methodology (MRSB) was proposed and adopted for the *Organic Agriculture* ontology evolution. The methodology relies on involving experts playing different roles and guiding them, step-by-step through tasks and critical scenarios, towards the collaborative evolution of the multilingual ontology.

3.1 MRSB: A Multi-Role Methodology

Three different types of experts are involved in the ontology evolution process: *domain experts*, *language experts* and *knowledge engineers*. Domain experts play a key role as they are in charge of driving the core changes in the ontology; language experts have to revise existing translations and to provide new translations for the newly added terms; finally, knowledge engineers provide a general support for the evolution process and ensure correctness from a formal point of view.

To coordinate all these different experts, the MRSB methodology proposes to guide them step-by-step through tasks and guidelines. In detail:

- domain experts are asked to create or contribute to discussions, suggesting actions to be taken on the ontology and/or commenting on existing issues: they are recommended to clearly state whether all the suggestions should be implemented, if only some and why;
- once reached an agreement, knowledge engineers can be notified and take the final decision: either approve or discard the suggestion and proceed with the update on the English version of the ontology;
- finally, the language experts are asked to provide, check and revise the translations of labels and definitions, often produced by translation services.

3.2 MRSB: A Scenario-Based Methodology

The MRSB methodology proposes to guide domain experts in the ontology evolution through critical scenarios which represent the major types of changes that have arisen from the analysis carried out for the *Organic Agriculture* ontology. Such scenarios address three main categories of activities: the general revision of the ontology (from scenario 1 to 3), the revision of the ontology by taking into account environments in which ontologies are exploited by external tools (scenarios 4 and 5), and, finally, the revision of the linguistic layer. To further guide the experts in the evolution process, the scenarios have been complemented with the results of a rigorous analysis carried out by knowledge engineers on the use of the entities in the existing version of the ontology

(e.g., which terms are less frequently used for searches, how many resources are retrieved by different terms). This choice is motivated by the fact that often ontologies are not used only for representing the knowledge of a particular domain, but they are also exploited for other tasks like the annotation and the retrieval of resources with/through semantic information.

The devised scenarios are the following:

1. *Entity Deprecation Scenario.* A complete analysis of the active version of an ontology, generally performed jointly by the domain experts and the knowledge engineers, may lead to a report containing a set of instances considered unnecessary for the representation of the target domain. Therefore, domain experts are asked to identify all candidate entities for removal and start the relative discussions, with the support of the knowledge engineers, for reaching an agreement about the status of each identified concept.
2. *Ontology Mapping Scenario.* A linguistic analysis of the terms used for defining entities in an ontology, may lead to the consideration that some of the terms used for defining the entities generate ambiguities when a user needs to understand the meaning of the label. This scenario is designed for finding a solution to this problem by asking domain experts to find concepts, in external domain-related knowledge bases (KBs), that may be linked with the ones defined in the ontology that they are revising. Moreover, it is also desirable that, the terminology used for revising the ontology, is the same used in the external knowledge base.
3. *Ontology Enrichment Scenario.* In this scenario, domain experts are asked to complete the ontology by including concepts in those areas that were more poorly covered. For accomplishing this task, domain experts are invited to consult domain-related resources (concept lists, knowledge bases, etc.) for finding new concepts and relationship not previously described in the ontology.
4. *Entity Specialization Scenario.* This scenario addresses the necessity of increasing the granularity of some branches of the ontology. Such an increase is required, not only for the completeness of the ontology, but also when there are concepts used for annotating a huge number of resources, e.g., terms qualifying up to 500 resources or more. Indeed, a huge set of results is not useful in searches, because it does not really help users to find effective results.
5. *Entity Generalization Scenario.* The opposite of what has been described in the previous scenario happens when some concepts have been used for annotating a very small number of resources or are not used at all. In this case, domain experts are appointed to study each particular case in detail and to suggest, eventually, the deprecation for those concepts that lead to the retrieval of few resources. As for the previous scenario, the number of retrieved resources is used as a distinguishing parameter for evaluating which kind of action the domain experts should propose for evolving the ontology.
6. *Entity Translation Scenario.* Considering the usage of multilingual ontologies, the revision of the linguistic layer is important for having a high quality artifact. For instance, when a term is translated by non-language experts, a common error is that the translated term is correct from a language point of view (dictionary-based translation), but it is not the optimum one for the domain described by the ontology.

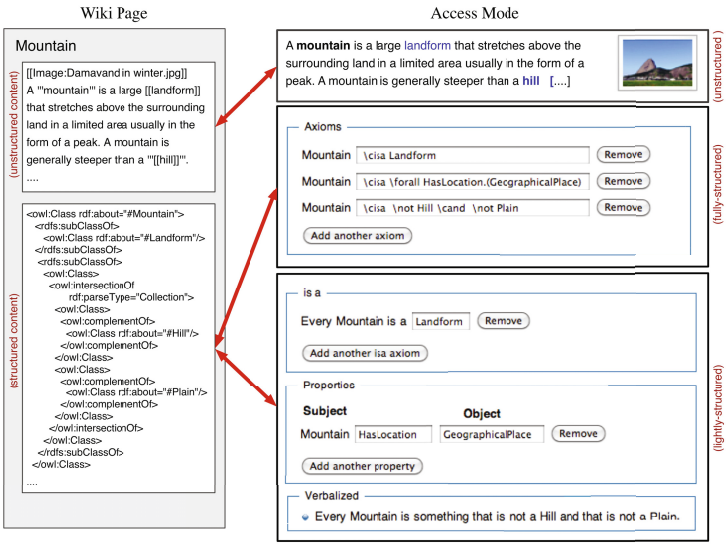


Fig. 1. A page and the access modes in MoKi

Therefore, in this scenario, domain experts and language experts are asked to complete and revise the translations of each entity defined in the ontology by carefully considering the domain described by the ontology.

4 Supporting the Evolution: The MoKi Tool

MoKi[1] is a collaborative MediaWiki-based [4] tool for modeling ontological and procedural knowledge in an integrated manner[2]. MoKi is grounded on three main pillars, which we briefly illustrate with the help of Figure 1:

- each basic entity of the ontology (i.e., concepts, object and datatype properties, and individuals) is associated to a wiki page. For instance, the concept Mountain in Figure 1 is associated to a wiki page which contains its description;
- each wiki page describes an entity by means of both unstructured (e.g., free text, images) and structured (e.g. OWL axioms) content;
- a multi-mode access to the page content is provided to support easy usage by users with different skills and competencies. Figure 1 shows three different access modes, for accessing the unstructured and structured content of the wiki page.

A comprehensive description of MoKi is presented in [5].

In order to meet the specific needs of the Organic.Lingua project, MoKi has been customized with the addition of: (i) multilingual services for the automatic translation of

[1] `http://moki.fbk.eu`

[2] Though MoKi allows to model both ontological and procedural knowledge, here we will limit our description only to the features for building ontologies.

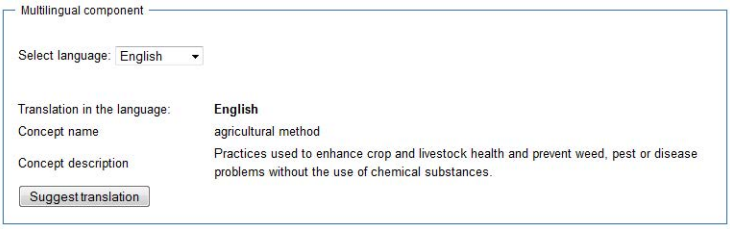


Fig. 2. Multilingual box for facilitating the entity translation

labels and descriptions associated to the ontology entities; and (ii) collaborative features specifically targeting linguistic issues. Translating domain-specific ontologies, in fact, demands that experts discuss and reach an agreement not only with respect to modeling choices, but also to (automated) term translations.

In the context of the Organic.Lingua project, MoKi has been customized with facilities that enable a new type of profile, the Language Expert, to manage the translations carried out on the ontology entities. Such a profile has the role of coordinating the translation activities by approving the terms translation that the other actors involved into the ontology revision process suggest.

The MoKi collaborative nature together with these customizations make it a good technological layer for the application of the multi-role scenario-based methodology.

4.1 Supporting the Different Scenarios with MoKi

In this subsection, we briefly describe the main customizations implemented in MoKi with a particular emphasis on how these customizations specifically address the different scenarios described in Section 3.

Domain and Language Experts View The semi-structured access mode, dedicated to the Domain and Language Experts, has been equipped with functionalities that permit to accomplish the revisions of the linguistic layer. This set of functionalities permits to revise the translations of names and descriptions of each entity (concepts, individuals, and properties).

For facilitating the browsing and the editing of the translations, a quick view box has been inserted into the mask (as it is shown in Figure 2); this way, language experts are able to navigate through the available translations and, eventually, invoke the third-party translation services for retrieving a suggestion or, alternatively, to edit the translation by themselves (Figure 3).

This customization aims to address all scenarios described in Section 3.

Approval and Discussion Facilities. Given the complexity of translating domain specific ontologies, translations often need to be checked and agreed upon by a community of experts. This is especially true when ontologies are used to represent terminological standards which need to be carefully discussed and evaluated. To support this collaborative activity we foresee the usage of the wiki-style features of MoKi, expanded with

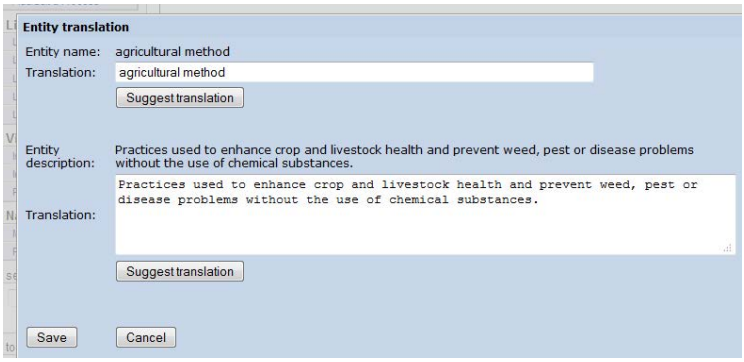


Fig. 3. Quick translation box for editing entities translations

the possibility of assigning specific translations of ontology entities to specific experts who need to monitor, check, and approve the suggested translations. This customization permits to promote the management of the changes carried out on the ontology (in both layers) by providing the facilities necessary to manage the life-cycle of each change.

These facilities may be split in two different sets of features. The first group may be considered as a monitor of the activities performed on each entity page. When changes are committed, approval requests are created. They contain the identification of the expert in charge of approving the change, the date in which the change has been performed, and a natural language description of the change. Moreover, a mechanism for managing the approvals and for maintaining the history of all approval requests for each entity is provided. Instead, the second set contains the facilities for managing the discussions associated with each entity page. A user interface for creating the discussions has been implemented together with a notification procedure that alerts users when new topics/replies, related to the discussions that they are following, have been posted.

This customization aims to address all scenarios described in Section 3.

Quick Translation Feature. For facilitating the work of language experts, we have implemented the possibility of comparing side-by-side two lists of translations. This way, the language expert in charge of revising the translations, avoiding to navigate among the entity pages, is able to speed-up the revision process.

Figure 4 shows such a view, by presenting the list of concepts in the English and Italian translations. At the right of each element of the table, it is placed a link allowing to invoke a quick translation box (as shown in Figure 3) that gives the opportunity to quickly modify information without opening the entity page. Finally, in the last column, it is placed a flag indicating that changes have been performed on that concept, and a revision/approval is requested.

This customization aims to address the *Entity Translation Scenario* described in Section 3.

Ontology Translator Component. This component manages the translation operations required by MoKi. When a translation, for an entity name or description, is requested, the Ontology Translator invokes the external translation services for performing the

Fig. 4. View for comparing entities translations

translation. The component sends the request to the interface exposed by the third-party translation services and, after the retrieval of the result, the representation of the entity is updated with the information coming from the translation services. Further details, about the translation services used by MoKi can be find in [6].

This customization aims also to address the *Entity Translation Scenario*.

Interface and Ontology Multilingual Facilities. In order to complete the set of features available for managing the multilingual aspects of the Organic.Lingua project, MoKi has been equipped with two further components that permit to switch between the languages available for the tool interface, to add a new language to the ontology, and to select the language used for showing the ontology in the different views.

Through these facilities, it is also possible to add a new language to the MoKi interface and to manage the translation of its labels. This module has been implemented on top of the multilingual features of MediaWiki.

Instead, concerning the ontology, when a new language is added to the ontology, the Ontology Translator component described above, is invoked for retrieving, for each entity described in the ontology, the translations related to its labels and descriptions.

Finally, the Ontology Export functionality has been revisited by adding the possibility to choose the export languages, among the available ones.

This customization has not been implemented for addressing a particular scenario, but for improving the usability of the tool in a multilingual context.

Linked Open Data Service. In order to permit the exposure of the ontology artifact to the other components deployed on the Organic.Lingua platform, MoKi has been equipped with a service that exposes entity information by using the Linked Open Data format. Such a service permits to perform operations on the ontology remotely; examples of available remote operations are the retrieval of the entire ontology, or of part of it, or the possibility to edit the ontology e.g., by adding a new translated label. The service

provides a RESTful interface for receiving the requests, while the results are exposed by using the SKOS language [3].

This customization has not been implemented for addressing a particular scenario, but for linking the tool with the other components deployed on the Organic.Lingua platform.

5 The Evaluation

Our goal is evaluating the usage and the usefulness of the MRSB methodology (guiding step by step users through tasks and scenarios) and of the underlying tool, i.e. MoKi, to support different experts in the collaborative evolution of a multilingual ontology. Evolving a multilingual ontology, indeed, adds to the traditional difficulties characterizing the evolution of an ontology, such as the involvement of domain experts (DEs) and their collaboration with knowledge engineers (KEs), also the issues related to the multilinguality, including the need of a third role, the language experts (LEs), and their collaboration with DEs and KEs. In detail, we are interested in answering two main research questions:

RQ1. Is it *useful* guiding step by step through tasks and scenarios the different experts involved in the collaborative evolution of a multilingual ontology?
RQ2. Do the MoKi functionalities provide an *effective* support to the the collaborative evolution of a multilingual ontology?

In order to answer these questions we performed two types of analysis: a quantitative and a qualitative one. In the former, data about the activities carried out by the three categories of experts in the context of the evolution of the *Organic Agriculture* ontology have been analyzed; in the latter, instead, experts have been asked to answer questions aiming at investigating their perception about the usefulness of the MRSB methodology as well as of the MoKi tool in supporting the realization of the different tasks and scenarios foreseen by the MRSB methodology.

Design, Material and Procedure Eleven experts with average experience in their field ranging from 5 to 10 years, were overall involved in the ontology evolution: 3 ontology experts, 4 domain experts and 4 language experts, although one ontology expert and one domain expert also played the role of language experts. The first languages of the 6 LEs were different one from another, thus allowing us to translate the evolved ontology into 6 different languages: Estonian, Spanish, French, Greek, Turkish and Italian.

Most of the experts had no previous knowledge of the tool, hence an initial phase of training was necessary. The training was organized according to the following steps:

- A one-day overall introduction to the tool.
- A few short, on-line, training sessions with the MoKi tool guided by ontology and tool experts, targeted to help domain experts to better understand the capabilities of the tool.

[3] http://www.w3.org/2004/02/skos/

Table 1. Usage of MoKi by the team of experts for accomplishing the multilingual evolution task

Expert Category	Entity Creation	Entity Update	Entity Deletion	Entity Translation	Discussion Creation	Discussion Update
DEs	52	367	15		27	75
KEs	1	50	4		3	11
LEs				629	2	24
total	53	417	19	629	32	110

– Hands-on usage of the tool: domain experts were left to "play" with MoKi in order to become familiar with the functionalities that they would use during the revision process. This exercise also had the secondary objective to collect doubts and problems encountered by experts.

After the initial training, according to the MRSB methodology, the experts were provided with detailed guidelines (including the description of tasks and scenarios) for the multilingual evolution of the *Organic Agriculture* ontology. At the end of their ontology evolution activity, experts were asked to fill a questionnaire aiming at investigating their perception about the methodology and the MoKi tool (ease of use, usefulness and capability to support the different scenarios and tasks required by MRSB). Questions were organized in four main parts: (i) one collecting information on the experts' background; (ii) one about the support provided by the methodology used for guiding them in the ontology evolution; (iii) a third one on the subjects' evaluation about MoKi and the role of its different functionalities for accomplishing the MRSB tasks; and (iv) a last one for retrieving information, impressions and questions related to the work performed for the ontology evolution. Some of the questions were provided in the form of open questions, while most of them were closed questions. The latter type mainly concern the experts' evaluation of the tool usefulness on a scale from 1 to 5, varying according to the target of the evaluation (e.g., 1 = *extremely ease/useful/effective*, ... , 5 = *extremely useless/difficult/ineffective*).

5.1 Quantitative Evaluation Results

We analyzed the data on the usage of MoKi during the phases of the project devoted to the evolution of the *Organic Agriculture* ontology (June - November 2012). Overall each expert spent on average between 8 and 15 hours for accomplishing her work, with peaks of more than 15 hours for DEs. The data are obtained by combining the information stored in the MoKi database and the tool logs. Table 1, reporting the number of the main operations carried out by the team of experts during the ontology evolution phase, shows that the tool has actually been used by all the experts' categories. As reasonable, DEs and KEs have been more involved in the entity editing (creation, update and deletion) and in the discussions (both creation and update), while the LEs have actively participated in the translation activity.

Looking more in detail at the MoKi functionalities exercised by KEs, DEs and LEs, such a trend is overall confirmed (Table 2a). The highest percentage of operations related to the editing and the discussion functionalities has been carried out by DEs,

while LEs actively exercised the multilingual ones. Surprisingly, the browsing and visualization functionalities have been mainly used by LEs, probably feeling the need to translate labels and descriptions after getting a better understanding of their semantics (through browsing and view functionalities). This result confirms our intuition about the importance of allowing all the different categories of experts involved in the collaborative evolution to easily access to and work on the ontology.

Table 2. Usage of the MoKi functionalities per category of experts and topics of discussions

(a)

MoKi Functionality Group	DEs	KEs	LEs
Discussion and Approval	46.4%	21.6%	31.4%
Browsing	39.2%	15.3%	45.5%
Multilingual	33.7%	3.4%	64.2%
Editing	61.4%	7.8%	30.8%
Visualization	42.7%	29.7%	45.8%

(b)

Topic	Percentage
Specialization	35.5 %
Mapping to external KBs	22.6%
Entity Deprecation	19.4%
Ontology Enrichment	16.1%
Entity Moving	3.2 %
Definition Rephrasing	3.2 %

Furthermore, inspecting the topics of the discussions allowed us to get a coarse idea of the methodology usage and effectiveness. Table 2b reports the percentage of discussions carried out by experts, classified according to the topic. The table suggests that the highest percentage relates to scenarios proposed in the methodology. In detail, most of the discussions had as topic the need to specialize existing entities (*Entity Specialization Scenario*), followed by those related to the mapping of the *Organic Agriculture* ontology to external knowledge bases ((*Ontology Mapping Scenario*) and finally by the deletion of (deprecated) entities and addition of (new relevant) entities. The analysis of the discussions also revealed the effectiveness of the MRSB methodology: only 15% of the discussions (5 out of 32) is still open, i.e., no decision has been made yet.

5.2 Qualitative Evaluation Results

To investigate the subjective perception of the 11 experts about the support provided by the methodology and the tool to the multilingual ontology evolution activity, we analyzed the subjective data collected through the questionnaire.

In order to evaluate the statistical significance of the positivity/negativity of the collected results we applied the (one-tailed) Mann-Whitney test [7] verifying the hypothesis that $\tilde{F} \leq 3$, where $\tilde{F}$ represents the median of the evaluations for the factor F and 3 is the intermediate value in the 1 to 5 Likert scale. Moreover, to evaluate whether the results are strongly positive, we also applied the same test for the hypothesis that $\tilde{F} \leq 2$, where 2 is the lower level of positive answer in the 1-5 Likert scale. In this case, a significant outcome would mean that, overall, the obtained results are strongly positive. All the analyses are performed with a level of confidence of 95% (p-value < 0.05), i.e., there is only 5% of probability that the results are obtained by chance.

Figure 5 (left) reports the distribution of the experts' evaluations about the usefulness of scenarios and tasks used in the MRSB methodology for guiding the experts in their work: scenarios have been mostly judged as *absolutely useful*, while 10% of the experts

revealed minor doubts about the task-based approach. By applying the Mann-Whitney test we found that the usage of scenarios has, also at statistical level, been judged as *absolutely useful*, while the usage of tasks has been evaluated as *useful* (with statistical significance of 95%). The fact that the usage of tasks was less appreciated by LEs and KEs can partially justify the result: their work indeed, that is mainly driven by DEs actions, demands for a less intensive task guidance.

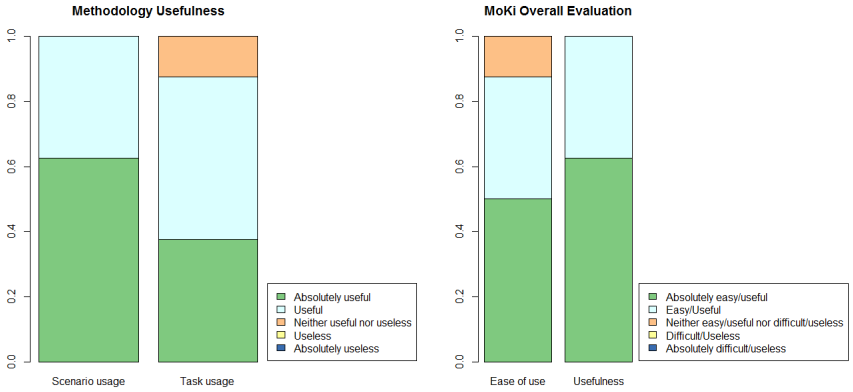


Fig. 5. Experts' evaluation about methodology and MoKi

Figure 5 (right) shows the distribution of the experts' evaluations (on the 5 point scale) of MoKi's ease of use and usefulness for the accomplishment of the ontology evolution activity. Although 10% of the experts showed doubts about the ease of use of the tool, the evaluations are overall positive (*easy*) also at statistical level. Moreover, all the experts recognized the usefulness of the tool, resulting in an overall evaluation of MoKi as *absolutely useful* (also at statistical level). The difficulty in the ease of understanding can be partially explained complementing these results with experts' answers in the open questions: some of the experts asked for a MoKi tutorial. Indeed, while most of the experts have been using MoKi starting from the training phase, a few of them did not immediately practice it, thus finding its usage more difficult later on.

To better understand the relationship between the role of the tool in supporting the methodology used for guiding the experts, we asked the experts to express their evaluation about the effectiveness of the support provided by each typology of functionality to each scenario of the methodology. Table 3 reports the corresponding evaluations according to a 95% statistical significance, e.g., *effective* means that there is only a 5% of possibility that the subjects' evaluations are overall equal or higher than *effective* by chance. Except for the non-convincing support of the browsing functionalities to the *Ontology Mapping Scenario*, all the MoKi functionalities have been evaluated as overall at least *effective* in supporting the MRSB methodology for the multilingual ontology evolution. In detail, the table shows that the discussion functionalities have been considered as *absolutely effective* in supporting the highest number of scenarios. Indeed, many of the evolution scenarios demand for experts' discussions to reach an agreement. The overall *absolute effectiveness* of the browsing functionalities has also been assessed in 2

Table 3. MoKi functionality effectiveness in supporting the MRSB methodology

Functionality typology	Ontology enrichment	Entity deprecation	Entity Specializ	Entity Generaliz	Ontology Mapping	Entity Translation
Discussion and Awareness	*Absolutely effective*	*Absolutely effective*	*Effective*	*Effective*	*Effective*	*Absolutely effective*
Browsing	*Absolutely effective*	*Absolutely effective*	*Effective*	*Effective*	*Neither effective nor ineffective*	*Effective*
Multilingual	*Effective*	*Effective*	*Effective*	*Effective*	*Effective*	*Absolutely effective*
Editing Visualization	*Effective*	*Effective*	*Effective*	*Effective*	*Effective*	*Effective*

out of the 7 scenarios, despite the moderate support for the *Ontology Mapping Scenario*. Finally, as expected, the multilingual functionalities have been evaluated as *absolutely effective* for the *Entity Translation Scenario*.

5.3 Findings and Lesson Learned

The quantitative results reported in Subsection 5.1 show that the MRSB methodology has been actually applied for evolving the *Organic Agriculture* ontology: the experts' discussions, indeed, were mainly guided by scenario-related topics. Combining these results with the preferences expressed by the experts, we can positively answer **RQ1**: the MRSB methodology, guiding users step-by-step via scenarios and tasks, provides a concrete help to experts in the evolution of a multilingual ontology, although experts prefer scenarios to tasks.

Together with the application of the MRSB methodology, the quantitative results also show the actual usage of the tool and of its functionalities by the three categories of experts. Moreover, their positive evaluation about ease of use and usefulness of the tool, as well as about effectiveness of the different functionalities in supporting the methodology suggest a positive answer also for **RQ2**: the MoKi functionalities support the experts in the application of the MRSB methodology for the collaborative evolution of multilingual ontologies.

By further inspecting evaluations and subjects' expertise we found that some relations[4] exist between the evaluations provided by subjects on the effectiveness of the MoKi functionalities with respect to specific scenarios and the typology of expertise of the subject. In particular we found that DEs, differently from the other two categories of experts, perceived the browsing functionalities as more effective in supporting the *Entity Deprecation Scenario* than the other experts. On the contrary, the KEs found more effective the discussion functionalities for the *Entity Specialization Scenario*. These results are inline with the overall evaluations of the experts on the effectiveness of the functionalities in supporting the MRSB methodology, i.e., across all the scenarios. Indeed each group of functionalities got an *effective* average evaluation by all the three categories of experts, except for a higher score by DEs for the browsing functionalities and by KEs for the discussion functionalities.

[4] We applied the Anova statistical test to investigate whether the provided evaluations are influenced by the role of the subject.

Finally, the answers to the open questions provided us with suggestions about possible improvements of MoKi. In particular, besides the need to improve the quality of the translation suggestions, features for enhancing the formatting of content and discussion pages, for better supporting concept mappings to external ontologies and for helping experts in the decision making process have been suggested by experts.

Hence, we can conclude that combining a tool easy to use and provided with useful functionalities as MoKi, with a methodology guiding experts step-by-step through concrete scenarios seem to be a winning strategy to overcome the complexity of a problem mixing the two dimensions: the one related to the multilinguality and the one related to the collaboration of different experts.

6 Related Works

In this Section, we present a brief review of the main ontology management tools applied to support collaborative creation and sharing of ontological knowledge.

Knoodl [5] facilitates community-oriented development of OWL based ontologies and RDF knowledge bases. It also serves as a semantic technology platform, offering a Java service-based interface or a SPARQL-based interface so that communities can build their own semantic applications using their ontologies and knowledge bases.

Protégé [8] is an open source visual ontology editor and knowledge-base framework. Recently, Collaborative Protégé has been released as an extension of the existing Protégé system. It supports collaborative ontology editing as well as annotation of both ontology components and ontology changes. In addition to the common ontology editing operations, it enables annotation of both ontology components and ontology changes. It supports the searching and filtering of user annotations, also known as notes, based on different criteria.

Semantic MediaWiki+ [9], which includes the Halo Extension, is a further extension on Semantic MediaWiki with a focus on enhanced usability for semantic features. Especially, it supports the annotation of whole pages and parts of text, and offers "knowledge gardening" functionalities, that is maintenance scripts at the semantic level, with the aim to detect inconsistent annotations, near-duplicate entries etc.

The tools above support the collaboration between users for the creation and the evolution of ontologies but do not deal with multilingual issues, which have significantly grown in importance during the last years [10]. To testify the importance of multilinguality in the field of ontology engineering a recent example is provided by the Monnet Project [6] that targets the problem of multilingual information access at the semantic level [11]. Its aim, is to define novel models for cross-lingual information access by using semantic web approaches. Concerning tools, the only instrument supporting the management of multilinguality in ontologies is *NeOn* [3]. It is a state-of-the-art, open source multi-platform ontology engineering environment, which provides comprehensive support for the ontology engineering life-cycle. The last version of the toolkit is based on the Eclipse platform and provides an extensive set of plug-ins covering a variety of ontology engineering activities. However, this tool does not provide facilities

[5] http://www.knoodl.com

[6] http://www.monnet-project.eu

for supporting the multi-role collaboration. Thus, as far as we know, MoKi provides the first significant effort to produce a tool that supports the collaborative evolution of multilingual ontologies, by combining features for the support of collaboration and features for the support of multilinguality and translation.

7 Conclusions

In this paper we have presented our experience in applying a scenario-based methodology for modeling complex ontological artifacts composed of a knowledge layer representing the domain, and a linguistic layer making the knowledge available from a multilingual point of view. Such a methodology has been concretely used in the context of the Organic.Lingua EU project with the support of a customized version of the MoKi tool. Three different profiles (domain experts, language experts, and knowledge engineers) evolved the ontology in a collaborative way. Their work together with their subjective evaluation revealed that the synergistic use of the tool and of the methodology permits to evolve the ontology effectively.

References

1. Stojanovic, L.: Methods and Tools for Ontology Evolution. PhD thesis, University of Karlshrue, Karlshrue, Germany (2004)
2. Flouris, G., Manakanatas, D., Kondylakis, H., Plexousakis, D., Antoniou, G.: Ontology change: classification and survey. Knowledge Eng. Review 23(2), 117–152 (2008)
3. Espinoza, M., Gómez-Pérez, A., Mena, E.: Enriching an ontology with multilingual information. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 333–347. Springer, Heidelberg (2008)
4. Wikimedia Foundation: Mediawiki, `http://www.mediawiki.org`
5. Ghidini, C., Rospocher, M., Serafini, L.: Conceptual modeling in wikis: a reference architecture and a tool. In: eKNOW 2012, Valencia, Spain, pp. 128–135 (2012)
6. Dragoni, M., Ghidini, C., Stoitsis, G., Sicilia, M.A., Sanchez-Alonso, S.: Recommendations for revising existing ontologies and schemas. Deliverable D3.1.1 (2011)
7. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in software engineering: an introduction. Kluwer Academic Publishers (2000)
8. Gennari, J., Musen, M., Fergerson, R., Grosso, W., Crubézy, M., Eriksson, H., Noy, N., Tu, S.: The evolution of protégé: an environment for knowledge-based systems development. Int. J. Hum.-Comput. Stud. 58(1), 89–123 (2003)
9. Hansch, D., Schnurr, H.P.: Practical applications of semantic mediawiki in commercial environments - case study: semantic-based project management. In: 3rd European Semantic Technology Conference, ESTC 2009 (2009)
10. Peters, C., Braschler, M., Nunzio, G.D., Ferro, N., Gonzalo, J., Sanderson, M.: From research to application in multilingual information access: the contribution of evaluation. In: LREC. European Language Resources Association (2008)
11. McCrae, J., Spohr, D., Cimiano, P.: Linking lexical resources and ontologies on the semantic web with lemon. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) ESWC 2011, Part I. LNCS, vol. 6643, pp. 245–259. Springer, Heidelberg (2011)

Using BMEcat Catalogs as a Lever for Product Master Data on the Semantic Web

Alex Stolz, Benedicto Rodriguez-Castro, and Martin Hepp

E-Business and Web Science Research Group, Universität der Bundeswehr München
Werner-Heisenberg-Weg 39, D-85577 Neubiberg, Germany
{alex.stolz,bene.rodriguez}@ebusiness-unibw.org, mhepp@computer.org

Abstract. To date, the automatic exchange of product information between business partners in a value chain is typically done using Business-to-Business (B2B) catalog standards such as EDIFACT, cXML, or BMEcat. At the same time, the Web of Data, in particular the GoodRelations vocabulary, offers the necessary means to publish highly-structured product data in a machine-readable format. The advantage of the publication of rich product descriptions can be manifold, including better integration and exchange of information between Web applications, high-quality data along the various stages of the value chain, or the opportunity to support more precise and more effective searches. In this paper, we (1) stress the importance of rich product master data for e-commerce on the Semantic Web, and (2) present a tool to convert BMEcat XML data sources into an RDF-based data model anchored in the GoodRelations vocabulary. The benefits of our proposal are tested using product data collected from a set of 2500+ online retailers of varying sizes and domains.

1 Introduction

Online shopping has experienced significant growth during the last decade. Preliminary estimates of retail e-commerce sales in the US show an increase of 17.3% from the third quarter of 2011 to the third quarter of 2012, while they grew to almost five times 2003 levels, totaling 5.2 percent (57 billion dollars) of the entire retail sales market [15]. These recent statistics indicate a large body of different-sized online stores ranging from major retailers like Amazon, BestBuy or Sears to small web shops offering only tens or hundreds of products. Hence it comes as no surprise that instances of popular commodities are offered by a fairly large number of shopping sites. Many of those online shops maintain databases where they can store information and data to describe their goods. Nonetheless, for site-owners it proves difficult to get hold of rich and high-quality product data for all of their items over time, especially if their specifications originate from product catalogs by different manufacturers. Large size retailers might obtain this information in a semi-automated fashion via some form of catalog exchange format. However, small shop owners might have to enter products and feature data manually. This scenario produces repeated definitions of the same product

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 623–638, 2013.

Table 1. Comparison of product features between manufacturers and retailers

Manufacturer Product Features		Retailer Product Features		Coverage[1]
Samsung LED TV ES6300	89	15	amazon.de	28.09%
		39	notebooksbilliger.de	
		22	conrad-electronics.de	
		24	voelkner.de	
Siemens Kettle TW86103	25	10	amazon.de	23.64%
		22	redcoon.de	
		4	quickshopping.de	
		13	elektro-artikel-shop.de	
Suunto M5 Running Pack	33	12	amazon.de	49%
		3	sportscheck.com	
		1	otto.de	
		15	klepsoo.com	
		8	tictactime.de	

features, but mainly with incomplete, inconsistent and outdated information across various online retailers. Little and inaccurate information about products ultimately hampers the effective matchmaking.

Another source of product data for commodities are their manufacturers. These compile and maintain specifications of all of their products. Typically, their product catalogs are managed in Product Information Management (PIM) systems that can export content to different types of media, e.g. via electronic product catalogs as seen on many manufacturer sites or printed catalogs. PIM systems host essential and core product data also known as *product master data*.

Table 1 presents a simple illustration of the situation using the example of three random products. The table compares the number of features provided by the goods' manufacturers with the features found at a large leading online retailer and other online merchants of various sizes selected arbitrarily via the "Shopping" service of Google Germany[2]. Unless otherwise specified, by "features" we mean structured product specifications (i.e datasheets in tabular form published on the shop pages) without taking into account product pictures, product name and product description. It can be seen that the product data provided across the different sources vary significantly.

To date, product master data is typically passed along the value chain using Business to Business (B2B) channels based on Electronic Data Interchange (EDI) standards such as BMEcat (catalog from the German Federal Association for Materials Management, Purchasing and Logistics[3]) [12]. Such standards can significantly help to improve the automatic exchange of data. However, trading partners still have to negotiate and set up information channels bilaterally, which

[1] "Coverage" = Ratio of average number of retailer features and manufacturer features.

[2] `http://www.google.de/shopping/`

[3] English for "Bundesverband Materialwirtschaft, Einkauf und Logistik e.V. (BME)".

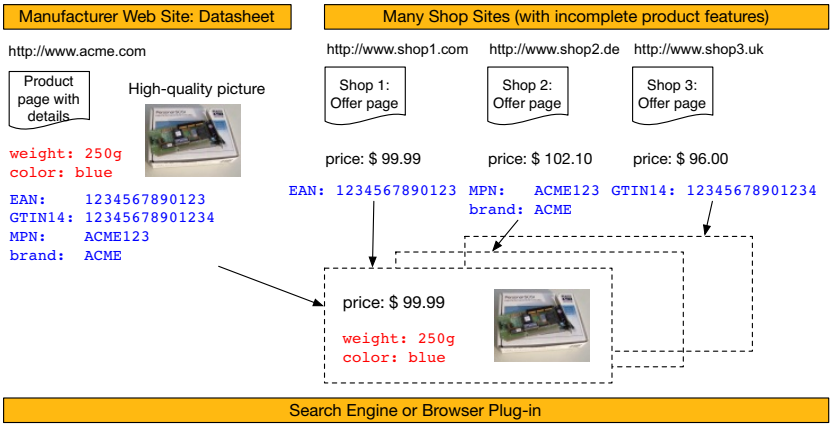


Fig. 1. Lever of manufacturer product master data using *strong identifiers*

prevents them from establishing ad-hoc business relationships and raises the barriers for potential business partners that either do not have the means or the money to connect via imposed B2B standards. Similarly, end users, who could benefit from enterprise data liberalization by facing better search and matchmaking services for products, are neglected [4].

An approach to tackle this issue is to publish rich product master data straight from the Product Information Management (PIM) systems of manufacturers on the Web of Data, so that it can be electronically consumed by other merchants intending to trade these goods. Under this premise, retailers and web shop owners could then rely on widely used product *strong identifiers* such as European/International Article Number (EAN), Global Trade Item Number (GTIN), or Manufacturer Part Number (MPN), to leverage this rich data straight from manufacturers. Fig. 1 illustrates an example of this approach, where three different online merchants benefit from product descriptions and features as published by the manufacturer relying on the corresponding product strong identifier. Each online merchant can then use this rich manufacturer information to augment and personalize their own offering of the product in question.

In this paper, we propose to use the BMEcat XML standard as the starting point to make highly structured product feature data available on the Web of Data. We describe a conceptual mapping and the implementation of a respective software tool for automatically converting BMEcat documents into RDF data based on the GoodRelations vocabulary [9]. This is attractive, because most PIM software applications can export content to BMEcat. With our approach, a single tool can nicely bring the wealth of data from established B2B environments to the Web of Data. Our proposal can manifest at Web scale and is suitable for every PIM system or catalog management software that can create BMEcat XML product data, which holds for about 82% of all of such software systems that we are aware of, as surveyed in [17]. Furthermore, it can minimize the proliferation of repeated, incomplete, or outdated definitions of the same product master

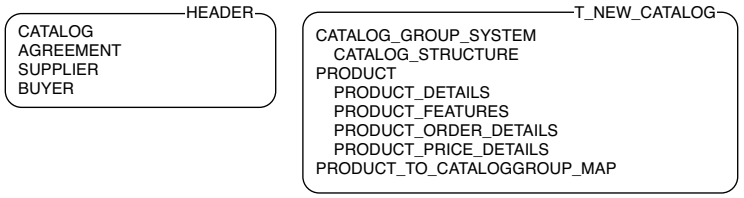


Fig. 2. BMEcat 2005 skeleton

data across various online retailers; by means of simplifying the consumption of authoritative product master data from manufacturers by any size of online retailer. It is also expected as a result that the use of structured data in terms of the GoodRelations vocabulary by manufacturers and online retailers will bring additional benefits derived from being part of the Web of Data, such as Search Engine Optimization (SEO) in the form of rich snippets[4], or the possibility of better articulating the value proposition of products on the Web.

To test our proposal, we converted a representative real-world BMEcat catalog of two well-known manufacturers and analyzed whether the results validate as correct RDF/XML datasets grounded in the GoodRelations ontology. Additionally, we identified examples that illustrate the problem scenario described relying on structured data collected from 2500+ online shops together with their product offerings. Our tests allowed us to confirm the immediate benefits and impact that adopting our approach can bring to both manufacturers and retailers.

2 Conversion from BMEcat to GoodRelations

In this section, we first introduce background information on the BMEcat standard and the GoodRelations vocabulary. Then we present key alignments and challenges underlying the conversion from BMEcat to GoodRelations.

2.1 Background

Both BMEcat and GoodRelations share the goal to facilitate e-commerce transactions and product data exchange between business parties.

BMEcat. BMEcat is a powerful XML standard for the exchange of electronic product catalogs between suppliers and purchasing companies in B2B settings. The current release is BMEcat 2005 [12], a largely downwards-compatible update of BMEcat 1.2. The most notable improvements over previous versions are the support of external catalogs and multiple languages, and the consistent renaming of the ambiguous term *ARTICLE* to *PRODUCT*. Fig. 2 presents a high-level view of the document structure for the transmission of a catalog using BMEcat 2005. A valid BMEcat document comprises a header and a payload section:

[4] `http://support.google.com/webmasters/bin/answer.py?hl=en&answer=99170`

- The header defines global settings such as defaults for currency, eligible regions or catalog language, and specifies seller and buyer parties involved in the transaction. It further may state the agreement or contract that the document is based on. The default values specified in the document header can be overwritten by values defined at product instance level in the document.
- The payload section consists of a product data section and data related to classification standards (e.g. eCl@ss, UNSPSC)[5] or vendor-specific catalog group systems. Product data sections consist of product-related information, feature data, price details, and order details. The element name of the payload part determines the transaction type and can be one of *T_NEW_CATALOG* (new catalog), *T_UPDATE_PRODUCTS* (update of product data), and *T_UPDATE_PRICES* (update of price data).

GoodRelations. GoodRelations [9] is a light-weight vocabulary (ontology, schema, data dictionary) for e-commerce on the Semantic Web. Its expressivity is targeted at the description of an offer and its related entities, i.e. the description of relationships between business entity, offer, and product or service. The ontology provides basic support for the most frequently used properties and individuals in offering descriptions, such as product details, prices, and terms and conditions. The GoodRelations ontology allows to extend products (*gr:SomeItems* or *gr:Individual*) with product models (*gr:ProductOrServiceModel*), or datasheets, that can contribute detailed product information like product features. For that purpose, it provides a fully-fledged meta-model for expressing quantitative and qualitative product properties in OWL. In addition, to further categorize products and to describe them more precisely, GoodRelations allows to extend products and product models with classes and features of comprehensive product classification standards (e.g. eClassOWL [6] or the Product Types Ontology[6]).

To refer to GoodRelations elements in the remainder of this paper, we will use the commonly accepted namespace prefix *gr:*, which can be employed to shorten the full URI of the ontology, i.e. `http://purl.org/goodrelations/v1#name` becomes *gr:name*. Accordingly, we will omit any namespace declarations in text and tabular descriptions.

2.2 Alignments

In the following, we outline correspondences between elements of BMEcat and GoodRelations and propose a mapping between the BMEcat XML format and the GoodRelations vocabulary. Given their inherent overlap, a mapping between the models is reasonable with some exceptions that require special attention. We will highlight these cases, nonetheless we can not cover the full alignment here.

For the mapping between the two schemas the following aspects were considered: Company details (address, contact details, etc.), product offer details, catalog group structures, product features including links to media objects,

[5] `http://www.eclass.de/`, `http://www.unspsc.org/`
[6] `http://www.productontology.org/`

Table 2. Mapping of product details from BMEcat to GoodRelations

BMEcat	**GoodRelations**
PRODUCT	gr:Offering, gr:Individual/gr:SomeItems, gr:ProductOrServiceModel
SUPPLIER_PID type={*ean*, *gtin*}	gr:hasEAN_UCC-13, gr:hasGTIN-14
PRODUCT_DETAILS	
DESCRIPTION_SHORT lang={*en*, *de*, ...}	gr:name with language *en*, *de*, ...
DESCRIPTION_LONG lang={*en*, *de*, ...}	gr:description with language *en*, *de*, ...
INTERNATIONAL_PID type={*ean*, *gtin*}	gr:hasEAN_UCC-13, gr:hasGTIN-14
MANUFACTURER_PID	gr:hasMPN
MANUFACTURER_NAME	gr:hasManufacturer → gr:BusinessEntity → gr:name
PRODUCT_STATUS type={*new*, *used*, ...}	gr:condition

and references to external product classification standards. Furthermore, multi-language descriptions in BMEcat are handled properly, namely by assigning corresponding language tags to RDF literals. An illustrative example of a catalog and its respective conversion is available online[7]. However, in the context of this paper we focus solely on product model data. Also, we do not provide alignments for full classification standards that can be exchanged since BMEcat 2005, primarily because of the complexity and for legal reasons especially gaining in importance when converting licensed classification standards. Moreover, there already exist proposals that focus on the conversion and publication of product classification standards (e.g. eClassOWL [6]).

Product Details. At the center of the proposed alignments are product details and product-related business details. Table 2 shows the BMEcat-2005-compliant mapping for product-specific details. Table 2 adds an additional level of detail to the *PRODUCT* → *PRODUCT_DETAILS* structure introduced in Fig. 2. The element name highlighted in bold font face determines a new nesting level, e.g. *PRODUCT* consists of an attribute for the product identifier of the supplier and a sub-element *PRODUCT_DETAILS*. The elements discussed in the present context are all mapped to properties of product instances, product models and offers in GoodRelations. However, our main interest lies in the alignment to *gr:ProductOrServiceModel*. The product identifier can be mapped in two different ways, at product level or at product details level, whereby the second takes precedence over the other. Whether the European Article Number (EAN) or the Global Trade Item Number (GTIN) is mapped depends on the *type*-attribute supplied with the BMEcat element. Furthermore, the mapping at product level allows to specify the manufacturer part number, product name and description, and condition of the product. Depending on the language attribute supplied along with the *DESCRIPTION_SHORT* and *DESCRIPTION_LONG* elements in BMEcat 2005, multiple translations of product name and description can be

[7] `http://www.ebusiness-unibw.org/projects/bmecat2goodrelations/example/`

Table 3. Mapping of product features from BMEcat to GoodRelations

BMEcat	GoodRelations
PRODUCT_FEATURES	
REFERENCE_FEATURE_SYSTEM_NAME	referenced classification system identifier
REFERENCE_FEATURE_GROUP_ID	rdf:type (class id of classification system)
REFERENCE_FEATURE_GROUP_NAME	gr:category
FEATURE	
FNAME	rdfs:label and property name in GR
FDESCR	rdfs:comment
FVALUE	gr:hasValueFloat
FUNIT	gr:hasUnitOfMeasurement
FREF	feature id of referenced classification system, property name in GR context

Table 4. Mapping of a catalog group system in BMEcat to a *rdfs:subClassOf*-hierarchy

BMEcat	GoodRelations
CATALOG_GROUP_SYSTEM	
CATALOG_STRUCTURE	owl:Class
GROUP_ID	class name of owl:Class
GROUP_NAME lang={*en*, *de*, ...}	rdfs:label with language *en*, *de*, ...
GROUP_DESCRIPTION lang={*en*, *de*, ...}	rdfs:comment with language *en*, *de*, ...
PARENT_ID	rdfs:subClassOf (class id of superclass)

obtained. Lastly, the manufacturer name is mapped to a little more complex pattern in GoodRelations, i.e. the value of *MANUFACTURER_NAME* maps to the name of the legal entity attached to the product model via *gr:hasManufacturer*.

Product Features. BMEcat allows to specify products using vendor-specific catalog groups and features, or to refer to classification systems with externally defined categories and features. The mapping of product classes and features is shown in Table 3. The target GoodRelations property of the *REFERENCE_FEATURE_SYSTEM_NAME* (e.g. ECLASS-5.1) and *REFERENCE_FEATURE_GROUP_ID* have no direct mapping, rather a combination of them unambiguously determines the class identifier of a reference classification system (e.g. eClassOWL [6]). Likewise, the *FREF* element can be used together with *FVALUE* and an optional *FUNIT* element to specify a feature whose property is referenced externally. Otherwise, if no *FREF* is available for a feature, then the feature is defined locally. The *FUNIT* element can be used to discern property types in GoodRelations, i.e. to assign a quantitative object property to the product model in RDF if a value for *FUNIT* is given, otherwise a datatype property. The distinction will be addressed in more detail in Section 2.3.

Catalog Group Systems. Catalog groups are a means to further refine product descriptions. A catalog group system is mapped building up an

rdfs:subClassOf-hierarchy based on the GenTax algorithm [10], which permits to create meaningful ontology classes for a specific context while at the same time preserving the original hierarchy, i.e. the catalog group taxonomy. Table 4 outlines the mapping of catalog groups in BMEcat to RDF. The hierarchy is determined by the group identifier of the catalog structure that refers to the identifier of its parent group.

Product and Catalog Group Map. In order to link catalog groups and products, BMEcat maps group identifiers with product identifiers using *PRODUCT_TO_CATALOGGROUP_MAP*. Accordingly, products in GoodRelations are assigned corresponding classes from the catalog group system, i.e. they are defined as instances (*rdf:type*) of classes derived from the catalog group hierarchy.

2.3 Design Decisions

In the following, we cover aspects of the conversion where the alignment of the two schemas turned out to be challenging.

Datatype versus Object Properties. OWL distinguishes between object properties and datatype properties [1]. The former category describes properties that link between individuals, whereas the latter links individuals with data values (literals), e.g. an entity with a numeric value or a textual description. The GoodRelations vocabulary further refines the categorization made by OWL by discerning qualitative and quantitative object properties. On the other side, BMEcat does not explicitly discriminate types of features, so features (*FEATURE*) typically consist of *FNAME*, *FVALUE* and, optionally, an *FUNIT* element. The presence of the *FUNIT* element helps to distinguish quantitative properties from datatype and qualitative properties, because quantitative values are determined by numeric values and units of measurements, e.g. *150 millimeters* or *1 bar*. Thus, any other feature is either a qualitative or a datatype property.

It is impossible to reliably discern qualitative properties and datatype properties in an automated way during conversion (e.g. are S, M, and L qualitative values describing garment sizes or rather simple literal values?), so we reserve this task for solving in the RDF world (potentially bringing in additional knowledge) and declare all such properties as datatype properties with a range of type string.

For those features whose values likely qualify as boolean values we provide a simple heuristic, i.e. if the feature value is one of “y”, “n”, “yes”, “no”, “true”, or “false”, then the property is assumed to be a boolean datatype property. Similarly, all rules that apply to properties also apply to their respective values, i.e. a quantitative property implies quantitative values, and so forth.

Float Value Ranges in Datatype Properties. Unlike GoodRelations, BMEcat does not allow to model range values by definition. There are two possibilities to model them in BMEcat, though. Either the BMEcat supplier defines two separate features, or the range values are encoded in the *FVALUE* element of the feature. The first option defines a feature for the lower range value and a feature for the upper range value, respectively. The downside of this approach is that two unrelated GoodRelations properties arise. The second alternative, i.e. range values encoded as single feature values, leads to invalid values (e.g. *gr:hasValueFloat "10-20"^^xsd:float*) when mapped to GoodRelations. For that reason, typical value patterns describing upper and lower ranges (like operating temperature of "5-40" degrees Celsius) are mapped to *gr:hasMinValueFloat* and *gr:hasMaxValueFloat* of quantitative values in GoodRelations. This approach, however, works only for common encoding patterns for range values in text.

Units of Measurement. BMEcat and GoodRelations recommend to use UN/CEFACT [14] common codes to describe units of measurement. In reality, though, it is common that suppliers of BMEcat catalogs export the unit of measurement codes as they are found in their PIM systems. Instead of adhering to the standard 3-letter code, they often provide different representations of unit symbols, e.g. *cm*, *centimeters*, etc. in place of CMT, which would be the correct UN/CEFACT code. This is inconvenient with regard to potential applications that should consume the data and compare products upon feature descriptions.

2.4 Implementation

The implementation of the logic behind the alignments to be presented herein resulted into the BMEcat2GoodRelations tool. BMEcat2GoodRelations is a portable command line Python application to facilitate the conversion of BMEcat XML files into their corresponding RDF representation anchored in the GoodRelations ontology for e-commerce. Due to the limited length of this paper, we refer readers to the project landing page hosting the open source code repository[8], where they can find a detailed overview of all the features of the converter, including a comprehensive user's guide.

3 Evaluation

To evaluate our proposal, we implemented two use cases that allowed us to produce a large quantity of product model data from BMEcat catalogs. We tested the two BMEcat conversions using standard validators for the Semantic Web, presented in Section 3.1. Then we compare the product models obtained from one of the BMEcat catalogs with products collected from Web shops through a focused Web crawl. Finally, we show the potential leverage of product master data from manufacturers with regard to products offered on the Web.

[8] `http://code.google.com/p/bmecat2goodrelations/`

3.1 Validation of Use Cases

We tested our conversion using BMEcat files from two manufacturers, one in the domain of high-tech electronic components (Weidmüller Interface GmbH und Co. KG[9]), the other one a supplier of white goods (BSH Bosch und Siemens Hausgeräte GmbH[10]). In the case of Weidmüller, the conversion result is available online[11]. The products in the BSH catalog were classified according to eCl@ss 6.1, whereas Weidmüller provide their own proprietary catalog group system. This allowed us to validate the BMEcat converter comprehensively. Although the conversions completed without errors, still a few issues could be detected in each dataset that we will cover subsequently.

To validate the output of our conversion, we used publicly available online and offline validators. In addition to that, our converter prints helpful warning messages to the standard output. In summary, the converter was tested using the following validation steps: (1) BMEcat2GoodRelations converter output (including error and warning messages, if any), (2) RDF/XML syntax validity[12], (3) Pellet validation[13] for spotting semantic, logical inconsistencies, and (4) GoodRelations-specific compliance tests[14] to spot data model inconsistencies.

The converter has built-in check steps that detect common irregularities in the BMEcat data, such as wrong unit codes or invalid feature values. In Table 5, we list a number of warning messages that were output during the conversion of the BMEcat files, together with the validation results of the different validation tools. As shown in the table, the two conversions pass most validation checks, with a few data quality issues reported by some validators. In the BSH catalog for example, some fields that require floating point values contain non-numeric values like "/", "0.75/2.2", "3*16", or "34 x 28 x 33.5", which originates from improper values in the BMEcat. Another data quality problem reported is the usage of non-uniform codes for units of measurement, instead of adhering to the recommended 3-letter UN/CEFACT common codes (e.g. "MTR" for meters, "VLT" for Volt, etc.).

3.2 Missing Product Features on the E-Commerce Web of Data

Table 1 in the introduction showed how the number of features published by manufacturers does not always end up in the descriptions of the offerings published by online retailers. In this section, we elaborate on a complementary example that uses structured data on the Web of Data.

In addition to the manufacturer BMEcat files, we took a real dataset obtained from a focused crawl whereby we collected product data from 2629 shops. The dataset has a slight bias towards long-tail shops. Furthermore, the Web

[9] http://www.weidmueller.com/
[10] http://www.bsh-group.com/
[11] http://catalog.weidmueller.com/semantic/sitemap.xml
[12] http://www.rdfabout.com/demo/validator/, http://www.w3.org/RDF/Validator/
[13] http://clarkparsia.com/pellet/
[14] http://www.ebusiness-unibw.org/tools/goodrelations-validator/

Table 5. Validation of BMEcat conversions

Validation	**BSH**	**Weidmüller**
BMEcat2GoodRelations converter	warnings: (a) wrong values where numeric values were expected; (b) non-standard unit codes detected	warnings: (a) non-standard unit codes detected
RDF Validator	valid. warning: invalid lexical value for literal	valid
W3C RDF Validation	valid	valid
Pellet	valid. warning: malformed xsd:float detected	valid
GoodRelations Validator	step 32 failed: non-compliance of float literal with xsd:float	valid

shops were not crawled entirely. Nonetheless, Fig. 3 illustrates the distribution of the product count across shops for a snapshot of the crawl. To remove any potential bias caused by multiple definitions of the same product on different pages (because of non-canonical URIs containing query strings like `prod_id=1&sess_id=XYZ`), the boxplot was generated using the count of products with distinct EANs per shop. The upper quarter of shops offer more than 493 products according to Fig. 3. More interestingly, half of the shops offer less than 89 distinct products, whereas one quarter of the shops have less than 14 products. This could be explained either by the fact that several shops are rather small and provide only a limited set of offers, or by the non-comprehensive crawl of shop domains.

In Table 6, we complement the example given in the introduction with insights from our collected data. The products listed in the table represent product models from the BSH dataset and product instances from Web shops based on overlapping EANs. In the current dataset, there exist 95 of such matches based on EANs. The comparison of the amount of properties from the manufacturer with the number of properties from the retailers shows a significant gap. For instance, take the vacuum cleaner (German: *Bodenstaubsauger*) in row 2 of Table 6. It shows 30 product properties coming from the manufacturer and an average number of nine properties across the three shops that offer the product. Therefore, the properties in the shops only amount to a fraction (30%) of the properties available from the manufacturer. The relatively constant number of properties for product instances may be explained by the shop extensions that typically only express standard features like product name, GTIN, EAN, SKU, product weight and dimensions. Although this might to a certain extent explain the numbers, it does not change our premise that structured product master data is still lacking on the Web.

We collected all the data in an SPARQL-capable RDF store and extrapolated some statistics to substantiate the potential of our approach. The number of product models in the BSH was 1376 with an average count of 29 properties,

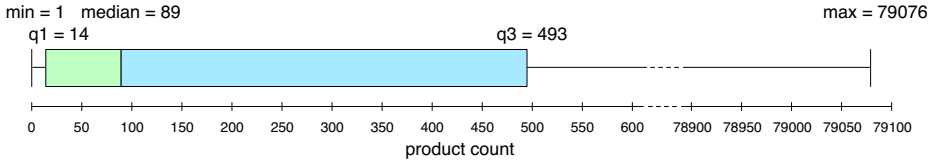


Fig. 3. Boxplot of distribution of product count across Web shops

Table 6. Product features in BSH BMEcat and retailers publishing GoodRelations

BSH Product Features		Retailer Product Features		Coverage[15]
TW86103 Wasserkocher (EAN: 4242003535615)	25	10	marketplace.b2b-discount.de	40%
Bodenstaubsauger Beutel VS06G2410 2400 W (EAN: 4242003356364)	30	10	www.ay-versand.de	30%
		9	www.megashop-express.de	
		8	fairplaysport.tradoria-shop.at	
Mikrowelle HF25M5L2 Edelstahl (EAN: 4242003429303)	51	7	www.european-gate.com	13.73%

while the Weidmüller BMEcat consisted of 32585 product models with 47 properties on average created by our converter. By contrast, the nearly 2.7 million product instances from the crawl only contain eleven properties on average.

3.3 Potential Leverage of Product Master Data on the Web

Table 6 from Section 3.2 confirmed the scenario presented in Table 1 in the introduction in the context of BSH product models and a sample of 2500+ online shops that provide structured data.

In this section, we present some specific examples of the number of online retailers that could readily benefit from leveraging our approach. To remain in the scope of the use cases discussed, the examples are chosen from the BSH BMEcat products catalog, within the German e-commerce marketplace.

We chose to check for the number of shops offering products using a sample size of 90 random product EANs from BSH BMEcat. The sample size was selected based on a 95% confidence level and 10% confidence interval (margin of error), i.e. requiring a minimum of 90 samples given the population of 1376 products in the BMEcat. Using the sample of EANs, we then looked up the number of vendors that offer the products by entering the EAN in the search boxes on Amazon.de, Google Shopping Germany, and the German comparison shopping site preissuchmaschine.de[16]. This gave us a distribution of shops grouped by EAN as outlined in the boxplots in Fig. 4.

[15] "Coverage" = Ratio of average number of retailer features and BSH features

[16] `http://www.amazon.de/`, `http://www.google.de/shopping/`, `http://www.preissuchmaschine.de/`

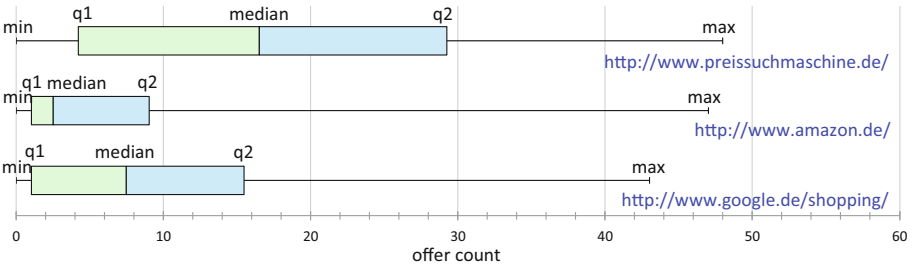


Fig. 4. Boxplots of distribution of shop offers per European Article Number (EAN)

The numbers we got from this experiment were lower than expected. For example, the maximum number of sellers offering a specific product was 48. For half of the products that we tested at least 16 offers appeared in the price comparison search engine. In the Amazon.de and Google Shopping Germany marketplaces by comparison, the number of offers for a product among the sample of product EANs was even lower. We can think of various explanations for this, namely that the marketplace regulations try to limit competition among market participants and, more importantly, that adding products to the marketplace presents a barrier to smaller shop owners (in the case of Google Shopping, a shop is asked to upload product data using a populated product feed or an API). Furthermore, the small numbers may be due to (1) localized searches (*.de*-domain), (2) the fact that shops rarely populate their products with EAN identifiers, or (3) the type of products in our sample, in this case from the domain of white goods that are likely not that popular for being sold online. More precisely, unsupported small shop owners may not find it very attractive to sell dishwashers online given the effort involved for logistics.

To put Fig. 4 (boxplots) in perspective, we did a comparison with a more popular product, i.e. "Canon PowerShot A2300 schwarz" (with EAN "8714574578828"). We repeated the above searches with the same online services, but now using (a) the EAN of this digital camera and (b) the product name, suspecting that many retailers do not populate their products with EAN but use other strong identifiers instead. Amazon.de and preissuchmaschine.de constantly gave 45 and 233 results, respectively. Google Shopping Germany, however, returned only 4 results when searching by the EAN number, but 144 results for a search by product name. These results indicate that using a combination of different types of strong identifiers could leverage product master data on the Semantic Web.

4 Related Work

The rise of B2B e-commerce revealed a series of new information management challenges in the area of product data integration [5,13]. Separately, the gradual realization of the Semantic Web vision has motivated significant efforts aimed

at representing existing e-commerce product related data and classification standards adopting open semantic technologies and data models [7,8,2].

Yet, in the context of managing *product master data* in particular, two previous solutions stand out [3,16] based on their similarities with respect to our problem scenario. The study in [3] presents a meta-model in OWL DLP (which expressivity profile lies between OWL 1 Lite and OWL 1 DL) as part of a semantic application framework that can provide semantic capabilities to a generic PIM system. On the other hand, [16] has developed an extension that allows lifting the data from existing relational databases of leading Master Data Management (MDM) systems into RDF format. This allows semantic interoperability across organizations' core data, applications and systems.

Both solutions share our reliance on Semantic Web technologies to facilitate product master data integration and consistency across separate data sources. However, there are several aspects where they deviate from our proposal as presented in the sections above, most notably: (a) their scope focuses on closed corporate environments which may involve proprietary applications or standards rather than open technologies at the scale of an open Web of Data; and (b) being aimed at generic PIM and MDM systems, their level of abstraction is very broad, introducing additional degrees of separation with respect to the applicability to the problem scenario targeted by the BMEcat2GoodRelations converter tool.

In that sense, BMEcat2GoodRelations is to the best of our knowledge the only solution developed with open standards, readily available to both manufacturers and retailers to convert product master data from BMEcat into structured RDF data suitable for publication and consumption on the Web of Data.

5 Conclusions and Outlook

The proliferation of online retailers in recent years was accompanied by a growing number of products being offered on the Web. Such a substantial increase of online goods introduces new data management challenges. More specifically, it involves how information, in particular products, features or descriptions, can be processed by stakeholders along the product life cycle. Our experience after a survey of 2500+ different-sized online merchants indicates that in the current conditions product data suffers from incomplete, inconsistent or outdated information.

As a partial solution to mitigate the shortage of missing product master data in the context of e-commerce on the Web of Data, we propose the BMEcat2GoodRelations converter. This ready-to-use solution comes as a portable command line tool that converts product master data from BMEcat XML files into their corresponding OWL representation using GoodRelations. All interested merchants have then the possibility of electronically publishing and consuming this authoritative manufacturer data to enhance their product offerings relying on widely adopted product *strong identifiers* such as EAN, GTIN, or MPN.

We argue that the construction of a firm basis of product master data is the prerequisite for useful matchmaking scenarios. The data we collected and

analyzed, provides enough evidence to motivate on the one hand a critical mass of manufacturers to release their product master data and on the other hand retailers to attach strong identifiers to their products. The immediate impact would be a huge lever for enriching online offers by product features and less effort to be put into data cleansing thanks to a gain in more high-quality data. Both factors would pave the way to more granular data analysis and search experience for organizations and individuals.

Acknowledgments. The authors would like to thank Mark Mattern, who provided a first mapping from BMEcat to GoodRelations as part of a master thesis supervised by Martin Hepp [11]. The work on this paper has been supported by the German Federal Ministry of Research (BMBF) by a grant under the KMU Innovativ program as part of the Intelligent Match project (FKZ 01IS10022B), and by the Eurostars program (within the EU 7th Framework Program) of the European Commission in the context of the Ontology-based Product Data Management (OPDM) project (FKZ 01QE1113D).

References

1. Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A.: OWL Web Ontology Language Reference. Tech. rep., World Wide Web Consortium (2004), `http://www.w3.org/TR/owl-ref/`
2. Beneventano, D., Montanari, D.: Ontological Mappings of Product Catalogues. In: Shvaiko, P., Euzenat, J., Giunchiglia, F., Stuckenschmidt, H. (eds.) OM. CEUR Workshop Proceedings, vol. 431. CEUR-WS.org (2008)
3. Brunner, J.S., Ma, L., Wang, C., Zhang, L., Wolfson, D.C., Pan, Y., Srinivas, K.: Explorations in the Use of Semantic Web Technologies for Product Information Management. In: Proceedings of the 16th International Conference on World Wide Web, pp. 747–756. ACM, New York (2007)
4. Di Noia, T., Di Sciascio, E., Donini, F.M., Mongiello, M.: A System for Principled Matchmaking in an Electronic Marketplace. In: Proceedings of the 12th International Conference on World Wide Web, pp. 321–330. ACM, New York (2003)
5. Fensel, D., Ding, Y., Omelayenko, B., Schulten, E., Botquin, G., Brown, M., Flett, A.: Product Data Integration in B2B E-Commerce. IEEE Intelligent Systems 16(4), 54–59 (2001)
6. Hepp, M.: eClassOWL: A Fully-Fledged Products and Services Ontology in OWL. In: Poster Proceedings of the 4th International Semantic Web Conference, Galway, Ireland (2005)
7. Hepp, M.: Products and Services Ontologies: A Methodology for Deriving OWL Ontologies from Industrial Categorization Standards. International Journal on Semantic Web and Information Systems 2(1), 72–99 (2006)
8. Hepp, M.: ProdLight: A Lightweight Ontology for Product Description Based on Datatype Properties. In: Abramowicz, W. (ed.) BIS 2007. LNCS, vol. 4439, pp. 260–272. Springer, Heidelberg (2007)
9. Hepp, M.: GoodRelations: An Ontology for Describing Products and Services Offers on the Web. In: Gangemi, A., Euzenat, J. (eds.) EKAW 2008. LNCS (LNAI), vol. 5268, pp. 329–346. Springer, Heidelberg (2008)

10. Hepp, M., de Bruijn, J.: GenTax: A Generic Methodology for Deriving OWL and RDF-S Ontologies from Hierarchical Classifications, Thesauri, and Inconsistent Taxonomies. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 129–144. Springer, Heidelberg (2007)
11. Mattern, M.: Transforming BMEcat Catalogs into Semantic Web Annotation Data for Offerings. Master thesis, University of Innsbruck, Innsbruck, Austria (2009)
12. Schmitz, V., Leukel, J., Kelkar, O.: Specification BMEcat 2005. Bundesverband Materialwirtschaft, Einkauf und Logistik e.V., Frankfurt am Main, Germany (2005)
13. Stonebraker, M., Hellerstein, J.M.: Content Integration for E-Business. In: Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data, pp. 552–560. ACM, New York (2001)
14. United Nations Economic Commission for Europe (UNECE): Recommendation No. 20: Codes for Units of Measure Used in International Trade. UN/CEFACT Information Content Management Group (2006)
15. United States Census Bureau: Quarterly Retail E-Commerce Sales: 3rd Quarter 2012. U.S. Department of Commerce, Washington, DC, USA (2012)
16. Wang, X., Sun, X., Cao, F., Ma, L., Kanellos, N., Zhang, K., Pan, Y., Yu, Y.: SMDM: Enhancing Enterprise-Wide Master Data Management Using Semantic Web Technologies. Proc. VLDB Endow. 2(2), 1594–1597 (2009)
17. Weber, A.: Marktanalyse von Software für Produkt-Informations-Management (PIM). Bachelor thesis, Universität der Bundeswehr München, Neubiberg, Germany (2011)

Ontology-Supported Document Ranking for Novelty Search

Michael Färber*

Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe, Germany
michael.faerber@kit.edu

Abstract. Within specific domains, users generally face the challenge to populate an ontology according to their needs. Especially in case of novelty detection and forecast, the user wants to integrate novel information contained in natural text documents into his/her own ontology in order to utilise the knowledge base in a further step. In this paper, a semantic document ranking approach is proposed which serves as a prerequisite for ontology population. By using the underlying ontology for both query generation and document ranking, query and ranking are structured and, therefore, promise to provide a better ranking in terms of relevance and novelty than without using semantics.

Keywords: Document ranking, Ontology-based information extraction, Novelty detection, Semantic similarity.

1 Motivation

The existence and steady growth of the Web has granted us vast amounts of web documents in which contained information can be discovered and utilised for certain information needs. Some of the existing information extraction (IE) techniques make use of background information provided by Semantic Web ontologies. In the past, various ontology-based information extraction (OBIE) systems have been proposed, where ontologies are used within the IE process. Although there exist quite a lot of notable ontologies, in many application areas appropriate ontologies are, due to domain-specificity, too small and, hence, need to be populated in terms of adding instances and properties. For ontology population, it is a crucial task to find new textual information which is relevant to the domain expert, but has not been stored in the knowledge base (KB) and, therefore, has been made usable. In this work, we focus on the worthwhile interplay between an existing KB and a text document corpus, which – in case of the use case of trend detection – is created on demand.

Within the area of ontology population, we propose a novel approach for document ranking in the context of structural search for "novel" items in text documents. We claim that semantics can be used to rank documents according to their expected novel items contained therein.

* Work leading to this paper has been partially supported by the German Ministry of Education and Research (BMBF) under grant no. 02PJ1000.

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 639–644, 2013.

2 Related Work

Our approach is part of an ontology population system with the task of finding relevant and novel information and integrating it into a – e.g., company wide – KB. There are already many OBIE systems [1]. However, concerning novelty search on documents, current approaches show only little [2] or no semantic components [3, 4], although semantics can resolve inconsistencies and ambiguities. Existing approaches are subject to different definitions of novelty and different application areas and granularities. Within the TREC "novelty track" in 2002–2004 [5], systems for detecting novelty were designed. However, the task took place on sentence level, was limited to event and opinion detection, and was aligned for non-domain-specific texts such as news. A similar case is the novelty detection task of the Text Analysis Conference (TAC) Knowledge Base Population (KBP) track [6]. Li and Croft [2] address the field of novelty formalisation in depth. Under the semantic point of view, they merely make use of a low-key named entity recognition and classification (NERC) component and primarily rely on statistical patterns. Zhang et al. [4] regard the challenge of novelty and redundancy detection as a filtering process. Documents are filtered at first according to relevance to the topic, and in a second step according to novelty defined as non-redundancy with respect to previously seen documents. Contrary to systems like "Newsjunkie" [3], we face domain-specific documents like technical reports and patents, and therefore do not have to deal with the problem of analysis of huge amounts of articles in a very short time period, known as "burst of novelty".

Besides the novelty aspect, our work touches upon the research area of query generation as well as graph comparison techniques and similarity metrics. Work here [7–10] might show good results for query suggestion or expansion techniques. Our novel approach, however, uses an underlying ontology as a bridge for both query generation and document ranking.

Last but not least, Aleman-Meza et al. [11] and several researchers at the TAC KBP track [6] whose task it was to find property values in documents (called "slot filling") provide a document ranking approach which also exploits named entities (NEs) found in documents. In the first case, a weighting schema is proposed, where domain experts need to assign weights to the edges between classes of the KB schema in order to model the relatedness. The existence of huge ontologies like DBLP and many different data sources is assumed here. Contrary to this assumption, we want to populate our own, rather small, domain-specific ontology with instances and properties and need to take novelty detection into consideration.

3 Proposed Approach

Given our own KB with instances and schema, our goal is to search for documents and to rank them, so that the documents most novel to the KB and relevant to the query and to the KB have the highest ranking. In the overall OBIE system

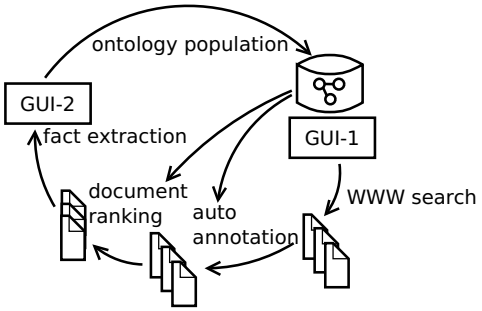


Fig. 1. According to a user's context a structured query is generated with the help of an underlying ontology. After the creation of a document corpus (using the query in an unstructured fashion), annotation and ranking of documents are performed. In a further step, on which we do not focus on, the annotations are verified by the user and used for populating the ontology. In succeeding search rounds search is based on the enriched ontology.

in which our approach is embedded, a second step follows in which the user is able to import phrases marked in the document into his/her KB as property values. Figure 1 gives an overview of the interplay between an ontology and document texts with potentially novel information. In the following, we describe our ranking approach.

Assuming that we have a pre-defined KB schema with assigned weights on the edges expressing the strengh of relatedness and some instances, we start our search by defining the search query – and, hence, the query graph – by the user and his/her context. Besides instances and property values from the KB, additional search keywords can be defined by the user. After expanding the query graph with neighbouring entities of the KB (or neighbouring instances of merely the targeted entity type), we can transform the query graph plus additional keywords into a keyword phrase for simple document search, getting a crawled set of web documents. Of course, we can also operate on a fixed document collection, although this would hamper the overall goal of getting external novel information like in the use case of trend detection and forecast.

As the extended query graph is a subgraph of the KB instance graph and each instance has a fixed set of possible properties, we can find out which relationships (i) between instances and property values and (ii) between instances and other instances of the KB exist and which are still missing. To include the "real" filling degree in terms of personalised importance or novelty degree, we use the weights of the edges in the KB schema graph. By means of the KB, we construct for each document a graph containing all instances found as NEs in the focused document and their relationships among these instances read from the KB. According to further features such as the frequency of the found NEs, additional weights can be assigned to the nodes in the document graphs. For each document,

we can compute a final score compliant with the local severity of found instances in the document, with their novelty degree (inverse filling degree), and with the actual weighting of edges in the KB schema graph. New detected NEs and string matches are also included.

The documents are ranked according to the document scores they obtained. Furthermore, we can use implicit user feedback in the following way: If the user determines which properties or instances are important and novel in the focused document, the weights in the KB schema graph between the classes of the instances (or properties) which were found in the document are adapted. By this means, we can defer to the personal views what relationships between certain classes and properties (or other classes) are of great significance.

The proposed ranking approach is geared to the need of having an approach for ranking documents as a prerequisite for the ontology population task. This involves the inclusion of the novelty aspect into ranking and the adaptation of context and user-dependent association weights between classes.

4 Implementation and Research Methodology

The proposed framework of ontology supported novelty search is currently under development, so that experiments and evaluation could not be performed yet. As use case we chose technology companies, since they are interested in technology forecasts and novelty detection. The lightweight use case ontology consists of classes like technology, company, product, and person. For a valid and comparable evaluation, we plan to evaluate our approach also on a non-specific domain, using the AQUAINT collection, which consists of newswire articles, as used in the TREC 2005 HARD track. Here, DBpedia will be used as underlying KB.

Annotation is done by the wikify service of the Wikipedia Miner [12]. We adopt ideas from wikifier, but adapt it to specific domains, by using the content of our domain-specific semantic-based wiki. In order to detect also new entities, property values, and relationships, we use GATE[1], a well-established rule-based framework.

Our research focuses on semantic document ranking. We implement and plan to evaluate a ranking score function as proposed above. Concerning our domain-specific use case, the final evaluation will be done by students and experts in companies. During the evaluation, we compare the approach of manually assigning weights to the edges in the schema graph with the approach of learning weights. Possible evaluation scenarios entail: 1. We measure whether the users need less time to find a specific amount of relevant and novel documents in comparison to the time they needed in case of using generic search engines like Google. 2. We can also determine whether more relevant and novel documents were found in a specific time interval. This is the main aim of innovation partners in companies and serves as practical motivation.

[1] `http://gate.ac.uk`

5 Conclusion and Prospects

Semantic-based solutions for document ranking do not regard novelty as a criterium so far. In this work, a new ranking approach is proposed. It is designed to improve document retrieval, since users generally face the problem of being commited to review too many text documents containing irrelevant or already known information. With the help of the proposed ranking schema, the more relevant and potentially novel information a document contains, the higher it is ranked and, hence, more likely to be worth reading and the more useful for ontology population. The next steps will involve the implementation and valid evaluation of the semantic ranking approach. In the medium term, we plan to integrate our work into a theoretical foundation like Markov random models.

References

1. Wimalasuriya, D.C., Dou, D.: Ontology-based information extraction: An introduction and a survey of current approaches. Journal of Information Science 36(3), 306–323 (2010)
2. Li, X., Croft, W.B.: An information-pattern-based approach to novelty detection. Information Processing & Management 44(3), 1159–1188 (2008)
3. Gabrilovich, E., Dumais, S., Horvitz, E.: Newsjunkie: providing personalized newsfeeds via analysis of information novelty. In: Proceedings of the 13th International Conference on World Wide Web, WWW 2004, pp. 482–490. ACM, New York (2004)
4. Zhang, Y., Callan, J., Minka, T.: Novelty and redundancy detection in adaptive filtering. In: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2002, pp. 81–88. ACM, New York (2002)
5. Soboroff, I., Harman, D.: Novelty detection: the TREC experience. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT 2005, pp. 105–112. Association for Computational Linguistics, Stroudsburg (2005)
6. Ji, H., Grishman, R., Dang, H.T.: Overview of the TAC2011 Knowledge Base Population Track (2011)
7. Bendersky, M., Metzler, D., Croft, W.B.: Effective query formulation with multiple information sources. In: Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, WSDM 2012, pp. 443–452. ACM, New York (2012)
8. Meij, E., Bron, M., Hollink, L., Huurnink, B., de Rijke, M.: Learning Semantic Query Suggestions. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 424–440. Springer, Heidelberg (2009)
9. Bendersky, M., Croft, W.B.: Modeling higher-order term dependencies in information retrieval using query hypergraphs. In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2012, pp. 941–950. ACM, New York (2012)

10. Bendersky, M., Metzler, D., Croft, W.B.: Parameterized concept weighting in verbose queries. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, pp. 605–614. ACM, New York (2011)
11. Aleman-Meza, B., Arpinar, I.B., Nural, M.V., Sheth, A.P.: Ranking Documents Semantically Using Ontological Relationships. In: Proceedings of the 2010 IEEE Fourth International Conference on Semantic Computing, ICSC 2010, pp. 299–304. IEEE Computer Society, Washington, DC (2010)
12. Milne, D., Witten, I.H.: Learning to link with wikipedia. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, pp. 509–518. ACM, New York (2008)

Semantic Web for the Humanities

Albert Meroño-Peñuela*

Department of Computer Science, VU University Amsterdam, NL
albert.merono@vu.nl

Abstract. Researchers have been interested recently in publishing and linking Humanities datasets following Linked Data principles. This has given rise to some issues that complicate the semantic modelling, comparison, combination and longitudinal analysis of these datasets. In this research proposal we discuss three of these issues: representation round-tripping, concept drift, and contextual knowledge. We advocate an integrated approach to solve them, and present some preliminary results.

Keywords: Semantic Web, Formats, Concept Drift, Contexts.

1 Motivation and Research Questions

Humanities researchers have been interested recently in publishing and linking their datasets following Linked Data principles, in order to enhance their decentralization, openness, changeability and integration. Traditionally, the unique demands of the Humanities, their limited technical and modelling interests, and the highly contextualized nature of their source materials have kept this field distant from the Semantic Web.

We make efforts to bridge the gap. As case studies, we convert Dutch historical censuses (1795-1971) and the catalogue of publications in the Netherlands during the Golden Age (STCN, 16th century onwards) to RDF [10], we model them using standard vocabularies, and we publish them on the Web.

These datasets are messy and heterogeneous. Different dataset versions contain inconsistent structuring rules, concepts with a changing meaning over time, and multiple representation formats. Comparison, combination and longitudinal queries (e.g. *evolution on the number of shoemakers in Amsterdam from 1795 to 1971*) are notoriously difficult. Researchers are forced to manually rewrite data and queries, incurring in high labour costs and non repeatable practices.

Figure 1 shows data heterogeneity interacting with other indicators. Since our goal is to increase data integration, data heterogeneity has to lower, as shown by arrows and signs. Lowering data heterogeneity is no trivial task, and we identify *format round-tripping*, *concept drift* and *contextual knowledge* as influencing indicators that can indirectly improve data integration.

Format Round-Tripping. Lots of data formats are used to encode semistructured datasets. Tools for legacy conversion between these formats are required: Humanities researchers use non RDF compatible tools, and providing data in

* Advisors: Stefan Schlobach and Frank van Harmelen

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 645–649, 2013.

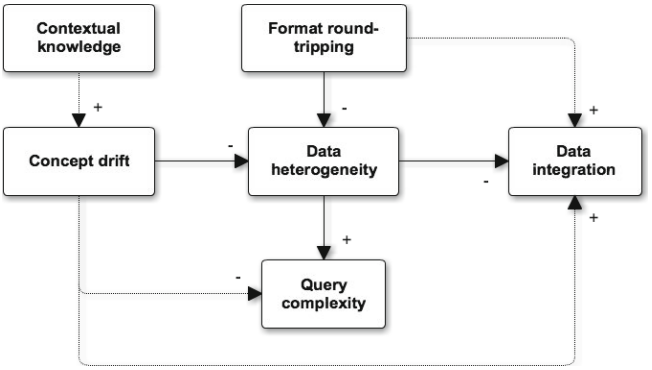


Fig. 1. Indicators influence each other as indicated by arrows. The increase of any given indicator increases/decreases, respectively, the one influenced by it as indicated by the +/- signs. E.g., increasing format round-tripping decreases data heterogeneity.

various formats on demand is a requirement. Under this topic we will investigate, first, how to perform any plain, tabular, tree-based, graph-based or relational-based format conversion from a holistic point of view and, second, whether the original data can be retrieved after arbitrary conversions.

Concept Drift. Different versions of the same dataset show that concepts change their meaning over time, especially if the time gap is wide. Although not meaning exactly the same, two time gapped instances of the same concept may preserve some degree of sameness. For example, the concept *shoemaker* in the 17th century (someone who makes shoes with leather) has drifted until nowadays (someone who owns a company). Mapping drifted concepts correctly is necessary to solve longitudinal queries in Humanities data.

Contextual Knowledge. Humanities ontologies require dynamic concept formalizations instead of static ones, especially for contested, open-textured or ambiguous concepts. The definition of such concepts needs to be dynamically built depending on their contexts. Examples of contextual knowledge are the time when and the space where the concept occurs, subjective opinions on the concept, or domain expert statements about the concept. Multiple contradictory definitions may need to coexist in one ontology.

Concept drift and knowledge from contexts are closely related. Since the context of a concept often changes over time, a definition of concept drift based on the varying properties in contexts can be established. Despite less connected, formats often define metadata describing dataset contextual information, which needs to be appropriately modelled. We realize these phenomena are not exclusive to the Humanities, and this proposal looks further on solving longitudinal analyses in dynamic domains of any kind.

We define a general goal of *providing algorithms, formalisms and tools to disambiguate, clean, prepare, normalize, transform, link and query Humanities datasets, conforming a framework for effective Humanities data publishing in the Semantic Web.* Under this umbrella, our research questions are:

1. Can RDF data models faithfully represent the Humanities sources? Is an RDF-based format round-tripping framework possible?
2. How can we model concept drift? Can drifted concepts be aligned?
3. Can we infer dynamic concept definitions from explicitly formalized contexts? Can these contexts help solving concept drift?

2 State of the Art

Work has been developed on translating RDF, spreadsheet formats and relational databases. Conversion from relational databases to RDF is covered by [7,13], and the W3C has developed a standard (R2RML) for this purpose. Some tools like D2RQ allow accessing relational databases as virtual RDF graphs. Translating RDF backwards to the relational model is developed in [12] under some assumptions. Conversion between spreadsheet formats and RDF is also possible [6,10]. Google Refine is a power tool for working with messy data and generic format translations, with plugins supporting RDF.

Concept drift in the Semantic Web has been studied in [14], where the authors establish a theory for concept drift defining the meaning of a concept in terms of its intension, extension and labeling. Other Semantic Web approaches have used conceptual clustering [2] or concept signatures [5] to detect concept drift. In Description Logics, ontology diff [3] can be used to determine meaning differences. The question has been discussed in Philosphy around the confrontation of history of unit-ideas versus a pure linguistic intellectual history [9].

Some work has been done recently with respect to contexts in the Semantic Web, although they emphasize the specific goals of improving data integration [4] or speeding up reasoning [11]. Rule interchange languages for the Semantic Web like RIF are also related to dynamic concept construction [8].

3 Proposed Approach

Format Round-Tripping. Existing approaches on format conversion pair any data format with RDF and perform a forward or backward transformation between the two. Our proposal is to take an holistic approach, studying the expressivity of these languages and checking whether arbitrary translation workflows are possible. We are interested in round-tripping translation paths to check if original representations can be regenerated without data loss. We aim at canonical RDF graph forms [1] and centric RDF data representations.

Concept Drift. We will study what precise relationship holds between two different versions of a changing concept, identifying the presence of a drift and its nature. Using Description Logics work on ontology diff [3], we will define a minimum meaning concept core, which keeps stable over time despite other non essential transformations. A data model to represent drifted concepts will be needed. A systematic comparison between unstable concept properties will tell whether a drift occurred, and its type. We consider discussions on history of unit-ideas [9] and theories of concept drift for the Semantic Web [14] as inspiration.

Contextual Knowledge. We will study how concepts can be dynamically defined depending on their graph contexts. To solve contextual knowledge questions we aim at a two step process. First, we target an explicit semantic representation of the context of a concept, and we will use various data models and vocabularies to define contexts. Second, we consider inference for deriving logical consequences from previously selected contextual graphs. This process can be further integrated with our concept drift framework.

4 Research Methodology

We establish an iterative workflow that runs the proposed topics in parallel, first developing theories and then proposals. Proposals will be evaluated with at least the two Humanities case-studies referred in Section 1. All models and automated methods will be validated by domain experts. At the end of each iteration, resulting design methods will be scaled up and refined.

5 Results and Future Work

Regarding format round-tripping, we implemented some preliminary tools[1,2]. `TabLinker` is a MS Excel to RDF converter supporting translation of annotations and interactive user defined mappings. We also developed scripts generating RDF from various semistructured data formats. We plan to evaluate round-tripping by comparing an original file with its circularly translated homologue. With respect to concept drift, a first set of mappings between possibly label-drifted concepts have been defined using label similarity functions. We run simple longitudinal queries with `MP2Demo`, relying on an hybrid top-down/bottom-up approach that combines upper ontologies (e.g. Historical International Standard Classification of Occupations, HISCO) with automatically extracted local ontologies.

Three more yearly iterations will be carried out. Format round-tripping will be generalized from current scripts, defining transformation entities that will abstract specific format dependencies to modelling artifacts. We will create a data model for concept drift and an RDF/OWL simulation framework to test it with ontology diff and intension, extension and labeling functions. We will extend this framework to integrate reasoning with contexts.

6 Conclusion

In this research proposal we motivate the problems of format round-tripping, concept drift, and contextual knowledge in the context of a Humanities enabled Semantic Web. We propose an approach with novel perspectives extending the state of the art, and we describe an iterative research method to sort these issues out. Finally, we show work that has been done during the first year iteration, and we establish a plan for the remainder.

[1] `http://github.com/Data2Semantics/`
[2] `http://github.com/CEDAR-project/`

Acknowledgements. The work on which this paper is based has been partly supported by the Computational Humanities Programme of the Royal Netherlands Academy of Arts and Sciences, under the auspices of the CEDAR project. For further information, see `http://ehumanities.nl`.

References

1. Carroll, J.J.: Signing RDF Graphs. Tech. Rep. HPL-2003-142, HP Lab (2003)
2. Fanizzi, N., d'Amato, C., Esposito, F.: Conceptual Clustering and Its Application to Concept Drift and Novelty Detection. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 318–332. Springer, Heidelberg (2008)
3. Gonçalves, R.S., Parsia, B., Sattler, U.: Analysing multiple versions of an ontology: A study of the NCI Thesaurus. In: Proceedings of the 24th International Workshop on Description Logics, DL 2011 (2011), `http://ceur-ws.org/Vol-745/`
4. Guha, R., McCool, R., Fikes, R.: Contexts for the Semantic Web. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 32–46. Springer, Heidelberg (2004)
5. Gulla, J.A., et al.: Semantic Drift in Ontologies. In: Filipe, J., Cordeiro, J. (eds.) WEBIST 2010, Proceedings of the 6th International Conference on Web Information Systems and Technologies, pp. 13–20. INSTICC Press (2010)
6. Han, L., Parr, C., Sachs, J., Joshi, A.: RDF123: a mechanism to transform spreadsheets to RDF. Tech. rep., University of Maryland, Baltimore County (2007)
7. Korotkiy, M., Top, J.L.: From Relational Data to RDFS Models. In: Koch, N., Fraternali, P., Wirsing, M. (eds.) ICWE 2004. LNCS, vol. 3140, pp. 430–434. Springer, Heidelberg (2004)
8. Krisnadhi, A., Maier, F., Hitzler, P.: OWL and rules. In: Polleres, A., d'Amato, C., Arenas, M., Handschuh, S., Kroner, P., Ossowski, S., Patel-Schneider, P. (eds.) Reasoning Web 2011. LNCS, vol. 6848, pp. 382–415. Springer, Heidelberg (2011)
9. Kuukkanen, J.M.: Making Sense of Conceptual Change. History and Theory 47, 351–372 (2008)
10. Meroño-Peñuela, A., et al.: Linked humanities data: The next frontier? A case-study in historical census data. In: Proceedings of the 2nd International Workshop on Linked Science (LISC 2012). International Semantic Web Conference, ISWC (2012), `http://ceur-ws.org/Vol-951/`
11. Peñaloza, R., Baader, F., Knechtel, M.: Context-Dependent Views to Axioms and Consequences of Semantic Web Ontologies. Web Semantics: Science, Services and Agents on the World Wide Web 12 (2012)
12. Ramanujam, S., et al.: R2D: Extracting Relational Structure from RDF Stores. In: Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, WI-IAT 2009, vol. 01, pp. 361–366. IEEE Computer Society, Washington, DC (2009)
13. Sequeda, J.F., Arenas, M., Miranker, D.P.: On directly mapping relational databases to RDF and OWL. In: Proceedings of the 21st World Wide Web Conference, WWW 2012, pp. 649–658. ACM, New York (2012)
14. Wang, S., Schlobach, S., Klein, M.: What Is Concept Drift and How to Measure It? In: Cimiano, P., Pinto, H.S. (eds.) EKAW 2010. LNCS, vol. 6317, pp. 241–256. Springer, Heidelberg (2010)

Maintaining Mappings Valid between Dynamic KOS

Julio Cesar Dos Reis

CR SANTEC, CRP Henri Tudor and LRI, University of Paris-Sud XI
6, Av. des Hauts-Fourneaux, L-4362 Esch-sur-Alzette, Luxembourg
julio.dosreis@tudor.lu

Abstract. Knowledge **O**rganization **S**ystems (KOS) and the existing mappings between them have become extremely relevant in semantic-enabled systems especially for interoperability reasons. KOS may have a dynamic nature since knowledge in a lot of domains evolves fast, and thus KOS evolution can potentially impact mappings, turning them unreliable. A still open research problem is how to adapt mappings in the course of KOS evolution without re-computing semantic correspondences between elements of the involved KOS. This PhD study tackles this issue proposing an approach for adapting mappings according to KOS changes. A framework is conceptualized with a mechanism to support the maintenance of mappings over time, keeping them valid. This proposal will decrease the efforts to maintain mappings up-to-date.

Keywords: mapping evolution, mapping adaptation, mapping maintenance.

1 Introduction

Knowledge **O**rganization **S**ystems (KOS) aim at encompassing all types of conceptual models for organizing knowledge [1] as, for example, semantic networks, ontologies, taxonomies and thesauri. In various contexts and domains, such as the Semantic Web (SemWeb) and Bioinformatics, it is necessary to have a combined use of different KOS, since a unique KOS is not able to cover the totality of a domain due to its size and complexity. Mappings representing semantic correspondences between elements belonging to different KOS therefore need to be established.

The highly dynamic aspect of the knowledge leads to frequent KOS changes. Klein [3] proposed a first categorization of changes, which can affect ontologies, dividing them into atomic and complex operations. The first refers to the change of only a single specific element (*e.g.*, concepts, attributes and relationships) while the second denotes operations that are composed of multiple atomic ones. The impact of these changes on mappings associated to KOS has not been deeply studied. Actually, KOS evolution challenges the reliability of dependent artifacts such as mappings, in the sense that changes affecting KOS elements may invalidate existing mappings. This requires mappings to be adequately maintained over time. Nevertheless, how to adapt mappings impacted by KOS evolution as automatic as possible, without re-computing the whole set of mappings each time a KOS evolves, is still an open research problem. Many research questions arise in the context of this problem: (1) How to perform

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 650–655, 2013.

mapping adaptation taking the way KOS evolve into account? (2) What information regarding mappings and KOS evolution is necessary to support the mapping adaptation? (3) How to correlate different types of KOS changes with actions suited to adapt mappings? (4) How might the different types of semantic relations of mappings be taken into consideration?

Maintaining mappings valid over time is crucial since various applications may rely on them [2]. In the SemWeb context, for instance, up-to-date mappings could allow more trustable semantic searches over integrated ontologies in the Web [4]. In other domains, such as the biomedicine, mappings are very important to support data integration among different applications [5]. Usually, hundreds of thousands of mappings are explored by applications such as the Unified Medical Language System (UMLS). Therefore, after releasing new KOS versions, re-computing the whole set of mappings is a time-consuming task demanding huge efforts of validation.

The aim through this PhD study is to define a framework coping with the mapping maintenance problem between dynamic KOS. The proposed approach developed in the framework is to adapt mappings relying on the exploitation of information derived from KOS evolution, combined with information coming from existing mappings. We aim at considering different types of semantic relations ($=, \leq, \geq, \approx$) in mappings.

The remainder of this article is organized as follows: Section 2 presents the state-of-the-art. Section 3 describes the proposed approach including the research methodology and the evaluation method. Section 4 presents the results achieved so far and the future work envisaged.

2 The State of the Art

Although significant research efforts in the past years have coped with issues related to ontology evolution [6], the understanding of the impact of this evolution in dependent artifacts such as mappings has received very little attention. We organize the different approaches coping with the maintenance of mappings in two main categories. The first category tackles the problem by re-calculating mappings. The most naïve approach is the full re-calculation of the set of mappings, which does not consider any information from KOS changes or mappings. Nowadays, there is a high frequency of new KOS versions, and usually the rate of KOS evolution does not justify a full re-calculation [7]. A partial re-calculation approach was proposed by Khattak *et al.* [8] re-creating only those mappings associated to ontology elements which had changed. Matching algorithms are used to perform a new alignment between those changed elements and the whole target ontology. However, the size of KOS still challenges the compromise between precision and recall of available techniques for mapping calculation [2]. Partial re-calculation slightly minimizes the efforts of validation.

The second category concerns approaches attempting to adapt mappings after KOS evolution. KOS changes are usually used to support adaptation of mappings without performing re-calculation. The first propositions appeared in the context of database schema mappings [9]. For ontologies, Martins & Silva [10] propose that evolution of

mappings should behave similarly with the strategies applied for ontology evolution. More recently, aiming at better understanding mapping evolution, Groß *et al.* [11] empirically investigated the evolution of life sciences ontology mappings. In fact, it is still unknown how to fully perform mapping adaptation as automatic as possible according to KOS evolution. The influence of KOS changes on how mappings should change deserves deeper investigations and various research problems remain open. For instance, considering the change in the semantic relation type of a mapping as a possible mapping adaptation action is still an issue. It is also crucial to conduct further investigations to better understand the impact of complex changes operations (*e.g.,* split and merge of concepts) on mappings for their adequate adaptation.

3 Approach and Methodology

This research relies on the hypothesis that there is a correlation between changes affecting KOS elements and the evolution of their associated mappings, which has been observed in experimental studies. In this sense KOS evolution shall be well described for supporting the adaptation of mappings. This is the characterization of a refined categorization of underlined KOS (complex) change operations (the most fine-grained types of KOS changes) containing information judged important for the adaptation of mappings. We determine that as *Change Patterns* (CPs) in a way to recognize different behaviours of changes between KOS versions and a richer context to adapt mappings. Different types of split complex operations are examples of CPs. These are expressed as distinct types of atomic change operations (*i.e.*, addition and removal of KOS elements) as well as KOS complex change, including whenever possible, information regarding the semantic and structural impact of these changes.

We have identified different behaviours of complex changes such as split and merge of concepts. These behaviours are recognized according to a categorization of semantic similarity shared between concepts in a change operation. We also consider how involved concepts in the change are structurally organized. For instance, whether merged concepts were related through an 'is_a' relationship or whether they were sibling concepts. These aspects are further explored for the mapping adaptation.

The proposition is to adapt mapping elements such as the source element of the mapping, and/or the type of its semantic relation supported by information from the CPs that have affected the mapping combined to information coming from mappings. *Mapping Adaptation Actions* are proposed representing different strategies of mapping adaptation to change the adequate mapping elements, for instance, to adapt mappings associated to a removed concept transferring them to parent or sibling concepts (two different actions). In order to know the most appropriate action to be taken for each mapping independently, CPs information and identified elements used to establish the mapping including its semantic status are taken into account. These must represent the necessary conditions to model in which situation an adaptation action shall be applied. *Heuristics* in the proposal accounts for the modeling and the formalization of these conditions, thus expressing the correlations between information from mappings and CPs with the adaptation actions.

As an example, if a deletion of an attribute affects a concept and this attribute was identified as crucial for establishing an associated mapping of the concept concerned, then such mapping is removed. Also, if a complex change like a split of concept was identified resulting in new sub concepts (an example of CP), and an early mapping with the relation of less general type (≤) was associated to the old unsplit concept, then the sub-concepts may inherit this mapping, keeping the same semantic type. Note that how the mapping is adapted is dependent on the structural organization of the concepts in the change combined with information from the mapping.

The research methodology conducted firstly observe empirically the evolution of various KOS from the biomedical domain, and the way different types of KOS (complex) change have impacted the behavior of existing official mappings. The proposed approach in the framework is grounded on the results of these experiments. A further and deeper analysis on them serves also for the definition and refinement of CPs and *Heuristics*. Finally, the framework for mapping evolution is formally defined and a software tool implemented for evaluation purposes. In the evaluation method we aim at comparing the adapted mappings, as outcome of applying the proposed framework, with mappings generated by approaches totally based on matching techniques. Different measures will be observed regarding, for instance, the adaptation actions used, the quantity of mapping candidates involved and their semantic correctness. A qualitative evaluation of adapted mappings will also be conducted with experts of the domain.

4 Preliminary Results and Future Work

Empirical Basis. We have empirically studied the impact of KOS evolution on mappings by observing the evolution of official mappings between biomedical KOS. We investigate different aspects of changes in KOS elements aiming at understanding the correlations between KOS (complex) changes and how mappings are adapted. Different cases of mapping evolution are considered in the context of KOS complex changes, observing their influence on the changes applied in mappings. Initial results highlight that mappings cannot be adapted according to high level or general types of changes only, but that it is rather necessary to consider fine-grained information on the affecting KOS changes and mappings. Results have also pointed out that it is feasible to have correlations between KOS changes with actions adapting mappings.

The DyKOSMap Framework. We have organized the proposed components of the approach into an initial version of the DyKOSMap framework [7]. Figure 1 presents the components and how they are related one to another. The identification of CPs (1) uses two different versions of a same KOS as input, and a set of aspects is designed to describe and recognize CPs. We aim to determine the instances of CPs that took place between two KOS versions. The mapping evolution mechanism must select (2) the appropriate *Mapping Adaptation Action* having as input the current mappings and the set of identified instances of CPs. We perform that supported by the *Heuristics* to know the most adequate actions to apply on impacted mappings. In the last step up-to-date mappings and their history are generated (3) as outcome.

Future Work. It involves the refinement of CPs and their identification between KOS versions reusing software tools already available for this purpose. We aim at defining, formalizing and implementing the *Mapping Adaptation Action* and *Heuristics* computationally. The prototype for mapping evolution shall be developed. Finally, the evaluation will be conducted assessing the results provided by testing the prototype.

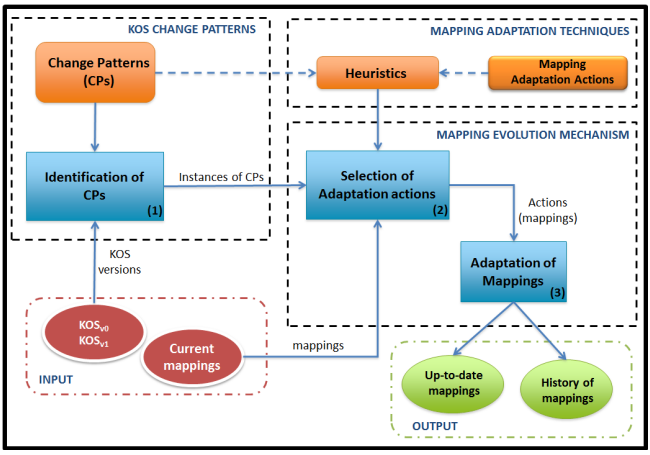


Fig. 1. The DyKOSMap framework for supporting mapping evolution

Acknowledgments. FNR of Luxembourg (grant C10/IS/786147 - DynaMO project).

References

1. Hodge, G.: Systems of Knowledge Organization for Digital Libraries: Beyond Traditional Authority Files. The Digital Library Federation - Council on Library and Information Resources, Washington, DC (2000)
2. Pavel, S., Euzenat, J.: Ontology Matching: State of the Art and Future Challenges. IEEE Transactions on Knowledge and Data Engineering 25, 158–176 (2013)
3. Klein, M.: Change Management for Distributed Ontologies, PhD Thesis, Vrije University (2004)
4. Kitamura, Y., Segawa, S., Sasajima, M., Tarumi, S., Mizoguchi, R.: Deep Semantic Mapping between Functional Taxonomies for Interoperable Semantic Search. In: Domingue, J., Anutariya, C. (eds.) ASWC 2008. LNCS, vol. 5367, pp. 137–151. Springer, Heidelberg (2008)
5. Lambrix, P., Strömbäck, L., Tan, H.: Information Integration in Bioinformatics with Ontologies and Standards. In: Bry, F., Maluszynski, J. (eds.) Semantic Techniques for the Web. LNCS, vol. 5500, pp. 343–376. Springer, Heidelberg (2009)
6. Hartung, M., Terwilliger, J., Rahm, E.: Recent advances in schema and ontology evolution. In: Schema Matching and Mapping, pp. 149–190. Springer (2011)
7. Dos Reis, J.C., Pruski, C., Da Silveira, M., Reynaud-Delaître, C.: Analyzing and Supporting the Mapping Maintenance Problem in Biomedical Knowledge Organization Systems. In: SIMI Workshop - 9th Extended Semantic Web Conference, pp. 25–36 (2012)

8. Khattak, A.M., Pervez, Z., Latif, K., Lee, S.: Time Efficient Reconciliation of Mappings in Dynamic Web Ontologies. Knowledge-Based Systems 35, 369–374 (2012)
9. Velegrakis, Y., Miller, J., Popa, L.: Preserving mapping consistency under schema changes. VLDB Journal 13, 274–293 (2004)
10. Martins, H., Silva, N.: A User-Driven and a Semantic-Based Ontology Mapping Evolution Approach. In: 11th ICEIS, Milan, pp. 6–10 (2009)
11. Gross, A., Hartung, M., Thor, A., Rahm, E.: How do computed ontology mappings evolve? A case study for life science ontologies. In: Joint Workshop on Knowledge Evolution and Ontology Dynamics (ISWC), pp. 1–12 (2012)

Automatic Argumentation Extraction

Alan Sergeant

SAP UK Ltd, The Concourse, Queen's Road, Belfast, BT3 9DT

Abstract. This extended abstract outlines the area of automatic argumentation extraction. The state of the art is discussed, and how it has influenced the proposed direction of this work. This research aims to provide decision support by automatically extracting argumentation from natural language, enabling a decision maker to follow more closely the reasoning process, to examine premises and counter-arguments, and to reach better informed decisions.

Keywords: Argumentation, Argument Extraction, Information Extraction.

1 Problem Overview

Automatic Argumentation Extraction (AAE) is a relatively new research area[1], and work carried out to date is still regarded as experimental[2]. Argumentation can be defined as the process by which arguments are constructed and handled[3], with four main tasks undertaken: identification, analysis, evaluation and invention[4]. Identification is the task of determining the conclusion, premises and scheme of an argument from natural discourse, and is the task with which this work is concerned with automating.

Motivation for this work comes from the question "Was the right decision made? Was it well founded?" For every decision made, one might be asked to justify, explain or defend how it was arrived at[5]. An antagonist can probe the reasoning process which led to the conclusion by asking for clarification or justification. Therefore, it can be said that arguments are constructed to express the reasoning process taken to reach a conclusion, with a view to persuading hearers that the conclusion is valid and the reasoning behind it well grounded[6].

What is an argument? The building blocks of every argument are propositions: a statement or assertion that expresses a judgement or opinion[7]. An argument consists of two or more propositions[3,8], one proposition functions as the claim (also known as the conclusion), and a set of one or more propositions serve as supports (also known as premises). The relationship between the propositions (premises and conclusions) is important. An argument is not simply a collection of propositions, it has a structure, which plays a key role in determining the presence or absence of an argument[7].

One useful approach to viewing arguments and their structure is that of argument diagramming. Argument diagramming enables the conclusions and related premises to be identified, and the relationships between them expressed in a tree

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 656–660, 2013.

structure[9]. It provides an overview of how well supported or attacked a conclusion or premise is. This overview can be used to inform argument-based decision making. The argument diagram is a representation of the reasoning process, and serves as a basis for reflection on how the conclusion was reached. It also enables an antagonist to target certain areas in the reasoning process for further examination. If a conclusion is well supported, or even if it has been attacked and successfully defended, it provides good ground on which to make a decision, as the reasoning process by which it was reached is demonstrated to be valid.

Automatic argumentation extraction incorporates the understanding of construction, handling, and visual representation of arguments and aims to support decision making. Given the new ways in which we communicate (newspapers, Facebook, Twitter, review sites etc.), often statements or assertions are made without explicit justification for the opinion, belief or conclusion. Without this, how can a reader reasonably decide to agree with a post or meaningfully assess whether a product is suitable for purchase?

This research will begin by finding means of automatically identifying argumentative propositions (premises or conclusions). However at this first stage no attention to type (whether a proposition is premise or conclusion) is considered. It has been shown that by filtering out propositions that have no role to play in an argument, a more accurate classification of type can be achieved[1]. Finally, once the propositions have been classified by their type, work will move towards identifying the relationship (support or attack) between the premises and conclusions, forming an argument diagram.

2 Related Work

While so far there has been little work in the area of AAE[1,2], several related areas have been the focus of research: text zoning[10], RST (rhetorical structure theory)[11], argumentation schemes, and argument diagramming[9]. Research which has been carried out in AAE has largely focused on the legal domain[1,12], with more recent work moving to online reviews[13], and online debates[14].

State of the Art. The state of the art encompasses two main approaches to automatic argumentation identification: statistical classification and rule-based parsing. In both cases, the goal is to identify and extract the parts of an argument (premises and conclusions), as well as their relationships. The work in [1] begins with experimentation on the Araucaria corpus [9], but quickly shifts focus to the annotation of a new corpus consisting of fifty-seven ECHR (European Court of Human Rights) cases. This was due to an interest in the full argumentation structure, i.e the relations between arguments, which Araucaria does not provide.

The results of this work have been outlined in Table 1. However issues with the statistical classification approach taken were also given. "This approach cannot detect the delimiters of each argument or their relations. Therefore, it is known which information forms the argumentation but not how this information is split into the different arguments."[1]. This eventually caused the research to shift towards rule-based parsing. These results are also shown in Table 1.

Table 1. State of the art automatic argumentation identification results (F1 Measure)

	Statistical Classification	**Rule-base Parser**
Premise	68.12%	64.3%
Conclusion	74.07%	67.4%
Structure	N/A	60.0%

3 Contributions

Firstly, this research will begin by providing a much needed argumentation corpus, as well as tools to enable easier production of argumentation corpora. This corpus will be annotated with tags suitable for analysis by many of the available Apache UIMA[1] Tools and components[15]. Annotation is underway and a need for clear definitions has already become evident. As in previous work [12], we will attempt to establish an appropriate definition of the elementary units of argumentation. There is broad consensus that arguments are the elementary units of argumentation, but what are the elements of an argument? Initial efforts in annotating a selection of car reviews, (our Car Review Corpus - CRC) highlight the fact that sentences are not appropriate (as compared to the state of the art[1]) as the fundamental elements of an argument (i.e. a complete, conventionally punctuated sentence often cannot simply be labeled as being a single 'conclusion' or a single 'premise'). Take for example the sentence "Other weight saving measures means it is 80kg lighter overall and Audi claims it is the lightest car in the class." This sentence, taken from an AA Car Review for an Audi A3, contains the conclusion, "Audi claims it is the lightest car in the class." However it also contains a premise: "Other weight saving measures means it is 80kg lighter overall" which adds support to the claim. Therefore we can say that this sentence contains both a conclusion and premise. This is a comparatively simple case - a complex sentence comprising two independent clauses joined by a coordinator - and can be easily annotated. However, devising a means of identifying propositions and their role in more complex syntactic constructions will require greater effort. This research regards the proposition, classified as a conclusion or premise, as the smallest element of an argument.

Our research addresses potential issues in moving from the more structured natural language found in legal cases (state of the art), to the less structured and therefore more computationally complex domain of journalistic argumentation and consumer comment, exemplified in this case by car reviews. Our work will explore which features achieve higher accuracies given various machine learning (ML) algorithms. It will begin by using Support Vector Machines (SVMs), evaluating results against other ML algorithms such as Maximum Entropy(ME). ML has been chosen because of the positive results achieved in [1], where it was shown to obtain higher F1 measures in identifying both premises and conclusions compared to rule-based parsing.

[1] `http://uima.apache.org/`

Statistical classification, whether SVN or ME, typically encounters difficulty when faced with the challenge of identifying individual argument parts and then associating them with over-arching argumentation structures. Our research will tackle this segmentation problem by use of semantic analysis. Whilst this was explored briefly in [1], results were unfavorable. However it was stated that, "A different type of document or a more complex clustering model could achieve better results, however it was decided to leave this research line for future work."

4 Evaluations

The evaluation of our system's effectiveness in identifying basic units of argument will be standardised with the state of the art to enable direct comparison. The state of the art uses well known evaluation metrics to count the number of correctly classified sentences. In our work we will be counting the number of correctly classified propositions instead of sentences. Therefore, in the context of classification tasks (cf. classification between two class labels: C1 and C2) the following four terms are used to compare the given labels with the label the items actually belong to:

- True Positive (T_p) : number of propositions correctly classified as C1
- True Negative (T_n) : number of propositions correctly classified as C2
- False Positive (F_p) : number of propositions incorrectly classified as C1
- False Negative (F_n) : number of propositions incorrectly classified as C2

Precision(P), recall(R) and F1 measure are defined as follows:

$$\mathrm{P} = \frac{T_p}{T_p + F_p} \qquad \mathrm{R} = \frac{T_p}{T_p + F_n} \qquad \mathrm{F1} = 2 * \frac{P * R}{P + R} \tag{1}$$

Accuracy is computed as the number of correctly classified propositions divided by the total number of propositions:

$$\text{Accuracy} = \frac{N_{corr}}{N_{total}} \qquad N_{corr} = T_p + T_n \qquad N_{total} = T_p + T_n + F_p + F_n \tag{2}$$

5 Work Plan

The initial effort has been on the creation of tools to help annotate a new argumentative corpus, the Car Reviews Corpus (CRC). These tools, based upon UIMA [15], can be used to annotate text. The annotations created are then capable of being utilised by any new or existing UIMA components.

The next stage of work is the annotation of propositions within the car review texts by several human annotators, followed by a study of annotator agreement. This will lead to the task of training a classifier to automatically identify propositions in unseen text in the same domain.

Upon successful completion, attention will shift back to fully annotating the CRC with argumentative annotations, describing the argument parts and structure. Once again, an evaluation of annotator agreement will be carried out before proceeding to the task of classifying argumentative propositions from non-argumentative, followed by classification of conclusions and premises.

The final stage of this work will be the investigation into a complex semantic clustering model to enable the automatic construction of an argument tree from an unseen text in a domain similar to that of the training corpus.

Acknowledgements. This work is supported by SAP AG and the Invest NI Collaborative Grant for R&D - RD1208002.

Supervisors - Ian O'Neill & Jun Hong (School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast, BT7 1NN)

References

1. Mochales Palau, R.: Automatic Detection and Classification of Argumentation in a Legal Case. PhD thesis, KU Leuven (2011)
2. Walton, D.: Argument mining by applying argumentation schemes. Studies in Logic 4(1), 38–64 (2011)
3. Besnard, P., Hunter, A.: Elements of argumentation, vol. 47. MIT Press (2008)
4. Walton, D.: Argumentation Theory: A Very Short Introduction. In: Rahwan, I., Simari, G.R. (eds.) Argumentation in Artificial Intelligence, 1st edn., pp. 1–22. Springer, US (2009)
5. Baker, G.P., Huntington, H.B.: The Principles of Argumentation. Ginn & Company (1905)
6. Govier, T.: Critical thinking as argument analysis? Argumentation 3(2), 115–126 (1989)
7. Copi, I., Cohen, C.: Introduction to Logic, 11th edn. Pearson Eductaion (2001)
8. Blair, J.A., Tindale, C.W.: Groundwork in the Theory of Argumentation. Argumentation Library, vol. 21. Springer (2012)
9. Reed, C.: Araucaria: Software for argument analysis, diagramming and representation. International Journal of AI Tools 13(4), 961–980 (2004)
10. Teufel, S.: Argumentative Zoning: Information Extraction from Scientific Text University of Edinburgh. PhD thesis (1999)
11. Mann, W.C., Thompson, S.A.: Rhetorical structure theory: A framework for the analysis of texts. IPRA Papers in Pragmatics 1(1), 79–105 (1987)
12. Mochales Palau, R., Moens, M.F.: Argumentation Mining: The Detection, Classification and Structure of Arguments in Text. In: Proceedings of the 12th International Conference on Artificial Intelligence and Law, pp. 98–107. ACM (2009)
13. Schneider, J., Wyner, A.: Identifying Consumers Arguments in Text. In: SWAIE 2012: Semantic Web and Information Extraction (2012)
14. Cabrio, E., Villata, S.: Natural Language Arguments: A Combined Approach. In: 20th European Conference on Artificial Intelligence, ECAI 2012 (2012)
15. Ferrucci, D., Lally, A.: UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. Natural Language Engineering 10(3-4), 327–348 (2004)

Guided Composition of Tasks with Logical Information Systems - Application to Data Analysis Workflows in Bioinformatics*

Mouhamadou Ba

IRISA/INSA Rennes, 35043 Rennes, France
mouhamadou.ba@irisa.fr

Abstract. In a number of domains, particularly in bioinformatics, there is a need for complex data analysis. For that issue, elementary data analysis operations called tasks are composed as workflows. The composition of tasks is however difficult due to the distributed and heterogeneous resources of bioinformatics. This doctorial work will address the composition of tasks using Logical Information Systems (LIS). LIS let users build complex queries and updates over semantic web data through guided navigation, suggesting relevant pieces and updates at each step. The objective is to use semantics to describe bioinformatic tasks and to adapt the guided approach of Sewelis, a LIS semantic web tool, to the composition of tasks. We aim at providing a tool that supports guided composition of semantic web services in bioinformatics, and that will support biologists in designing workflows for complex data analysis.

1 Motivation and Research Questions

The Workflow Management Coalition (WfMC)[1] defines a workflow as "the automation of business processes, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules." Originally used by industry for business processes, workflows have been increasingly used to lead *in silico* experiments in scientific areas. Web services [1] are being used as components of workflows, they provide access to data sources and to tools to analyse data. The bioinformatic domain is much involved in the use of workflows of web services, for example complex data analysis is performed by composing various elementary data analysis operations (e.g. search for homologous sequences, transcription). In bioinformatics however, the resources are complex and heterogeneous. They are produced and maintained by groups localized around the world. The nature of the bioinformatic domain raises distribution and heterogeneity problems, making it difficult to compose tasks.

* This work started in October 2012 under an ARED funding from Region Bretagne and is supervised by Mireille Ducassé (IRISA/INSA Rennes) and Sébastien Ferré (IRISA/University Rennes 1).

[1] http://www.wfmc.org/

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 661–665, 2013.

The semantic web [2] provides technologies that facilitate the composition of web services as workflows. Ontologies for example, are used to describe bioinformatic resources, including meta data, data types and tasks, which eases resource integration. Discovery helps to access services that will compose a workflow. The description of characteristics of services through technologies like RDF facilitates their discovery. Languages like OWL can be used to allow to constraint and reason on workflow management systems. Those technologies can also help manage tasks, results and data provenance. Semantic web technologies can support automation of some manual tasks (e.g. service selection) during workflow definition.

There are different approaches for workflow definition: the manual approach and the automatic approach. The manual approach requires users to entirely define the workflow. With that approach, the definition and update of workflows require too much training for end users. The automatic approach selects components and defines workflows in an automatic manner. That requires strong and complete specifications, which are themselves difficult to express.

Our goal is to provide an environment for the design of workflows in a semi-automatic approach that combines the advantages of manual and automatic approaches. We will use semantic web technologies to support guided composition of services. The work will be applied to the bioinformatic domain. We nevertheless aim to produce methods and tools that are generic and relevant to other fields.

2 State of the Art

A web service corresponds to a set of operations whose characteristics are generally described through an XML-based standard language. It is accessible through standardized web protocols such as SOAP (Simple Object Access Protocol). Web service technologies can serve as infrastructure for workflow development.

Many tools exist to define web services. Some of them operate at a syntactic level, others up to the semantic level. The XML standards WSDL (Web Service Definition Language) and UDDI (Universal Description Discovery and Integration) are defined for, respectively, the description of services and the publication and access to services. They operate at the syntactic level. New languages are proposed to add semantics to the definition of services, for example OWL-S, WSMO and SAWSDL. They lead to semantic web services. Some annotation models based on SAWSDL, OWL-S and WSMO for service annotation force users to think in terms of service interfaces, rather than of high-level functionality. Missier et al. [3], to increases the effectiveness of annotation models, define *Functional Units* (FU) as the elementary units of information used to describe a service. Ontologies like myGrid ontology [4] are also proposed to support the description of web services and data.

In bioinformatics, implementations of web services (e.g. BioMoby [5]) are proposed by institutes and web service providers. Those services are used by many systems for different needs, such as the definition of workflows [6].

Many languages are also proposed to define workflows. Wang et al. [7] present a survey of such languages. For example, the Scufl language (Semantic Conceptual Unified Flow Language) is provided to define scientific workflows in the context of the myGrid project[2]. In face of the great number of workflow languages, criteria are important to choose a language, for example complexity, semantic, license, stability, executability, generalizability, shareability. The quality and degree of automation in the workflow design process depend on the chosen language.

There are many approaches for semi-automatic composition of web services. Some of them use Semantic Web technologies and Artificial Intelligence techniques to assist users in web service selection and composition. Wang et al. [7] assess some of them using the following criteria: use of ontologies [8], filtering of inappropriate services, suggestion of partial plans, checking of the composition validity, use of a planning strategy, use of a modeling environment [8], control constructs and executable results.

Taverna is a component of the myGrid project. MyGrid aims at developing a middleware to support data intensive *in silico* experiments in biology. Taverna [8] is a tool to compose and enact bioinformatic workflows. Its GUI allows biologists to create, execute and share workflows. However, while being much simpler than raw programming, Taverna and similar systems are still difficult to use for average biologists. In Taverna the creation of workflows is neither interactive nor guided enough, there are no automatic data mediation, and no suggestions are made during the workflow design process [7].

3 Approach and Research Methodology

The LIS team[3] has an expertise in guided approaches for data exploration and authoring. Logical Information Systems (LIS) let users build complex queries and updates over semantic web data through guided navigation, suggesting relevant pieces and updates at each step. That approach combines query search and faceted search and is implemented in Sewelis [9]. For example, Sewelis has been applied to the exploration of films and related people, and to the semantic annotation of comics panels.

Our work will address the design of workflows using the LIS approach. The design of workflows requires the location of the relevant tasks, the LIS approach will facilitate integration and management of tasks as well as the selection of tasks that can be matched together to form a workflow. We aim at extending the guided approach of Sewelis to the composition of tasks in order to make it easier for biologists. A visual environment will help users to design workflows. That environment will integrate Sewelis for the retrieval of tasks. We will take advantage of related work and tasks will be wrapped as semantic web services.

[2] `http://www.mygrid.org.uk/`
[3] `http://www.irisa.fr/LIS/`

The suggestion mechanisms and reasoning engine of Sewelis will be adapted to enable automatic parameter matching [10], selection of services and guided edition of workflows. We will address the following tasks:

Resource Description: This task will be the semantic basis of our work. We envision the reuse of [3,4] and we will adapt it for the Logical Information System we use. This part requires an in-depth study of [3,4] and related work.

Resource Editing and Discovery: The objective of this task is to propose methods for guided search and editing on web services and data. Sewelis is a tool that supports easy and intuitive search on semantic web data. Semantic web services are semantic web data, thus Sewelis approach is applicable to discover them. However, web services are a particular kind of data, and they are diverse and heterogeneous. Their discovery depends not only on the representation of their characteristics and functionalities at the registration and update phases, but also on the techniques and algorithms used to match them at the retrieval phase. We will ensure that selected services for composition offer the required features.

Workflow Language: The orchestration of web services involves a workflow definition language. Many languages are proposed, we want to choose a language that helps to be domain independent. The language should also allow the workflows to be enacted and shared.

Guided Composition: We think that, at the architectural level, it is important to separate discovery and composition of web services. For users however the tasks supported by those components must be associated to improve interaction. The insertion of a new service in a workflow must depend on all services already in the workflow. The choice of a service should allow data links and suggestions of composition plans. Contextual information must allow search results and suggestions to be precise. The guided composition will be based on the guided discovery of Sewelis. The automation of tedious tasks and interaction during the process of defining workflow will be supported by resource description. Resource description will be tailored to support reasoning at a reasonable cost. We will adapt the user interface of Sewelis to the workflow edition maintaining its expressiveness and ease-of-use. The workflow will be expressed in the workflow definition language chosen in the previous task. The workflow edition is more difficult when the language is complex. A suitable level of abstraction for the users and simplifications on the patterns of the language will facilitate composition.

Workflows as Services: We will adopt a recursive view on services. We will consider primitive services and complex services. A primitive service will be a task component of a workflow and a complex service will be defined as a workflow that can be used as a task component of another workflow. That view will facilitate reuse and composition.

4 Evaluation Methodology

GenOuest[4] is a bioinformatic platform that provides a large collection of tools and services for data analysis. The platform offers a suitable environment to evaluate and validate our approach. We will use datasets and real cases of the GenOuest platform for evaluation. We will test our approach with biologist users of GenOuest and make a comparison with approaches of existing systems such as Taverna.

References

1. Alonso, G., Casati, F., Kuno, H., Machiraju, V.: Web Services: Concepts, Architectures and Applications. Springer (2003)
2. Hitzler, P., Krötzsch, M., Rudolph, S.: Foundations of Semantic Web Technologies. CRC, Boca Raton (2009)
3. Missier, P., Wolstencroft, K., Tanoh, F., Li, P., Bechhofer, S., Belhajjame, K., Pettifer, S., Goble, C.A.: Functional units: Abstractions for web service annotations. In: SERVICES, pp. 306–313. IEEE Computer Society (2010)
4. Wolstencroft, K., Alper, P., Hull, D., Wroe, C., Lord, P.W., Stevens, R.D., Goble, C.A.: The myGrid ontology: bioinformatics service discovery. Int. Journal of Bioinformatics Research and Applications 3(3), 303–325 (2007)
5. Wilkinson, M.D., Links, M.: Biomoby: An open source biological web services proposal. Briefings in Bioinformatics 3(4), 331–341 (2002)
6. Romano, P.: Automation of in-silico data analysis processes through workflow management systems. Brief Bioinform. 9(1), 57–68 (2008)
7. Wang, Z., Miller, J.A., Kissinger, J.C., Wang, R., Brewer, D., Aurrecoechea, C.: Ws-biozard: A wizard for composing bioinformatics web services. In: SERVICES I, pp. 437–444. IEEE Computer Society (2008)
8. Oinn, T., Greenwood, M., Addis, M., Ferris, J., Glover, K., Goble, C., Hull, D., Marvin, D., Li, P., Lord, P.: Taverna: Lessons in creating a workflow environment for the life sciences. Concurrency and Computation: Practice and Experience 18(10), 1067–1100 (2006)
9. Ferré, S., Hermann, A.: Semantic search: Reconciling expressive querying and exploratory search. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 177–192. Springer, Heidelberg (2011)
10. Lebreton, N., Blanchet, C., Claro, D.B., Chabalier, J., Burgun, A., Dameron, O.: Verification of parameters semantic compatibility for semi-automatic web service composition: a generic case study. In: Taniar, D., Pardede, E., Nguyen, H.Q., Rahayu, J.W., Khalil, I. (eds.) Int. Conf. on Information Integration and Web Based Applications and Services, pp. 845–848. ACM (2010)

[4] `http://www.genouest.org/`

Storing and Provisioning Linked Data as a Service

Johannes Lorey

Hasso Plattner Institute,
Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam, Germany
johannes.lorey@hpi.uni-potsdam.de

Abstract. Linked Data offers novel opportunities for aggregating information about a wide range of topics and for a multitude of applications. While the technical specifications of Linked Data have been a major research undertaking for the last decade, there is still a lack of real-world data and applications exploiting this data. Partly, this is due to the fact that datasets remain isolated from one another and their integration is a non-trivial task. In this work, we argue for a Data-as-a-Service approach combining both warehousing and query federation to discover and consume Linked Data. We compare our work to state-of-the-art approaches for discovering, integrating, and consuming Linked Data. Moreover, we illustrate a number of challenges when combining warehousing with federation features, and highlight key aspects of our research.

1 Introduction and Motivation

In recent years, the semantic and technical foundations of Linked Data have been widely studied and formalized. Well-known technologies, such as the Resource Description Framework (RDF) model for representing structured linked information or the SPARQL Protocol and RDF Query Language (SPARQL) for retrieving this information, have been established. Whereas in principle Linked Data allows leveraging information from multiple different providers quite easily, real-world applications oftentimes only rely on a single dataset or a handful of RDF data sources. The challenges for utilizing Linked Data include:

- Data Discovery, i.e., finding suitable resources for an information need,
- Data Integration, i.e., merging multiple data sources to allow homogeneous access to the combined information contained in all sources, and
- Data Consumption, i.e., retrieving and processing relevant data items.

For developers, finding suitable Linked Data sources for their application need can be cumbersome. While there exist a number of metadata models for describing the contents of Linked Data sources, such information is provided for few datasets, in different formats, and becomes outdated quickly. Moreover, using public SPARQL endpoints to query actual data is tedious: Typical infrastructures are not designed for large-scale access by multiple users. On the other hand, materializing data locally poses new challenges, such as updating issues.

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 666–670, 2013.

Therefore, we propose a Linked-Data-as-a-Service [10] approach combining aspects of both data warehousing and distributed query processing. We use a scalable infrastructure allowing users to create private SPARQL endpoints comprising datasets relevant for their application and to incorporate a federation of public SPARQL endpoints. We address a number of optimization issues for managing locally maintained data as well as remotely retrieved information, including metadata generation and semantic caching strategies.

2 State of the Art and Open Challenges

Data Discovery. There exist a number of projects to assist interaction with individual or sets of Linked Data sources, such as SIG.MA[1], RKB Explorer[2], or the Information Workbench[3]. Their focus mostly lies on visualizing and analyzing information provided during set-up time, while support for discovering, adding, and updating resources during run-time is limited. Typically, these tools are designed to allow information exploration and analysis in combination with a certain degree of UI customization. Whereas these features allow for straightforward interpretation of the contained information, the tools might be insufficient for application developers to further process and extend the knowledge base.

A more general approach to discover suitable knowledge for an application is by analyzing metadata of the available datasets. Usually, RDF is used to present information about Linked Data sources and their contents themselves. Whereas this common model allows easy access to metainformation, specific details may differ due to the variety of vocabularies available to describe Linked Data sources, either semantically (e.g., using Dublin Core[4]) or structurally (e.g., using VoID[5]). Certain Linked Data catalogues, such as the Data Hub[6], provide even further metadata, while many datasets are published without any such information.

Data Integration. In the Web of Data, there exist different techniques to integrate multiple data sources. Typically, ontologies and metadata are utilized for Linked Data integration [5,8]. Here, it is assumed that information from different sources can be mapped to and aligned with one another using common vocabulary elements. In the case of a distributed setting, queries are usually rewritten based on this ontological information and issued against a federation of suitable SPARQL endpoints. Other approaches for Linked Data integration rely on expressive rules to match entities or concepts from different data sources [9].

While the foundations of these techniques have been long established for relational data management and adapted for the use of Linked Data, they rely on a number of requirements which are not always satisfied in real-world scenarios.

[1] `http://sig.ma`
[2] `http://www.rkbexplorer.com`
[3] `http://iwb.fluidops.com`
[4] `http://dublincore.org/documents/dces/`
[5] `http://www.w3.org/TR/void/`
[6] `http://datahub.io/group/lodcloud/`

Consider the case of ontology-based data integration: Some methods assume a common ontology, whereas in a distributed setting individual data sources may use different ontologies. Thus, the problem of data integration becomes an issue of ontology mapping [2]. Moreover, even within a single cross-domain ontology, such as the one provided for DBpedia[7], there is often no clear-cut semantic differentiation between individual concepts or between their properties, leading to ambiguity and challenges in data integration. Additionally, as with Linked Data itself, ontologies are subject to revision, thus causing further problems when integrating datasets adhering to different releases of the same vocabulary.

Data Consumption. Linked Data is usually consumed using one of two setups, either by accessing a central repository containing one or more RDF datasets, or by querying publicly available SPARQL endpoints. Given an adequate hard- and software infrastructure, the first method enables high-performance access to the data at hand. However, maintaining the warehoused data requires sophisticated approaches for index creation, compression, and updating [2]. Gathering information by querying (a federation of) public endpoints alleviates some of these challenges, but may degrade execution performance. Typically, such optimization issues are addressed by different federated query processing techniques.

As with data integration, most research in retrieving RDF Data by federated query processing is based on related work in distributed relational data management. However, Linked Data exhibits several novel features that enable new possibilities for query execution against a federation of data sources. First and foremost, as entities in the Web of Data are identified by unique, dereferencable, and connected URIs [3], relevant resources can be iteratively determined during query evaluation. Again, ontologies can be used to aid this process by homogenizing the schemas of different sources. However, many challenges in real-world applications settings influence the success of distributed query processing, including latency and bandwidth restrictions [2], or reduced endpoint availability.

3 Proposed Approach and Previous Work

As both data warehousing and distributed query processing offer a number of benefits, we propose a hybrid approach to store and provision Linked Data for application developers. Using a Cloud infrastructure, our framework allows for scalable processing of large-scale RDF dumps. In addition, a mediator-based architecture [7] enables ad-hoc integration of new data sources by continuously materializing SPARQL query results. We plan on maintaining a lightweight metadata catalogue that is iteratively extended and updated with information generated within our architecture and gathered from outside sources. Leveraging this metadata collection, users can deploy customized SPARQL endpoints suitable to their information and scalability needs, which are then populated using publicly available data dumps and results retrieved from SPARQL endpoints.

[7] `http://dbpedia.org/ontology/`

In previous work, we focused on metadata generation [4] and ontology reconciliation [1]. These steps are necessary prerequisites for our hybrid framework where data from different sources may be added ad-hoc to a deployed endpoint and the metadata catalogue itself. Currently, we are analyzing real-world SPARQL query logs to identify typical human and machine agent query sequences. Our goal here is to deduce suitable caching strategies to store frequently accessed data. Moreover, we want to identify resources related to requested data and store this information for subsequent queries. Additionally, we hope that automating this process can assist in determining conceptual gaps between different datasets and ontologies by comparing user behavior.

We have implemented a prototype of our framework using the infrastructure provided by Amazon Web Services[8]. In particular, we use a customized virtual machine template to deploy SPARQL endpoint instances that are consequently populated with openly available RDF dumps and results retrieved while executing SPARQL queries. Monitoring the load received by the endpoint allows to scale the resources available to the system, e.g., by increasing the allocated memory. Moreover, we plan on combining the results derived from our query log analysis with run-time information to establish cost-efficient retainment policies.

4 Methodology and Ongoing Research

We have discovered several issues that hinder a more widespread utilization of Linked Data by application developers. While there exist individual approaches to deal with these challenges, they lack applicability and refinement when considering real-world scenarios. Furthermore, there have been only few attempts to combine these solutions to boost the accessibility of Linked Data for developers. In the remainder of this PhD project, our focus lies on the following components:

Materialized View Selection and Management. Materializing results retrieved from SPARQL endpoints for optimized query execution in subsequent requests can improve the responsiveness and usability of Linked Data applications. Whereas selecting adequate data to retain for future access is essential in this process, establishing proper strategies for updating and discarding this information are also difficult challenges [6]. Our focus here lies on view selection for dynamic query workloads, where a shift in access pattern frequency is reflected in the retention strategy for the corresponding resource.

Query Analysis and Expansion. Instead of simply caching received results potentially useful for future requests, it might prove beneficial to rewrite incoming queries to retrieve more information than actually requested by the user. In turn, subsequent queries may be evaluated more efficiently by using this additional data. Similar techniques have been used in information retrieval to expand keyword queries and determine related resources in explorative search settings. We are currently evaluating the quality of different strategies for prefetching Linked Data, e.g., by using ontological and structural information.

[8] `http://aws.amazon.com/`

Data Integration and Ontology Mapping. A core concept of our proposed framework combining both centralized data storage and distributed query processing is data integration. As discussed earlier, this has been as widely-studied research area. Given the use-case scenario of our work, we focus on continuous data integration by leveraging existing ontology mapping techniques. Here, we hope to contribute new results by exploiting the heterogeneous characteristics of different request patterns and corresponding results.

Scalable Data Processing. Whereas public SPARQL endpoints exhibit serious availability and accessibility problems, for example limited bandwidth or long periods of downtime, most Cloud-based platforms offer certain service quality guarantees in the form of service-level agreements. As developers are potentially already deploying their application stack in such an environment, hosting the data there as well may tremendously improve query execution time while leveraging multi-tenancy to reduce the operational expenditure of such a set-up. Using our Cloud-based framework, we hope to verify and exploit these advantages.

References

1. Abedjan, Z., Lorey, J., Naumann, F.: Reconciling ontologies and the web of data. In: Proceedings of the International Conference on Information and Knowledge Management (CIKM), Maui, HI, USA, pp. 1532–1536 (October 2012)
2. Betz, H., Gropengießer, F., Hose, K., Sattler, K.U.: Learning from the history of distributed query processing - a heretic view on linked data management. In: Proceedings of the International Workshop on Consuming Linked Data, COLD (November 2012)
3. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. International Journal on Semantic Web and Information Systems 5(3), 1–22 (2009)
4. Böhm, C., Lorey, J., Naumann, F.: Creating voiD descriptions for web-scale data. Journal of Web Semantics 9(3), 339–345 (2011)
5. Correndo, G., Salvadores, M., Millard, I., Glaser, H., Shadbolt, N.: SPARQL query rewriting for implementing data integration over linked data. In: Proceedings of the International Conference on Extending Database Technology (EDBT), Lausanne, Switzerland, pp. 4:1–4:11 (March 2010)
6. Goasdoué, F., Karanasos, K., Leblay, J., Manolescu, I.: View selection in semantic web databases. Proceedings of the VLDB Endowment 5(2), 97–108 (2011)
7. Langegger, A., Wöß, W., Blöchl, M.: A semantic web middleware for virtual data integration on the web. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 493–507. Springer, Heidelberg (2008)
8. Makris, K., Gioldasis, N., Bikakis, N., Christodoulakis, S.: Ontology mapping and SPARQL rewriting for querying federated RDF data sources. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM 2010, Part II. LNCS, vol. 6427, pp. 1108–1117. Springer, Heidelberg (2010)
9. Schenk, S., Staab, S.: Networked graphs: a declarative mechanism for SPARQL rules, SPARQL views and RDF data integration on the web. In: Proceedings of the International World Wide Web Conference (WWW), New York, NY, USA, pp. 585–594 (April 2008)
10. Wang, L., von Laszewski, G., Younge, A., He, X., Kunze, M., Tao, J., Fu, C.: Cloud computing: a perspective study. New Generation Computing 28, 137–146 (2010)

Interlinking Cross-Lingual RDF Data Sets

Tatiana Lesnikova

INRIA & LIG, Grenoble, France
{tatiana.lesnikova}@inria.fr
http://exmo.inrialpes.fr/

Abstract. Linked Open Data is an essential part of the Semantic Web. More and more data sets are published in natural languages comprising not only English but other languages as well. It becomes necessary to link the same entities distributed across different RDF data sets. This paper is an initial outline of the research to be conducted on cross-lingual RDF data set interlinking, and it presents several ideas how to approach this problem.

Keywords: Multilingual Mappings, Cross-Lingual Link Discovery, Cross-Lingual RDF Data Set Linkage.

1 Motivation

Semantic Web technologies comprise different languages for expressing data as graphs (RDF), describing its organization through ontologies (OWL) and querying it (SPARQL). The Web of Data uses this technology to publish data on the Web. In particular, Resource Description Framework (RDF)[1] - is a standard model for data representation on the Web proposed by W3C[2]. It is designed to represent meta-data about Web resources in the form of triples (Subject, Predicate, Object) and is intended to be processed by machines rather than humans.

The publication of data sets along the Linked Data principles is gaining an increasing importance. The Linked Data Cloud[3] contains data sets from several domains: geographic, media, government, etc. According to the statistics[4], the total number of triples over all 295 data sets reaches 31,634,213,770. Moreover, the Data Hub[5] contains 5034 data sets most of which are publicly available for use. Given this increasing number of the available data sources, one of the key challenges of Linked Data is to be able to discover links across data sets [1]. Interlinking RDF data sets is the process of setting links between related entities. Moreover, given the growth of linked data, automatic methods are necessary to scale. At present, the number of languages[6] of RDF data sets amounts to 474.

[1] http://www.w3.org/TR/rdfprimer/
[2] http://www.w3.org/2001/sw/wiki/Main_Page
[3] http://richard.cyganiak.de/2007/10/lod/
[4] http://wifo5-03.informatik.uni-mannheim.de/lodcloud/state/
[5] http://datahub.io/
[6] http://stats.lod2.eu/languages

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 671–675, 2013.

Thus, the interlinking problem becomes particularly difficult when entities are described in different natural languages since a simple string comparison of entity labels does not suffice.

Research Question. Our core research problem is to provide automatic reliable methods to link disparate RDF data sets published with labels and literals in various natural languages. Since different URIs can refer to the same real-world object, the focus is on identity links, i.e. a link established between two URIs referring to the same resource. The output of the interlinking process is a set of triples of type: URI owl:sameAs URI.

A reliable method in this context should be understood as a method that links the identical entities across data sets with a high precision and recall. It should also be adaptable to a variety of languages. Though some authors [2] distinguish between multilingual and cross-lingual aspects of matching, we consider them interchangeable.

Research sub-questions to be addressed are as follows:

- Are there monolingual methods adequate for our task? Under what conditions language-dependent methods perform better than language-independent ones?
- What are suitable methods for RDF data set interlinking from Computer Science and Natural Language Processing (NLP) perspectives?
- What method works best for cross-lingual RDF data linking? So far, we plan to identify methods working in broad domains.
- Is there a dependency between families of languages to be linked (Indo-European, Sino-Tibetan, Afro-Asiatic) and the quality of the generated links? This dependency could be traced by changing language pairs of data sets to be interlinked.

The potential contribution of this research is to provide or combine methods to facilitate discovering knowledge across data sets where the same entity is described in different natural languages. Some other cross-lingual applications may benefit from the obtained results: Cross-Language Information Retrieval via Semantic Search engines, Document Classification/Clustering, Question Answering, Machine Translation, to name a few. The cross-lingual mappings obtained as a result of the interlinking process will be shared on the Web for further exploitation by multilingual information access tools in order to facilitate access to knowledge across languages.

In the next section we outline several angles from which the entity linking problem can be looked at.

2 State-of-the-Art

Our research will draw upon the knowledge from different domains: Computer Science, Artificial Intelligence, Natural Language Processing, and Data Mining.

The problem of finding correspondences between entities representing the same world object in distinct data sets has been widely studied in the 1960s

in the context of databases. It is known as instance identification, record linkage or record matching problem. In [3], the authors use the term "duplicate record detection" and provide a thorough survey on the matching techniques. Though the work done in record linkage is similar to our research, it does not contain cross-lingual aspect and RDF semantics.

Our research topic belongs to the area of data linking. String similarity measures [4] and linguistic resources [5] are used to computed the distance between the entities. Another type of approach is to use the features of Linked Data [6].

In the NLP area, the problems of entity resolution, multilingual entity recognition and cross-document co-reference resolution [7] gain a close attention due to their complexity and importance for Information Retrieval, QA, etc. The task is to find out whether the occurrences of a name in different plain natural language texts are the same. There is no general solution to this problem, and the decision whether or not two names refer to the same entity usually relies on contextual clues. One of the differences with the task of finding correspondences between RDF data sets is the limited amount of textual data presented in such data sets which makes it more difficult to calculate similarity measure. Moreover, the RDF graph model and RDF semantics can be of use while elaborating linking strategies.

Recent developments have been made in the field of multilingual ontology matching [8,10]. Some work has also been done in creating a multilingual ontology known as BabelNet [9]. This resource can be used for word sense disambiguation and is available in RDF format.

To the best our knowledge, the area of multilingual RDF data sets interlinking which could combine both NLP techniques and information from Linked Data has not seen many studies. The current research will attempt to fill this gap.

3 Proposed Approaches

To achieve our goal, we may not invent a new approach but rather combine existing methods and adapt them for RDF data sets in a multilingual context. Below we highlight several commonly known methods to deal with natural language data which may contribute to this goal.

Semi-automatic or automatic linkage heuristics can be appropriate for generating RDF links between heterogeneous data sources. Machine learning techniques can be used to learn how to match entities. The major drawback of supervised learning would be its dependency on availability of training examples (cross-lingual entity links labeled as matching or not), whereas the difficulty for unsupervised learning would be to define a matching threshold.

Given the multilingual nature of the research topic, some applications from NLP are likely to be exploited. For example, Machine Translation can be used to translate one data set into the language of the other set thus attempting to facilitate computation of similarity metrics. Though it is not always true since the results of Machine Translation systems can be far from perfect and introduce errors decreasing the overall precision. Besides, more than one translation of a particular fact can exist.

Sometimes, a significant part of important information in a text is associated with named entities, for instance, people names, place names, company names. Those might be valuable discriminators when it is necessary to determine whether two documents are about the same entity. Such open-source free text analysis toolkits as GATE[7] and OpenNLP[8] can be used for Named Entity Recognition and Information Extraction tasks.

4 Planned Research Methodology

The main aim of this work is to find reliable and scalable methods and develop tools for linking different URIs used to identify the same resource represented in multiple natural languages and located in different RDF data sets.
To achieve this aim, the research will go through the following steps:

- Synthesize the work done in the research field
- Select the acceptable RDF data sets
- Deal with a problem of partially built data sets
- Explore the semi-automatic and automatic techniques for RDF interlinking
- Decide what methods to choose and how to combine them
- Run experiments on actual data sets
- Evaluate and analyze the obtained empirical results

The research procedure can be summed up as follows:

- Internet-based data collection method will be used to obtain RDF data sets.
- Once the research methods are refined, the experiments will be conducted in order to obtain RDF links between corresponding entities.
- Since we will attempt to automate the linking process as much as possible, standard statistical measures will serve for evaluation. As a starting point, the results of the best multilingual ontology matcher [10] with F-measure = 18% could be considered as a baseline. As of today, there is no official benchmark for doing evaluation. This poses difficulties to objective evaluation of method effectiveness. The problem could be addressed in several ways. One way would be to create reference links manually. The other way would be to exploit existing links between knowledge bases (for example, multilingual DBpedia): first, the existing links are deleted, then the methods are applied and the obtained links are compared against the initially deleted links. Another possible direction is to elaborate a task-oriented evaluation with a well-defined application for evaluating the correctness of the obtained links.

5 Schedule

Time allowance to complete the proposed research is 36 months. Below we present a rough schedule with important milestones for every 6 months.

[7] `http://gate.ac.uk/`
[8] `http://opennlp.apache.org/`

M0-M6: Attend courses and research seminars; bibliographic study
M0-M12: Finalize research methodology, collect corpora and configure software for experiments
M6-M18: Propose problem solutions and conduct preliminary experiments
M18-M24: Analyze results and prepare publication
M24-M30: Generalize results and conduct further experiments
M26-M32: Write up and prepare publications
M32-M36: Send for review and correct final version of thesis

References

1. Ferrara, A., Nikolov, A., Scharffe, F.: Data Linking for the Semantic Web. Int. J. Semantic Web Inf. Syst. 7(3), 46–76 (2011)
2. Spohr, D., Hollink, L., Cimiano, P.: A machine learning approach to multilingual and cross-lingual ontology matching. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 665–680. Springer, Heidelberg (2011)
3. Elmagarmid, A., Ipeirotis, P., Verykios, V.: Duplicate record detection: A survey. IEEE Transactions on Knowledge and Data Engineering 19(1), 1–16 (2007)
4. Winkler, W.: Overview of record linkage and current research directions. Tech. Rep. No. 2006-2. Statistical Research Division. U.S. Census Bureau
5. Scharffe, F., Liu, Y., Zhou, C.: RDF-AI: an architecture for RDF datasets matching, fusion and interlink. In: Workshop on Identity and Reference in Knowledge Representation, IJCAI, Pasadena, CA, USA (2009)
6. Hu, W., Chen, J., Qu, Y.: A self-training approach for re-solving object coreference on the semantic web. In: Proc. 20th International World Wide Web Conference (WWW 2011), Hyderabad, India, pp. 87–96 (2011)
7. Bagga, A., Baldwin, B.: Entity-Based Cross-Document Coreferencing Using the Vector Space Model. In: Proc. 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics, pp. 79–85 (1998)
8. Meilicke, C., Castro, R.G., Freitas, F., van Hage, W.R., Montiel-Ponsoda, E., de Azevedo, R.R., Stuckenschmidt, H., Sváb-Zamazal, O., Svátek, V., Tamilin, A., Trojahn, C., Wang, S.: MultiFarm: A Benchmark for Multilingual Ontology Matching. Journal of Web Semantics 15, 62–68 (2012)
9. Navigli, R., Ponzetto, S.: BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network. Artificial Intelligence 193, 217–250 (2012)
10. Meilicke, C., Trojahn, C., Sváb-Zamazal, O., Ritze, D.: Multilingual Ontology Matching Evaluation - a First Report on Using MultiFarm. In: Proc. 2nd International Workshop on Evaluation of Semantic Technologies, Heraklion, Greece, pp. 1–12 (2012)

Trusting Semi-structured Web Data

Davide Ceolin*

VU University, Amsterdam, The Netherlands

Abstract. The growth of the Web brings an uncountable amount of useful information to everybody who can access it. These data are often crowdsourced or provided by heterogenous or unknown sources, therefore they might be maliciously manipulated or unreliable. Moreover, because of their amount it is often impossible to extensively check them, and this gives rise to massive and ever growing trust issues. The research presented in this paper aims at investigating the use of data sources and reasoning techniques to address trust issues about Web data. In particular, these investigations include the use of trusted Web sources, of uncertainty reasoning, of semantic similarity measures and of provenance information as possible bases for trust estimation. The intended result of this thesis is a series of analyses and tools that allow to better understand and address the problem of trusting semi-structured Web data.

1 Research Questions

Trust is a crucial issue in the Web. The growth of the Web brings the impossibility to control and check every single piece of information we have to deal with. Moreover, the heterogeneity of data sources therein present makes the quality and the reliability of the data that these sources expose vary. Consequently, proper techniques need to be developed and proper analyses need to be performed to provide tools and indications to quantify the reliability of the data observed, so that users can properly handle them. This is the focus of the research described here, as summarized by the following overall problem statement.

> *How can the trustworthiness of semi-structured Web data be adequately estimated?*

I investigate about different aspects inherent to this problem: data, metadata and reasoning techniques useful to make adequate trust estimates.

Research Question 1. The first problem that I focus on is the usage of trusted semi-structured Web data to make trust evaluations of semi-structured data (not necessarily coming from Web sources). This gives a first insight into the possibility to use Web data for assessing the trustworthiness of data. Hence the first research question is:

> *Can Web data help the trust evaluation of semi-structured data?*

* *Supervised by* Guus Schreiber, Wan Fokkink, and Willem Robert van Hage.

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 676–681, 2013.

Research Question 2. Web data present peculiar characteristics that have to be taken into account when using them to make trust evaluations. For instance, they are often accessed incrementally (e.g. by crawling; so we do not always know how representative the data that we observe are), and also their reliability varies, and their source reputation is not always known. Proper reasoning techniques have to be employed to cope with this, and they will be investigated by addressing the following research question:

How can uncertainty reasoning be effectively used to estimate the trustworthiness of semi-structured data?

Research Question 3. Also the Web as such can be exploited for the computation of meta-information that facilitates the estimation of trust values. Web-based semantic similarity measures can be used to weigh data and metadata at disposal of the uncertainty reasoning techniques adopted to estimate the trustworthiness of a given subject, hence the following research question:

Can semantic similarity measures improve the accuracy of trust estimates of semi-structured data based on uncertainty reasoning?

Research Question 4. The Web offers also a meta-level of related information that is useful when dealing with trust, namely provenance information, that represents by whom and how data have been produced, manipulated and exposed. Reasoning over these data is important because this can provide indirect evidence about the reliability of a target object. Moreover, in general, this kind of data possibly enlarges our availability of reliable sources of evidence. This subject will be explored by addressing the following research question:

How can provenance information be used for making accurate trustworthiness estimations of semi-structured data?

2 State of the Art

Trust is a widely explored topic in computer science, in the Web and Semantic Web. Sabater and Sierra [14], Golbeck [10] and Artz and Gil [1] present three comprehensive surveys of the fields. In particular the definition of trust that I make use of is the one of Castelfranchi and Falcone reported by Sabater and Sierra, that is "the decision that an agent x (trustor) takes to delegate a task to agent y (trustee) is based on a specific set of beliefs and goals, and this mental state is what we call trust". Depending on the scenario where my case study locate, the trustors will vary and the goal of my research will be to build tools or models able to mimic their behavior given the constraints of the case. I do so by employing uncertainty reasoning, provenance analysis and semantic similarity measures. The link between provenance and trust, mentioned in the survey of Artz and Gil, has been explored by Golbeck [9] but, mainly for addressing socio-related issues, while my my focus is on the data trustworthiness estimation. Uncertainty reasoning techniques are often used to make trust assessments, like in the work of Fokoue et al. [8]. It is important to investigate further the

possibility to represent these data by means of multiple layers of probabilities, because of their adequateness to deal with vast amounts of heterogenous data.

The link between trust and semantic similarity measures has already been explored, for instance by Ibrahim et al. [11] and by Sensoy et al. [15]. This link can be further explored by considering the relation between different kinds of semantic similarity measures (e.g. deterministic or probabilistic ones) and evidential reasoning. Also, the trust evaluations obtained by means of semantic similarity measures may be effectively integrated with those based on provenance.

3 Proposed Approach

I propose the following approaches to tackle each research question.

Research Question 1. I propose a quantitative empirical approach for this research question, by using uncertainty reasoning to make sense of Web data to trust unknown data. This has merely explorative goals (proving the possibility to use Web data to make trust assessments), and its novelty resides in the use of evidential reasoning in combination with Web data for making trust assessments.

Research Question 2. The approach proposed for this question is quantitative and empirical, and aims at producing a description of how categorical Web data fit higher-order probability distributions. This approach is novel as it provides a first description of Web data in terms of higher-order probabilities.

Research Question 3. I employ a quantitative approach to determine whether I can improve the accuracy of trust values by into account semantic similarity measures. I adopt a theoretical approach to incorporate semantic similarity measures in uncertainty reasoning techniques, which is yet another novel result.

Research Question 4. This research question is tackled empirically. By obtaining an analysis of the use of provenance for trust estimation using statistical techniques, I obtain a novel application.

4 Methodology

Here I introduce the methodologies chosen to implement the above approaches.

Research Question 1. The Naturalis Museum in The Netherlands holds a collection of annotated bird specimen, which includes information like the species these specimens belong to, and the authors of the annotations. These annotations are not fully trustworthy, either because of their inaccuracy or because of the obsolescence of the taxonomy. I map these annotations to trusted Semantic Web sources to check them and, based on a gold standard, I estimate their trustworthiness using a probabilistic logic, named subjective logic [12], that allows to cope with uncertainty about the representativity of the sample observed. I use these trust values with range of decision strategies to decide whether to trust the annotations and I measure the accuracy of the algorithm.

Research Question 2. I investigate further about the statistical foundations of subjective logic, and I use second-order probability distributions and stochastic processes to model the data contained in the Linked Open Piracy dataset [17],

which contains a partial collection of piracy attacks descriptions. I focus on categorical data, which are among the most popular kind of data on the Web (URI). I model the data by means of Dirichlet-multinomial distributions and Dirichlet Processes, high-order probabilistic models for categorical data and I compare their ability to cope with the lack of a full view on the data with multinomial probability distributions based on the evidence at my disposal.

Research Question 3. Semantic similarity measures (e.g. the Wu & Palmer similarity [19]) are used to improve the precision of the uncertainty reasoning techniques adopted for trust estimation. I incorporate semantic similarity measures in the uncertainty reasoning techniques, in particular in subjective logic, proving theoretically whether they can be used as a "discounting" factor for probabilities in subjective logic. I compare the precision and the accuracy of trust values of tags of the Steve Museum [16] dataset (which annotate cultural heritage artworks) when semantic similarity weighing is used and when it is not.

Research Question 4. First, I build a bayesian network using subjective logic on top of provenance graphs, to derive a trust value for a data artifact from the analysis of how it has been produced. This is validated over a set of messages (AIS) sent by ships to coast guard authorities to communicate mandatory information (e.g. their nationality). The validation focuses on the feasibility of the approach, by proving the possibility to build an algorithm that provides such a network. Second, I use machine learning methods to make trust predictions based on the provenance graph of the target objects. In particular, I predict the trustworthiness of a collection of video tags provided by the gaming platform *Waisda?* [13]. Accuracy, precision and recall of the predictions are computed.

5 Results

Here I report the results obtained by addressing the research questions above.

Research Question 1. An algorithm based on subjective logic that uses Web data to assess trust values about the dataset of 65,600 bird specimen annotations of the Naturalis Museum (30% of which serve as training set) [6].

Research Question 2. An analysis of the effectiveness of second-order probability distributions in representing Web data, tested over 2,309 LOP piracy attacks [7]. A first extension of subjective logic to handle higher-order probabilities, which I demonstrate theoretically [5].

Research Question 3. An extension of subjective logic to incorporate semantic similarity measures as a means to weigh evidence within the logic, which I prove theoretically [5], and a first algorithm that employs this extension for computing trust estimates over samples from the 45,860 tags from the Steve.Museum dataset [4], which has been validated by means of a statistical hypothesis test.

Research Question 4. An algorithm that builds a subjective logic-based bayesian network over a provenance graph, compliant to AIS messages [2]; an algorithm that estimates trust based on provenance graphs of 37,850 *Waisda?* tags (training set 70%, test set 30%), by using machine learning classifiers [3].

6 Remaining Work

In this section I describe the remaining work and I indicate a time plan.

Research Question 2-3. Additional extensions of subjective logic incorporating semantic similarity measures and higher-order probabilities; 2 months.

An algorithm for trust computation using uncertainty reasoning combined with semantic similarity measures and provenance metadata; 2 months.

Research Question 4. An algorithm for trust computation based on the semantics of the PROV-O ontology [18]; 3 months. **Thesis writing** 4 months.

References

1. Artz, D., Gil, Y.: A survey of trust in computer science and the semantic web. Journal of Semantic Web 5(2), 131–197 (2007)
2. Ceolin, D., Groth, P., van Hage, W.R.: Calculating the trust of event descriptions using provenance. In: SWPM, vol. 670, pp. 11–16. ceur-ws.org (2010)
3. Ceolin, D., Groth, P., van Hage, W.R., Nottamkandath, A., Fokkink, W.: Trust evaluation through user reputation and provenance analysis. In: URSW, vol. 900, pp. 15–26. ceur-ws.org (2012)
4. Ceolin, D., Nottamkandath, A., Fokkink, W.: Automated evaluation of annotators for museum collections using subjective logic. In: Dimitrakos, T., Moona, R., Patel, D., McKnight, D.H. (eds.) IFIPTM 2012. IFIP AICT, vol. 374, pp. 232–239. Springer, Heidelberg (2012)
5. Ceolin, D., Nottamkandath, A., Fokkink, W.: Subjective logic extensions for the semantic web. In: URSW, vol. 900, pp. 27–38. ceur-ws.org (2012)
6. Ceolin, D., van Hage, W.R., Fokkink, W.: A Trust Model to Estimate the Quality of Annotations Using the Web. In: WebSci. Web Science Repository (2010)
7. Ceolin, D., van Hage, W.R., Fokkink, W., Schreiber, G.: Estimating uncertainty of categorical web data. In: URSW, vol. 778, pp. 15–26. ceur-ws.org (2011)
8. Fokoue, A., Srivatsa, M., Young, R.: Assessing trust in uncertain information. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010, Part I. LNCS, vol. 6496, pp. 209–224. Springer, Heidelberg (2010)
9. Golbeck, J.: Combining provenance with trust in social networks for semantic web content filtering. In: Moreau, L., Foster, I. (eds.) IPAW 2006. LNCS, vol. 4145, pp. 101–108. Springer, Heidelberg (2006)
10. Golbeck, J.: Trust on the World Wide Web: A Survey. Foundations and Trends in Web Science 1(2), 131–197 (2006)
11. Ibrahim, H., Atrey, P.K., El Saddik, A.: Semantic similarity based trust computation in websites. In: MS, pp. 65–72. ACM (2007)
12. Jøsang, A.: A Logic for Uncertain Probabilities. Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 9(3), 279–311 (2001)
13. Netherlands Inst. for Sound and Vision. Waisda? (August 2012), `http://wasida.nl`
14. Sabater, J., Sierra, C.: Review on computational trust and reputation models. Artificial Intelligence Review 24, 33–60 (2005)
15. Sensoy, M., Pan, J.Z., Fokoue, A., Srivatsa, M., Meneguzzi, F.: Using subjective logic to handle uncertainty and conflicts. In: TrustCom, pp. 1323–1326. IEEE Computer Society (2012)

16. Steve: The Museum Social Tagging Project (January 2013), http://www.steve.museum
17. van Hage, W.R., Malaisé, V., van Erp, M.: Linked Open Piracy (November 2012), http://semanticweb.cs.vu.nl/lop/
18. W3C. PROV-O (June 2012), http://www.w3.org/TR/prov-o//
19. Wu, Z., Palmer, M.: Verbs semantics and lexical selection. In: ACL, pp. 133–138. ACL (1994)

Augmented Reality Supported by Semantic Web Technologies

Tamás Matuszka

Eötvös Loránd University, Budapest, Hungary
tomintt@inf.elte.hu

Abstract. Augmented Reality applications are more and more widely used nowadays. With help of it the real physical environment could be extended by computer generated virtual elements. These virtual elements can be for example important context-aware information. With Semantic Web it is possible among others to handle data which come from heterogeneous sources. As a result we have the opportunity to combine Semantic Web and Augmented Reality utilizing the benefits of combination of these technologies. The obtained system may be suitable for daily use with wide range of applications in field of tourism, entertainment, navigation, ambient assisted living, etc. The purpose of my research is to develop a prototype of general framework which satisfies the above criteria.

Keywords: Semantic Web, Augmented Reality, Ontology, Mobile Application.

1 Motivation and Research Questions

Before my doctoral research I dealt in detail with Augmented Reality and Semantic Web, but I worked in two areas independently. Thanks to my acquired experiences I recognized the opportunities which follow from the combination of these two different technologies. It is these facilities what I try to exploit during my research (e.g. to link physical places, objects and people to digital content). Both areas are dynamically growing fields thus there are several papers in field of Semantic Web and Augmented Reality. The experiments of combining these two technologies mostly come from the last few years therefore the common literature is not sophisticated. According to the statement of Gartner's 2012 Hype Cycles Special Report [1] both technologies are in the peak and have at least five years to the mainstream adaptation. It is also a motivating factor to research in this field.

Augmented Reality is such technology which is able to combine the physical environment with virtual elements generated by computer in real time. These elements could be, for example, 3D models, videos, images, music, animations, information, etc. The system created this way is located between the real and the virtual world. There are two kinds of Augmented Reality: one called marker based and another called location-based. The first one uses so-called marker, which usually is an arbitrary photo. With help of the marker the system is capable to compute the position

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 682–686, 2013.

and orientation of the virtual elements in the physical environment. The virtual element will appear on the marker. A good example for this is to display the 3D model of a molecule structure in chemistry textbook because it could be hard to imagine using only the two-dimensional representation. The second one is related with the physical position of the user. Computing of this position is usually based on GPS coordinates. The system shows the virtual elements depending on the location. For example, one could view the restaurants located in a given range, represented by virtual icons on the display of mobile device.

Today there are several location based Augmented Reality applications (e.g. Wikitude[1], Layar[2]) but they extract the needed data typically from one given data source. Nowadays with using of Semantic Web we can access a lot of public datasets, see the LOD cloud [2]. With the help of the LOD cloud the visualization capability of Augmented Reality could be extended. It would be useful to extract the displayable information from the public datasets located on internet instead of one given data source.

Currently accessible applications are typically made for a specific area, about this we could read in Section 2. I did not find any framework which was sufficiently general, or which was appropriate to develop arbitrary Augmented Reality application. To reach the general purposes we have to separate the data model from specific application area, because this is not a typical property of the accessible applications. For the above reasons I feel necessary to develop a general Augmented Reality framework, which is able to separate the data model and the logic of the application using Semantic Web and to provide the showable information from continuously expanding public datasets. To achieve this aim I have to examine semantic mobile applications and semantic database management system solutions, work out architectures and approaches, develop a prototype and test the efficiency of this system.

Based on the aforementioned facts the following questions are arising: How can static databases used by AR applications be connected and extended with semantic datasets? What kind of architecture and information model is necessary for the effective implementation and for ensuring the generality? What are the needed functions of the system? In which application fields could the system be used?

2 State of the Art

The literature has several application areas which use both Semantic Web and Augmented Reality (e.g. navigation, ambient assisted living, manufacturing, etc.). This section shows some of them.

In a previous paper [3] we describe an indoor navigation system which uses Augmented Reality to visualization. Storing of the map's data was based on an ontology and to generate the possible paths we ran rule-based inferences.

---

[1] http://www.wikitude.com/

[2] http://www.layar.com/

Hervás, García-Lillo and Bravo present in [4] a mobile application for supporting daily life of elderly-people. They propose an adaptive model to transform physical information into virtual representation. To do this they use the accelerometers and digital compass of device. Users and their environment are represented in a formal context model. Based on this model and using semantic axioms and inference rules they can determine what the users want to do.

Schmalstieg, Langlotz and Billinghurst [5] intended to combine Web 2.0 and Augmented Reality. For this they implemented a location-based mobile Augmented Reality application, which enhances creativity, collaboration, communication and the reliable information sharing. Based on their system, they developed an indoor navigation system called Signpost [6] which is used for location-based conference guide.

According to Schmalstieg and Reitmayr [7] the data model has to be independent from specific application and their implicit assumptions. The georeferenced Semantic Web provides such a data model. In the paper they investigate how this model fits the requirements of Augmented Reality applications and how such a system can be developed.

Nixon et al. [8] suggest a possible solution for the cooperation of Semantic Web and Augmented Reality. They present how things of internet are described semantically and how can link into the LOD cloud. They implemented an application which is capable of manual annotation of concert posters (i.e. posters in the street advertise concerts and clubs.). The application can recognize these posters and then displays the extracted information.

There exist some touristic applications in [9] and [10], applications which support manufacturing processes in [11] and which use of robotics in [12].

We can see there are many solutions in different fields but it is conspicuous, that there is not any tool which is capable to make arbitrary Augmented Reality applications. The investigated programs are typically only marker based or only location-based. The system what I will to develop has to answer several open questions. One of the important factors is to specify the information model. To efficient operation of framework I should design the needed architecture. It is a problem, that there is not standardized evaluation method for location based applications

3 Proposed Approach

The aim of my research is to create a general Augmented Reality framework which exploits the advantages of Semantic Web. The framework could be divided into two parts. Arbitrary marker based application can be created with the first part. The idea is similar to the approach in [8], but the application area of that solution is limited to using concert posters. In my case it would be possible to create any marker based Augmented Reality applications with my framework. The second part of the system will be a location-based Augmented Reality application which combines the advantages of existing solutions and complements their incompleteness.

To create such kind of model is mandatory to separate the data model from specific application areas. In order to achieve this goal it is needed to design the information

model. This model requires various new ontologies and linking existing ontologies to each other. These ontologies (which are made in OWL language) describe the conceptual hierarchy of the members which are located in different levels in the system.

One of my main objectives is the richer description of the existing georeferenced POIs (Point of interest) based on the LOD cloud. Information could also be taken from the LOD cloud in the case when some POIs do not have enough description (e.g. a POI has latitude and longitude but has not altitude). With help of Semantic Web and the LOD cloud I will dynamically link context-aware physical objects to virtual information, content and services. For this purpose different SPARQL queries and RDF datasets are needed. Let us consider an example. Suppose that we get the information about a building based on existing POIs. Afterward we can complete the given information from the LOD cloud (e.g. who was the architect of the building and what did he designed nearby.)

Social networks and Web 2.0 solutions are very popular. Therefore I feel important that my framework could share content in the various popular social networks (e.g. Facebook, Twitter). Using the location-based module of the framework the users would be able to share their current activities, to rate the viewed places, etc. To reach this aim, it is necessary to develop a user system. Handling of the profiles would happen with ontology. This ontology also provides the personalization.

After my framework is done, users could create applications which are capable to use arbitrary markers navigate through any area, finding different places, create and share content with the location-based module. The system could even serve as a base for smart city applications. My system will be built using client-server architecture or maybe on cloud architecture and it will be able to use the available services provided by Internet. The clients could be various devices, etc. smartphones, tablets, even Google Glass too.

4 Planned Research Methodology

In the first steps of the research I plan to explore and analyze the related work. Also an important objective is studying the basic ontology methods in this phase. With the possession of obtained knowledge the next steps are searching semantic mobile applications, examining semantic database management system solutions and finding new areas where the Semantic Augmented Reality is applicable.

After the preliminary study and determining the application the next step is to design the detailed specification of the prototype.

The implementation of the prototype follows the specification. This part can be divided into multiple parts because of modularization. The first part is specifying the information model which is the base of the system. For this purpose various ontologies are needed to describe the elements and their relations and the rules of the system. When the information model is done, I will implement the marker based and the location based modules. For this it is necessary to observe the existing open source solutions. If there are not such solutions, I have to develop it. Integration of the developed modules is also needed.

It is a problem, that there is not standardized evaluation so I will overview the frequently applied evaluation methods. On one hand I will test my framework with heterogeneous group and surveying, and on the other hand I will compare my system with the existing similar applications.

5 Schedule

This is the first year of my doctoral studies. I will specify the information model and extend the existing mathematical model by the end of the first year. I plan the beginning of implementation of the prototype and the server and the location-based module in the second year. Furthermore I hope that I also finish the marker based module in that year. At the end of my studies I will finish and evaluate my prototype of the framework, will make the connection to the social networks and write the dissertation.

References

1. Gartner Hype Cycles, http://www.gartner.com/technology/research/hype-cycles/
2. Bizer, C., Jentzsch, A., Cyganiak, R.: State of the LOD Cloud, http://wifo5-03.informatik.uni-mannheim.de/lodcloud/state/
3. Matuszka, T., Gombos, G., Kiss, A.: A New Approach for Indoor Navigation Using Semantic Webtechnologies and Augmented Reality. In: 15th International Conference on Human-Computer Interaction, Las Vegas (accepted, 2013)
4. Hervás, R., Garcia-Lillo, A., Bravo, J.: Mobile Augmented Reality Based on the Semantic Web Applied to Ambient Assisted Living. In: Bravo, J., Hervás, R., Villarreal, V. (eds.) IWAAL 2011. LNCS, vol. 6693, pp. 17–24. Springer, Heidelberg (2011)
5. Schmalstieg, D., Langlotz, T., Billinghurst, M.: Augmented Reality 2.0. In: Virtual Realities, pp. 13–37. Springer (2011)
6. Mulloni, A., Wagner, D., Schmalstieg, D., Barakonyi, I.: Indoor Positioning and Navigation with Camera Phones. Pervasive Computing 8(2), 22–31 (2009)
7. Schmalstieg, D., Reitmayr, G.: The World as a User Interface: Augmented Reality for Ubiquitous Computing. In: Location Based Services and TeleCartography, pp. 369–391. Springer (2007)
8. Nixon, L., Grubert, J., Reitmayr, G., Scicluna, J.: SmartReality: Integrating the Web into Augmented Reality. In: I-SEMANTICS 2012 Posters & Demonstrations Track, pp. 48–54. CEUR-WS, Graz (2012)
9. Henrysson, A., Ollila, M.: UMAR - Ubiquitous Mobile Augmented Reality. In: 3rd International Conference on Mobile and Ubiquitous Multimedia, pp. 41–45. ACM, Maryland (2004)
10. Serrano, D., Hervás, R., Bravo, J.: Telemaco: Context-aware System for Tourism Guiding based on Web 3.0 Technology. In: International Workshop on Contextual Computing and Ambient Intelligence in Tourism, Riviera Maya (2011) ISBN: 978-84-694-9677-0
11. Khan, W.A., Raouf, A., Cheng, K.: Augmented Reality for Manufacturing. In: Virtual Manufacturing, pp. 1–56. Springer (2011)
12. Kim, M.-H., Lee, M.-C.: A Path Generation Method for Path Tracking Algorithms that use the Augmented Reality. In: International Conference on Control, Automation and Systems, pp. 1487–1490. IEEE, Gyeonggi-do (2010)

Search Result Ontologies for Digital Libraries

Emanuel Reiterer

School of Information Systems, Curtin University, Perth, Western Australia, Australia
emanuel.reiterer@postgrad.curtin.edu.au

Abstract. This PhD investigates a novel architecture for digital libraries. This architecture should enable search processes to return instances of result core ontologies further on called result ontologies linked to documents found within a digital library. Such result ontologies would describe a search result more comprehensively, concisely and coherently. Other applications can then access these result ontologies via the web. This outcome should be achieved by introducing a modular ontology repository and an automatic ontology learning methodology for documents stored in a digital library. Current limitations in terms of automatic extraction of ontologies should be overcome with the help of seed ontologies, deep natural language processing techniques and weights applied to newly added concepts. The modular ontology repository will be comprised of a top-level ontology layer, a core ontology layer and a document and result ontology layer.

Keywords: ontology, ontology learning, ontology modularisation, digital library, semantic digital library, semantic data management, search result ontology.

1 Motivation and Research Questions

The following motivators led to this research: Firstly, the semantic accessibility of documents within a digital library could contribute to the semantic web. This could be achieved by ontologies created automatically and triggered by a conceptual search. Secondly, an ontology repository within a digital library could enhance the search process within a digital library by enabling ontological search that includes more than a meta-data search and does not necessarily have to incorporate a full-text search. Thirdly, result ontologies, if rendered properly, could provide a concise, coherent, yet comprehensive search result to the user. This result will be concise because it will consist of a conceptualisation about a query and not only a set of documents, coherent because these concepts will be related meaningfully, and comprehensive because the whole result set of documents will be represented through one ontology.

These motivations lead to the following research question: Can a digital library be improved to enable more coherent, concise, yet comprehensive query result presentations by using ontologies?

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 687–691, 2013.

2 State of the Art

This research covers two different research areas: digital libraries and ontologies. In computer science, the term ontology is used to mean "a formal, explicit specification of a shared conceptualisation" [1].

Cimiano [2] describes ontology learning as a reverse engineering process where an ontology reflects the author's point of view. Ontology learning includes several tasks: the extraction of terms, definition and hierarchical organisation of concepts, extraction of relations and attributes as well as the definition of axioms [2]. Deep natural language processing techniques, such as the use of lexico-syntactic patterns, are promising but not thoroughly investigated in current ontology learning processes [3]. This research proposes that patterns, which include the extraction of implicit information such as the train of reasoning, could improve the learning of expressive ontologies. Additionally, a standardised document core ontology could help to create consistent and reusable results.

There are an increasing number of ontology repositories available but current digital libraries could provide a wealth of new ontologies, although these ontologies have to be extracted first, which is part of this work.

A digital library is defined as "a focused collection of digital objects, including text, video, and audio, along with methods for access and retrieval, and for selection, organization, and maintenance" [4]. The goals of semantic digital libraries are to enhance information extraction, to connect information within a digital library, for query refinement, and also for recommendation services. Ontologies are used as bibliographic ontologies and community-aware ontologies [5]. Ontology repositories, built upon an ontology hierarchy, as well as implicit information, such as the extraction of the thesis statements could improve the search processes but also digital libraries in general. Additionally, a result ontology repository could combine information within a digital library, mentioned in a multitude of books or documents, by incorporating or referencing the actual document.

Open problems addressed in this research are (1) the learning of more expressive ontologies [6] by the use of deep natural language processing techniques, (2) the linkage between ontologies and unstructured documents, (3) the provision of standards for ontology repositories by strictly following a top-level ontology and the use of modularised ontologies within an ontology repository, and an (4) automatic ontology creation methodology.

3 Approach

This research proposes and will develop and evaluate five main new artefacts including (1) a generalised digital library architecture, which introduces (2) a modular ontology repository, (3) a search process, (4) an indexing process, and (5) an automatic ontology creation methodology. This work will extend a commonly used document repository system, a full-text search engine, as well as a natural language processing library. It also incorporates state-of-the-art ontology learning algorithms.

Figure 1 shows the initial architecture where a digital library is divided into a document repository, a document ontology repository, and a result ontology repository. The document ontology repository stores document ontologies about each document in the system. The result ontology repository consists of ontologies created on the fly if a search is not already represented by an ontology. This figure also depicts processes for search and indexing. The indexing process creates a document ontology out of an inserted document and updates affected result ontologies. The search process searches for existing result ontologies and combines document ontologies if no result ontologies are found. If no document ontology exists a full-text search will be initiated.

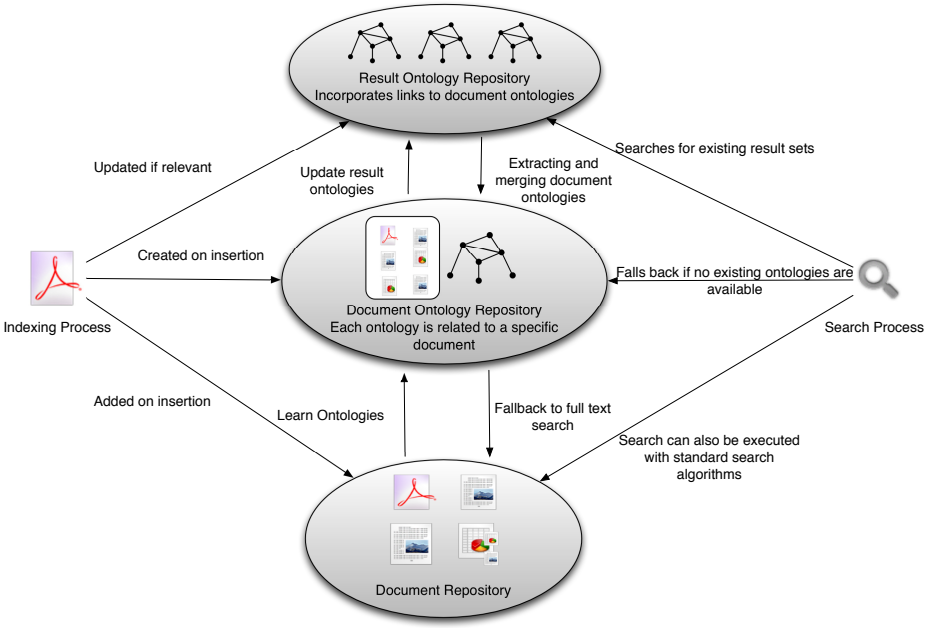


Fig. 1. Digital Library Processes and Repository Architecture

Figure 2 shows the ontology hierarchy used to integrate document ontologies and result ontologies. Although the main outcomes of this research are result ontologies, this hierarchy is essential to provide a consistent basis and is utilised to create such result ontologies. All ontologies are based on a common top-level ontology. The document core ontology describes four aspects: the structural aspect, the technical aspect, the syntactical aspect, and content. The result core ontology contains result based information. The reference ontology is comprised of contextual information. Each document is expressed by a document ontology. A result ontology is an instantiation of the result core ontology and incorporates

subsets of a set of document ontologies. With this hierarchy it will be possible to search for predefined concepts and relations in the document core ontology and the result core ontology but also more generally by using the reference ontology.

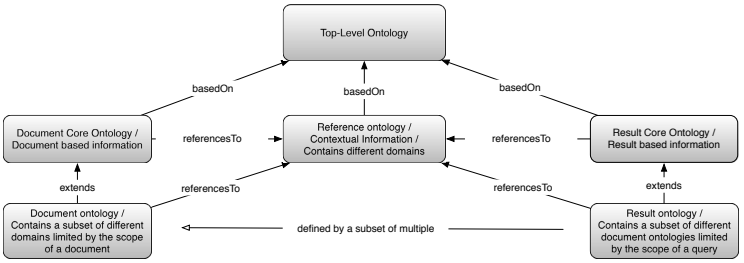


Fig. 2. Ontology Hierarchy

One difficulty of this approach is how to learn ontologies automatically. Well established seed ontologies, which are incorporated in a reference ontology, such as UMBEL (http://umbel.org), should mitigate this problem. Additionally, the use of lexico-syntactic patterns should make it possible to extract more valid and expressive ontologies. Also, weights for newly added concepts will be calculated, as proposed in Boese et al. [7], to minimise the influence of unimportant or false concepts. A limitation of this research is that it will not address the presentation of search result ontologies to the user.

4 Research Methodology

Because this research is about the design and evaluation of the artefacts mentioned in section 3, design science research [8] has been chosen. The artefacts will be evaluated ex-ante and ex-post [9]. Ex-ante evaluations should demonstrate the feasibility of the generalised architecture and algorithms. The implemented artefacts will then be evaluated ex-post. Artificial methods [9] will be used to analyse the artefact in terms of functionality and efficacy and naturalistic methods will be applied by asking ontology experts to evaluate the created ontologies.

5 Current Status and Future Work

This research started with the creation of an initial version of a document core ontology design and a result core ontology design. Afterwards, a proposal for an initial automatic ontology creation methodology has been defined, which meets the needs for use in a digital library. This methodology relies on seed ontologies that are already available and heavily utilised. Such ontologies are either hand selected or well established ones that are selected automatically. To support

automatic selection of such ontologies, it is planned to utilise the ontology usage analysis framework by Ashraf [10].

To evaluate the intended benefits of the result ontologies a result ontology will be created manually and presented to a small group of study participants. These participants will then be interviewed about the completeness of this ontology concerning the searched topic in terms of concepts, relations, and linkage to the actual documents and the improvement of such a result in contrast to a normal result list. The next step includes the automatic creation of document and result ontologies. Deep natural language processing for ontology learning by defining lexico-syntactic patterns will build the basis for learning ontologies. After that, a generalised architecture for digital libraries as well as indexing and search processes will be defined and evaluated ex-ante. Then the artefacts will be implemented and finally evaluated ex-post.

References

1. Studer, R., Benjamins, R., Fensel, D.: Knowledge engineering: Principles and methods. Data & Knowledge Engineering 25(1), 161–197 (1998)
2. Cimiano, P.: Ontology Learning from Text: Algorithms, Evaluation and Applications, p. 20. Springer Science and Business Media (2006)
3. Zouaq, A.: An Overview of Shallow and Deep Natural Language Processing for Ontology Learning. In: Ontology Learning and Knowledge Discovery Using the Web: Challenges and Recent Advances, Hershey, PA, pp. 16–37 (2011)
4. Witten, I.H., Bainbridge, D., Nichols, D.M.: How to Build a Digital Library, 2nd edn., p. xvi. Morgan Kaufmann Publishers, Burlington (2010)
5. Kruk, S.R., Mc Daniel, B.: Semantic Digital Libraries, pp. 5–73. Springer, Heidelberg (2009)
6. Völker, J., Haase, P., Hitzler, P.: Learning Expressive Ontologies. In: Ontology Learning and Population: Bridging the Gap between Text and Knowledge, pp. 45–69. IOS Press (2008)
7. Boese, S., Reiners, T., Wood, L.C.: Concept-based indexing in the design and construction of semantic document networks to support concept retrieval. In: Encyclopedia of Business Analytics and Optimization. IGI Global, Hershey (in press)
8. Hevner, A., March, S., Park, J., Ram, S.: Design science in information systems research. MIS Quarterly 28(1), 75–105 (2004)
9. Venable, J., Pries-Heje, J., Baskerville, R.: A comprehensive framework for evaluation in design science research. In: Peffers, K., Rothenberger, M., Kuechler, B. (eds.) DESRIST 2012. LNCS, vol. 7286, pp. 423–438. Springer, Heidelberg (2012)
10. Ashraf, J., Khadeer Hussain, O., Khadeer Hussain, F.: A Framework for Measuring Ontology Usage on the Web. The Computer Journal (2012) (in press)

Semantically Assisted Workflow Patterns for the Social Web

Ioannis Stavrakantonakis

Semantic Technology Institute (STI) Innsbruck, University of Innsbruck,
Technikerstr. 21a, 6020 Innsbruck, Austria
ioannis.stavrakantonakis@sti2.at

Abstract. The abundance of discussions in the Social Web has altered the way that people consume products and services. This PhD topic aims to materialise a novel approach to assist online communication in the Social Web by combining workflow patterns and behaviour modelling. Semantic Web technologies are considered beneficial in various aspects of this approach, like in the behaviour modelling, personalisation and context-aware workflows.

Keywords: Social Web, Semantic Web, Behaviour modelling, Ontologies, Workflow Patterns.

1 Motivation

The character of online communication has radically changed upon the introduction of Social Web in the daily life of people. This vast expose of users to information and opinions from colleagues, friends or acquaintances in their online social circle has definitely affected enterprises that rely on the traditional word-of-mouth regarding the quality of the offered services to the end-user by introducing both challenges and new opportunities in the Social Web [1]. The speed of message distribution and the number of people that a message reaches comprise the major aspects of word-of-mouth. Both dimensions have been drastically changed in the last few years; sharing an opinion requires only some internet connection (on a mobile or desktop device) and a few seconds to compile a message and share it; and the number of people that a message can reach has exponentially been increased as we can push a message simultaneously to the various ever-expanding social network graphs. Enterprises, in order to address the above-mentioned challenges and turn them into opportunities, should be able to understand the dynamics of communication and the behaviour of users in the Social Web. In this respect, behaviour modelling and workflow patterns could assist enterprises handling the online communication with end-users by applying them in a context-aware manner. The initial idea is to map behaviour patterns with actions in a workflow to assist the offering of services; these workflow patterns are refered as *communication patterns* in the proposed approach.

Therefore, the aim of this PhD research is the *specification of the infrastructure and the communication patterns that could assist the offering of services based on*

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 692–696, 2013.

the user behaviour in the Social Web. In the scope of my thesis, I consider various dimensions that should be combined for the realisation of the aforementioned conceptual idea as the following section presents.

2 State of the Art

Inspired by the work of Mika [2] regarding the tripartite model of ontologies for social networks (i.e. Actor-Concept-Instance), this thesis aims to define workflow patterns that are usable and adaptable to the needs of the Social Web. Moreover, there has already been considerable interest in the social network interactions, like the work in [3] which coined the 'social property' as a network of activity theory concepts with a given meaning. Social properties are considered as patterns that *"represent knowledge grounded in the social sciences about motivation, behavior, organization, interaction..."* [3]. The results of this research direction combined with the generic workflow patterns described in [4] are highly relevant with the objectives of the proposed approach and the materialisation of the communication patterns. Furthermore, the design of the patterns is related to the collaboration among the various agents as described in [5], in the scope of the social workflows. Besides the social properties, the work described in [6] introduces the usage of ontologies in the modelling of the user's activities in conjunction with content and sentiment. In the context of our approach, modelling behaviours will enable us to identify patterns in communication problems and understand the dynamics in discussions in order to discover ways of engaging more efficiently with the public in the Social Web. Extending the state of the art work in the existing behaviour modelling methods, the contribution will be the specialisation of the ontology towards specific domains, in respect to the datasets of the use cases.

Several researchers have proposed the realisation of context-aware workflows [7] and social collaboration processes [8], which are related to our initial idea of modelling the related actors and artifacts in order to enable adaptiveness and personalization in the communication patterns infrastructure. Moreover, research in the area of semantics regarding the retrieval of workflows [9] as well as the semantic annotation paradigms like described in [10],[11] is considered relevant to our planned contribution. The contribution could be considered as threefold: application of user and behaviour modelling methods on certain domains, design and implementation of workflow patterns specific for the communication in the Social Web, and context-aware adaptation and evolution of the patterns.

3 Approach

The proposed approach is closely related with the objectives and contributions as described in section 2. Figure 1 demonstrates the approach that will be followed to materialise the concept of the communication patterns. The design and implementation of the workflow patterns will be based on the open communication issues of the use cases that have been extracted from the datasets. According

to Figure 1, the first steps include the modelling of the user behaviour in order to understand it and retrieve her/his activities A that are related to a specific context. The next step is to match A with existing patterns (we assume that some basic patterns will have already been defined) by exploiting the benefits of inference as some semantic meta-data will be stored in conjunction with every pattern definition. These patterns while being predefined and seemingly static, will also be able to adapt in the context of specific cases by employing context-aware paradigms like those represented by the papers mentioned in the state of the art, in section 2.

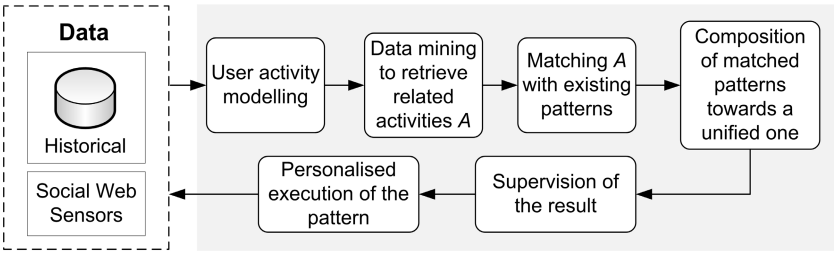


Fig. 1. General flow in the communication patterns' infrastructure

The research will be based on datasets from various domains, e.g. tourism including hotel reviews and points-of-interest reviews. Employing the Behaviour Ontology described in [6], an extented model (specific to the domains) will be applied on the datasets in order to understand the dynamics in the discussions of the users. Thus, helping to address possible issues and preventing the churning [12] of the users from the provided services. For example, a typical user activity could be the post of a bad review for a service (e.g. accommodation), which the affected enterprise should be able to handle in an appropriate way. The reaction from the side of the enterprise could be assisted by a communication pattern specific for this context.

Various technologies are available for modelling workflows like YAWL and BPEL. The YAWL ecosystem [13] (i.e. language, workflow engine, etc.) has been designed in an Open Source manner [14], which perfectly fits the proposed approach as it enables the extension of the workflow engine due to its high modularity. However, the contributions of the proposed approach will remain independent of platform in order to be easily adoptable on other research initiatives in the future.

The effectiveness of the designed patterns can be measured by employing and extending the findings of the work presented in [15] regarding metrics of the user engagement in terms of popularity, activity and loyalty. Moreover, research in the area of Quality of Service (QoS), e.g. [16], is considered to be important for our approach in the scope of evaluating the impact of the communication pattern infrastructure.

4 Methodology

The planned methodology of the proposed approach includes various steps that are interconnected, albeit they could run in parallel in order to enhance the overall research process.

1. *Idea Initialization*
   In the first phase of the research work, the usage scenarios will be defined as well as the pilots that will be used from various domains in order to evaluate the research results. Moreover, existing approaches regarding the various dimensions of the research plan will be evaluated in order to find gaps and specify our contribution.
2. *Specification of Communication Patterns Infrastructure*
   The state of the art analysis and the requirements analysis will qualify in this phase to the specification of the infrastructure that will support the concept of communication patterns.
3. *Implementation*
   The next step is to apply the design of step 2 and the theoretical background that has been acquired from the aforementioned steps in the development of the algorithms that will be used by the various components and the ontology for the behaviour modelling. The implementation phase will be assisted by experiments that will run in an iterative way in small-scale and per component in order to recognize problems and bottlenecks at an early-stage in the approach.
4. *Final Evaluation*
   The final validation of the results will consist of various indicators and measurements that will be gathered from user studies related to the domains of the use cases (e.g. tourism, social networking, product quality management or brand management).

The aforementioned methodology layers are reflected in the research working plan which is presented in section 5.

5 Schedule

The schedule of my thesis is based on the realization of the steps mentioned in section 4. The first phase of studying the state of the art in the research fields of my contribution has already started. In addition, the study of the state of the art runs in parallel with the specification of the usage scenarios, the retrieval of the datasets and the requirements analysis in order to specify the needs and the expectations in this early-stage. The next step is the theoretical design of the communication patterns infrastructure, which will have finished until the end of the second quarter of 2013 and the first research results should have come out of this process. The next two quarters of 2013 will be exploited in refining the theoretical part, the algorithms and the ontology for the behaviour modelling. In case that all the above-mentioned goals are achieved till the end of 2013, the refineminent of the running prototype will take place in the following

year (i.e. 2014) as well as the final validation in conjunction with the writing of the dissertation document. Throughout this research initiative several high-level conferences, workshops and journals have been considered for submissions.

References

1. Kaplan, A.M., Haenlein, M.: Users of the world, unite! the challenges and opportunities of social media. Business Horizons 53(1), 59–68 (2010)
2. Mika, P.: Ontologies are us: A unified model of social networks and semantics. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 522–536. Springer, Heidelberg (2005)
3. Fuentes-Fernandez, R., Gomez-Sanz, J.J., Pavon, J.: User-oriented analysis of interactions in online social networks. IEEE Intelligent Systems 27, 18–25 (2012)
4. van Der Aalst, W.M.P., Ter Hofstede, A., Kiepuszewski, B., Barros, A.: Workflow patterns. Distributed and Parallel Databases 14(1), 5–51 (2003)
5. Dorn, C., Taylor, R., Dustdar, S.: Flexible social workflows: Collaborations as human architecture. IEEE Internet Computing 16(2), 72–77 (2012)
6. Rowe, M., Angeletou, S., Alani, H.: Predicting discussions on the social semantic web. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) ESWC 2011, Part II. LNCS, vol. 6644, pp. 405–420. Springer, Heidelberg (2011)
7. Wieland, M., Kopp, O., Nicklas, D., Leymann, F.: Towards context-aware workflows. In: CAiSE, pp. 11–15 (2007)
8. Liptchinsky, V., Khazankin, R., Truong, H.-L., Dustdar, S.: A novel approach to modeling context-aware and social collaboration processes. In: Ralyté, J., Franch, X., Brinkkemper, S., Wrycza, S. (eds.) CAiSE 2012. LNCS, vol. 7328, pp. 565–580. Springer, Heidelberg (2012)
9. Bergmann, R., Gil, Y.: Retrieval of semantic workflows with knowledge intensive similarity measures. In: Ram, A., Wiratunga, N. (eds.) ICCBR 2011. LNCS, vol. 6880, pp. 17–31. Springer, Heidelberg (2011)
10. Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., Ciravegna, F.: Semantic annotation for knowledge management: Requirements and a survey of the state of the art. Web Semantics: Science, Services and Agents on the World Wide Web 4(1), 14–28 (2006)
11. Sivashanmugam, K., Verma, K., Sheth, A., Miller, J.: Adding semantics to web services standards. In: Proceedings of the International Conference on Web Services, pp. 395–401 (2003)
12. Karnstedt, M., Rowe, M., Chan, J., Alani, H., Hayes, C.: The effect of user features on churn in social networks. In: Third ACM/ICA Web Science Conference (2011)
13. van der Aalst, W.M.P., Aldred, L., Dumas, M., ter Hofstede, A.H.M.: Design and implementation of the YAWL system. In: Persson, A., Stirna, J. (eds.) CAiSE 2004. LNCS, vol. 3084, pp. 142–159. Springer, Heidelberg (2004)
14. Adams, M., ter Hofstede, A., La Rosa, M.: Open source software for workflow management: the case of YAWL. IEEE Software 28(3), 16–19 (2011)
15. Lehmann, J., Lalmas, M., Yom-Tov, E., Dupret, G.: Models of user engagement. In: Masthoff, J., Mobasher, B., Desmarais, M.C., Nkambou, R. (eds.) UMAP 2012. LNCS, vol. 7379, pp. 164–175. Springer, Heidelberg (2012)
16. Cardoso, J., Sheth, A., Miller, J., Arnold, J., Kochut, K.: Quality of service for workflows and web service processes. Web Semantics: Science, Services and Agents on the World Wide Web 1(3), 281–308 (2004)

An Architecture to Aggregate Heterogeneous and Semantic Sensed Data

Amelie Gyrard*

Eurecom, Sophia Antipolis, France
amelie.gyrard@eurecom.fr

Abstract. We are surrounding by sensor networks such as healthcare, home or environmental monitoring, weather forecasting, etc. All sensor-based applications proposed are domain-specific. We aim to link these heterogeneous sensor networks to propose promising applications. Existing applications add semantics to the sensor networks, more specifically, to the context, rather than to the sensed data. We propose an architecture to merge heterogeneous sensor networks, convert measurements into semantic data and reason on them.

Keywords: Semantic Sensor Networks, Semantic Web technologies, Resource Description Framework (RDF), Linked Open Data, Ontologies, Reasoning, Sensors.

1 Motivation and Research Questions

Sensor networks are used in a great deal of realms such as home monitoring, environmental monitoring (e.g., weather forecasting), health monitoring (e.g., pacemaker, brain waves), vehicular networks, etc. Each application focuses on a specific sensor network. We intent to link these existing heterogeneous sensor networks to provide new applications. For example, by merging the following sensor networks: the smart kitchen, the weather forecasting and the health we could propose a recipe according to ingredients available in the kitchen, the weather and the user's health (diets, diseases, allergies, emotional state). Merging heterogeneous sensor networks is a difficult task due to heterogeneous protocols, heterogeneous data format and the lack of description of measurements. For example, a temperature measurement is related to a body temperature or an outside temperature, with a body temperature we can deduce if the person is sick, this is not the case with the outside temperature.

The main challenges of this motivating scenario are: (1) manage heterogeneous data from sensor networks, (2) convert sensor measurements into semantic data using semantic web technologies and (4) reason on these semantic data.

* Supervisors: Christian Bonnet and Karima Boudaoud.

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 697–701, 2013.

2 State of the Art

SensorMasher [1] and the SemsorGrid4env [2] projects both manipulate environmental sensed data. Coyle et al. [3] propose semantic sensor networks for smart homes. Sense2Web [4] is a Linked Data Platform to publish sensor data and to link them to existing resources on the Web. SWAP (Sensor Web Agent Platform) [5] extracts sensor data automatically. The SSN (Semantic Sensor Network) Ontology [6] describes sensors and their measurements. The following sensor ontologies are specific to environmental sensors and do no not focus on the type of the measurement and the unit: Csiro[1] OntoSensor[2], Cesn[3], Sensei[4], SemSOS[5], OOSTethys[6]. SenML [7] and SWE (Sensor Web Enablement) [8] are protocols to retrieve sensor measurements. SenML is a lightweight protocol, SWE is more difficult to deploy but provides interesting services to manage sensors such as be alerted when a specific event occurred by email. Machine-to-Machine (M2M) means that computers can communicate with each other without human intervention. The M2M ETSI architecture [9] is an architecture to manage heterogeneous sensor networks and communication protocols. They propose to add semantics to the context rather than to the measurements.

Existing works focus on a specific sensor network: smart home, smart kitchen, weather forecasting or environmental monitoring. They design a domain ontology without be linked to the existing ones and add semantics to the context (i.e., shut off the light is the room is empty). There are a numerous sensor ontologies and domain ontologies but they are designed without considering the existing ones and propose to add semantics to the context rather than to the measured data. Further, they do not provide semantic-based reasoning (machine learning or recommender systems) on measurements.

3 Approach

We propose an architecture (Fig. 1) to get sensor measurements (sensor gateways), to annotate heterogeneous measurements with semantics (aggregation gateways) and reason on them (semantic-based applications). Our architecture is inspired by the M2M ETSI architecture. We have in mind a distributed architecture, and propose high energy treatments on the cloud computing if necessary. Our sensor gateways retrieve sensor measurements through the SenML protocol. Our aggregation gateways convert sensed data into semantic measurements using semantic web technologies

[1] http://www.w3.org/2005/Incubator/ssn/wiki/SensorOntology2009
[2] http://mmisw.org/ont?form=rdf&uri=http://mmisw.org/ont/univmemphis/sensor
[3] http://www.cesn.org/sensor/cesn.owl
[4] purl.oclc.org/net/unis/ontology/sensordata.owl
[5] http://archive.knoesis.org/research/semsci/application_domain/sem_sensor/ont/sensor-observation.owl
[6] http://mmisw.org/ont?form=rdf&uri=http://mmisw.org/ont/mmi/20090519T125341/general

(RDF, RDFS, OWL and domain ontologies). Semantic-based applications link our semantic measurements to the Linked Open Data[7] and perform reasoning (inference engine, rules, machine learning, recommender systems).

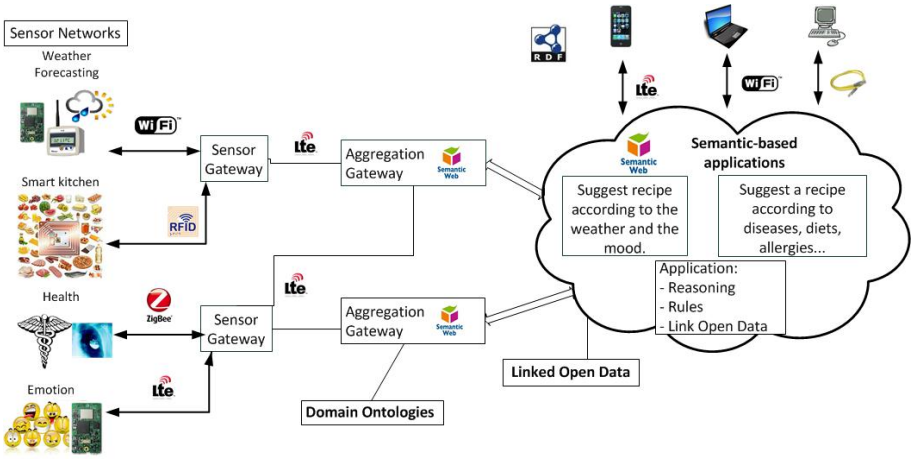


Fig. 1. The proposed architecture

In our scenario, we have two aggregation gateways, the former stores semantic data related to the weather, ingredients and health, the second manages semantic data related to the health and the brain waves. Semantic-based applications merge and query aggregation gateways to provide new services such as suggest the menu for dinner adapted to the weather, the season, available ingredients in the kitchen and the user's health (diseases, diets, allergies, emotional state).

We design the SenMESO (sensor Measurements Ontology)[8] to convert automatically heterogeneous sensor measurements into semantic data. This ontology acts as a hub to merge heterogeneous measurements and domain ontologies. Our ontology describes the measurement concept: a measurement has a name, a value, a unit and a type. SenMESO is linked to numerous domain ontologies to obtain additional information: health (ontoreachir[9]), sensor (SSN[10]), meteo (AWS[11]), smart home (dogont[12]), emotion[13], etc. We aim at constructing a tool to update automatically this ontology with other domain ontologies. Semantic measurements are linked to the linked Open Data to obtain additional information. An example is to link our food measurements to the SmartProduct[14] datasets defining a great deal of ingredients and recipes.

[7] http://linkeddata.org/
[8] http://sensormeasurement.appspot.com/
[9] Search on google (filetype:owl Ontoreachir).
[10] http://www.w3.org/2005/Incubator/ssn/ssnx/ssn
[11] http://www.w3.org/2005/Incubator/ssn/ssnx/meteo/aws.owl
[12] http://elite.polito.it/ontologies/dogont.owl
[13] http://emotion-ontology.googlecode.com/svn/trunk/ontology/
[14] http://projects.kmi.open.ac.uk/smartproducts/ontology.html

We want to create a generic algorithm to reason on the heterogeneous semantic data using machine learning, recommender system and semantic tools.

4 Research Methodology

We designed the architecture at the beginning of the thesis and an ontology to convert heterogeneous measurements into semantic data. We evaluate our ontology by using it in the prototype implementation.

Current steps are to work on the refinement of this architecture and the ontology. We are working on updating automatically this ontology with new domain ontologies. We are implementing a prototype to evaluate the components of our architecture (sensor gateway, aggregation gateway) and the M2M applications.

Future steps are to integrate a semantic-based recommender system on semantic measurements to propose applications as presented in the first section. Our prototype will be integrated to the Com4Innov[15] platform deploying a real architecture with heterogeneous sensors and communication protocols (4G). Finally, we will evaluate the performance of the prototype and the real architecture, more precisely, algorithms implemented to aggregate, convert sensed data and reason on them.

5 Results

We have implemented a first prototype to validate the proposed architecture. The sensor gateways[16] retrieve raw measurements and return them according to the SenML protocol. We obtain simple measurements: the name, the value, the unit, and the date (i.e., the temperature is 5°C, 250 grams of butter). The aggregation gateways convert XML data into RDF data. We have implemented the SenMESO ontology to annotate measurements with semantics. The M2M applications reason on semantic measurements to propose an application as the one presented in the first section. The architecture has been implemented with the following technologies: the Java language, Google Application Engine (GAE), the Jena framework, HTML5 and JavaScript. Both the prototype and the ontology are available online[17]. The final version of the prototype will be integrated to the Com4Innov project, to test it in a real environment with heterogeneous sensors and protocols.

6 Conclusion and Future Works

We proposed to merge heterogeneous semantic sensor networks. We annotate measurements with semantics rather than add semantics to the context. Currently, we are

[15] http://www.com4innov.com/platforms_presentation.en.htm

[16] http://emulator-box-servi ces.appspot.com/senmladmin/ahdzfmVtdWxhdG9yLWJveC1zZXJ2aWNlc3IbCxIJWm9uZUFkbWluIgxBbWVsaWVDb3JuZXIM/edit

[17] http://sensormeasurement.appspot.com/

working on the refinement of the architecture, the distributed aspect and the implementation. Future works are to integrate semantic-based machine learning algorithms and recommender systems to reason on heterogeneous semantic measurements. We are also interesting in the security aspects. We are designing a semantic-based security application[18] to help a non-expert in security to secure his/her application, by suggesting the best security mechanism to use.

References

1. Le-Phuoc, D., Hauswirth, M.: Linked open data in sensor data mashups. In: Second International Workshop on Semantic Sensor Networks Workshop (SSN 2009), in conjunction with the 8th International Semantic Web Conference (ISWC 2009) (2009)
2. Gray, A., Galpin, I., Fernandes, A., Paton, N., Page, K., Sadler, J., Koubarakis, M., Kyzirakos, K., Calbimonte, J.-P., Corcho, O., et al.: Semsorgrid4env architecture–phase i. Deliverable D1. 3v1, SemSorGrid4Env (2009), http://www.semsorgrid4env.eu/
3. Coyle, L., Neely, S., Stevenson, G., Sullivan, M., Dobson, S., Nixon, P.: Sensor fusion-based middleware for smart homes. International Journal of Assistive Robotics and Mechatronics 8(2), 53–60 (2007)
4. Barnaghi, P., Presser, M.: Publishing linked sensor data. In: 3rd International Workshop on Semantic Sensor Networks (2010)
5. Moodley, D., Simonis, I.: A new architecture for the sensor web: the SWAP-framework. In: Semantic Sensor Networks Workshop, A workshop of the 5th International Semantic Web Conference ISWC 2006, Athens, Georgia, USA, November 5-9 (2006)
6. Compton, M., Barnaghi, P., Bermudez, L., Garcia-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., Huang, V., Janowicz, K., Kelsey, W.D., Phuoc, D.L., Lefor, L., Leggieri, M., Neuhaus, H., Nikolov, A., Page, K., Passant, A., Sheth, A., Taylor, K.: The SSN ontology of the semantic sensor network incubator group (2011)
7. Jennings, C.: Media Type for Sensor Markup Language (SENML), draft-jennings-senml-09 (July 2012) (work in progress)
8. Botts, M., Percival, G., Reed, C., Davidson, C.: OGC sensor web enablement: overview and high level architecture. Open Geospatial Consortium White Paper, OGC 06-052r2 (2006)
9. Boswarthick, D.: M2M activities in ETSI. Presentation Report (July 2009)

[18] http://securitytoolbox.appspot.com/

Linked Data Interfaces for Non-expert Users

Patrick Hoefler

Know-Center, Graz, Austria
phoefler@know-center.at

Abstract. Linked Data has become an essential part of the Semantic Web. A lot of Linked Data is already available in the Linked Open Data cloud, which keeps growing due to an influx of new data from research and open government activities. However, it is still quite difficult to access this wealth of semantically enriched data directly without having in-depth knowledge about SPARQL and related semantic technologies. The presented dissertation explores Linked Data interfaces for non-expert users, especially keyword search as an entry point and tabular interfaces for filtering and exploration. It also looks at the value chain surrounding Linked Data and the possibilities that open up when people without a background in computer science can easily access Linked Data.

Keywords: linked data, interfaces, semantic web, sparql, rdf.

1 Motivation

The Linked Open Data cloud provides an impressive wealth of semantically enriched, openly available Linked Data. However, this Linked Data is basically only accessible for experts in semantic technologies who know how to write SPARQL queries. And even for those who know how to use SPARQL, it can be quite laborious at times, especially while trying to explore an unknown SPARQL endpoint.

Therefore, the goal of this dissertation is to research easy-to-use interfaces for accessing Linked Data. These interfaces should be usable without any prior knowledge of SPARQL or other semantic technologies.

2 Research Questions

The central research question is:

- **How can Linked Data interfaces for non-expert users look like?**

Subsequent research questions are:

- How can current search engine paradigms be used for Linked Data?
- How can Linked Data be displayed, filtered, and explored in tabular form?
- How can value be created when non-expert users can access Linked Data?

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 702–706, 2013.

3 Approach and Research Methodology

The first phase of this dissertation dealt with some of the fundamental problems related to the question of easy access. In [1], we looked into ways to turn search keywords into URIs and applied the technique in a simplified end-user interface for accessing Linked Data. This approach was extended into CAF-SIAL, a proof-of-concept application described in [2] and [3]. CAF-SIAL helped users to search information about concepts in the Linked Open Data cloud without having to know any of the mechanics of the Semantic Web.

We also created a Linked Data value chain model [4] that conceptualized the actors and processes in the Linked Data ecosystem. Though we did not pursue that strand of research further at the time, it turned out to become much more relevant to this dissertation later on.

The second phase of this dissertation is closely related to the CODE project [5], a research project funded by the European Union. As described in [6], the vision of CODE is to establish a sophisticated ecosystem for Linked Data. The current focus of this dissertation is the Linked Data Query Wizard[1] as part of CODE's Visual Analytics work package [7]. The goal of the work package is to develop a web-based visual analytics platform that enables non-expert users to engage in a visually supported, collaborative analysis of Linked Data, and the Linked Data Query Wizard will play a crucial role in this undertaking.

The current research methodology follows the principles of agile development and rapid prototyping: Small implementation cycles, resulting in new versions of the prototype on a weekly, sometimes daily basis.

4 The Linked Data Query Wizard

The working hypothesis for the Linked Data Query Wizard is: There's not a lot of people who speak SPARQL and are familiar with graph structures. On the other hand, many people know spreadsheet applications like Microsoft Excel. Therefore, the idea is to develop a web-based tool that brings the graph structure of Linked Data into tabular form (see figure 1) and provides easy-to-use interaction possibilities for filtering and exploring Linked Data by using metaphors and techniques the users already know.

4.1 Related Work

Although the Semantic Web has matured in recent years, and semantic technologies have become quite powerful, the Linked Open Data cloud is still only accessible for semantic technology experts and programmers. The problem of easy-to-use interfaces for accessing Linked Data is still largely unsolved. The majority of current tools are not aimed at non-expert users. As an example, the popular Semantic Web search engine Sindice [8] is practically unusable for people without a deep understanding of semantic technologies.

[1] `http://code.know-center.tugraz.at/search`

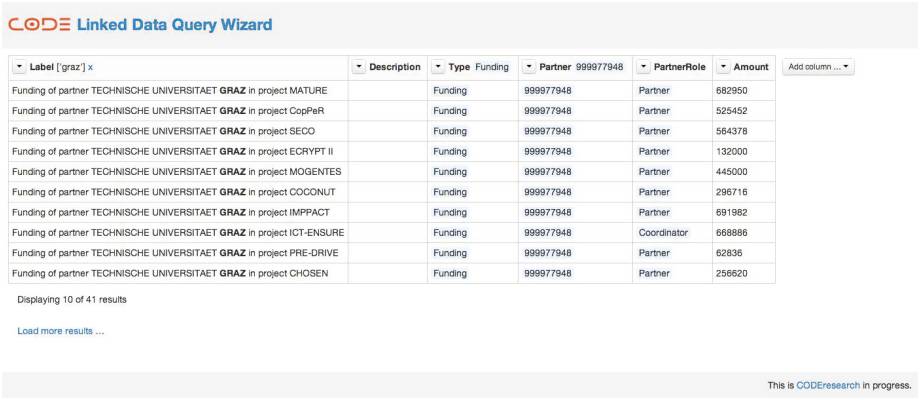


Fig. 1. Early beta version of the Linked Data Query Wizard

Currently, only very few web-based tools use tables for representing Linked Data. One such example would be Freebase Parallax [9]. Although its main feature is the ability to browse sets of related things, it also provides a table view for these result sets. Another online tool that shares similarities with our prototype is the Falcons Explorer [10]. Both tools feature a search box as the main entry point – an idea that is also central to our prototype. However, in both tools, the table view is not the central focus.

Another tool that shares a few similarities with our prototype is OpenRefine [11] (formerly known as Google Refine and Freebase Gridworks). It supports RDF, and there are also extensions such as LODRefine [12] that focus on Linked Data – however, OpenRefine's main focus is cleaning up tabular data, and it's also not available as a web service, even though its main interface is browser-based.

Our prototype also supports the current working draft of W3C's RDF Data Cube Vocabulary [13] which provides a semantic framework for expressing statistical data sets as Linked Data. Data sets that comply with the RDF Data Cube standard can easily be displayed, filtered, and explored using the Linked Data Query Wizard. To the best of our knowledge, there are currently no other tools with a feature set similar to our prototype that support RDF Data Cubes.

4.2 Limitations

Currently, the biggest limitation of the prototype is the rather demanding set of requirements it imposes on the SPARQL endpoints that serve as its back end. One critical feature that is needed for our current approach is support for full-text search. Sadly, full-text search is sorely lacking from the current SPARQL specification, which is why certain SPARQL endpoints have come up with workaround solutions. Therefore, only Virtuoso and bigdata are currently supported as SPARQL endpoints by the Linked Data Query Wizard.

The Linked Data Query Wizard also makes use of certain SPARQL 1.1 features, especially the aggregation functions. The `COUNT()` function is critical and already in use by the current prototype for displaying the number of results for a given query.

4.3 Evaluation

For evaluating the prototype, both formative as well as summative evaluations are planned:

The formative evaluations have already started informally, mostly with project team members, and with increasing feature completeness will also include friendly users outside of the team with a decreasing degree of expertise. The last formative evaluations will include non-expert users.

There are also summative evaluations planned. Some of them will be part of challenges and the planned use of the prototype in university courses, but additional quantitative and qualitative user studies might be conducted as needed.

4.4 Initial Results

The Linked Data Query Wizard is currently available online as an early beta version. In its current form, it offers two entry points: Users can either initiate a keyword search, or they can select any available dataset, represented as an RDF Data Cube. In both cases, the users get presented with the results in tabular form, similar to what they are used from spreadsheet applications. They can choose which columns (i.e. RDF predicates) they are interested in, and they can set filters to narrow down the displayed data.

Though the current functionality of the prototype is still rather limited, first usage experiments have shown that the tool can be helpful in exploring the data and respective data structures of unknown SPARQL endpoints.

5 Conclusions and Future Work

The creation process of this dissertation can be divided into two phases: The early first phase, focusing on keyword search for Linked Data and the value chain of the Linked Data ecosystem, and the current second phase, focusing on the Linked Data Query Wizard, a novel approach for filtering and exploring Linked Data.

The next steps regarding the prototype will be to expand its functionality, focusing on better filter mechanisms as well as more advanced exploration features that incorporate the underlying semantic structure. Additionally, the Linked Data Query Wizard will be integrated with a tool for visualizing Linked Data as well as a semantic enrichment service for turning generic RDF into RDF Data Cubes.

The development of the prototype will continue throughout the rest of the year, leading to a final evaluation at the beginning of 2014.

The writing process of the dissertation has already begun and will intensify as the year progresses. The majority of the dissertation should be finished by the end of 2013, leaving only the final evaluation results as well as the finishing touches for the beginning of 2014.

The dissertation should be finished in the first half of 2014.

Acknowledgments. Parts of this PhD thesis are being developed within the CODE project at the Know-Center, Graz, Austria. The CODE project is funded by the EU Seventh Framework Programme, grant agreement number 296150. The Know-Center is funded within the Austrian COMET Program – Competence Centers for Excellent Technologies – under the auspices of the Austrian Federal Ministry of Transport, Innovation and Technology, the Austrian Federal Ministry of Economy, Family and Youth, and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency (FFG).

References

1. Latif, A., Afzal, M.T., Hoefler, P., Us Saeed, A., Tochtermann, K.: Turning Keywords into URIs: Simplified User Interfaces for Exploring Linked Data. In: Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, pp. 76–81. ACM, New York (2009)
2. Latif, A., Afzal, M.T., Us Saeed, A., Hoefler, P., Tochtermann, K.: CAF-SIAL: Concept Aggregation Framework for Structuring Informational Aspects of Linked Open Data. In: Proceedings of NDT 2009, Ostrava (2009)
3. Latif, A., Us Saeed, A., Hoefler, P., Tochtermann, K., Afzal, M.T.: Harvesting Pertinent Resources from Linked Open Data. Journal of Digital Information Management 8(3), 205–212 (2010)
4. Latif, A., Hoefler, P., Stocker, A., Us Saeed, A., Wagner, C.: The Linked Data Value Chain: A Lightweight Model for Business Engineers. In: Proceedings of I-KNOW 2009 and I-SEMANTICS 2009. Verlag der Technischen Universität Graz (2009)
5. CODE: Commercially Empowered Linked Open Data Ecosystems in Research, `http://code-research.eu/`
6. Stegmaier, F., Seifert, C., Kern, R., Hoefler, P., Bayerl, S., Granitzer, M., Kosch, H., et al.: Unleashing Semantics of Research Data. In: Proceedings of the Second Workshop on Big Data Benchmarking (WBDB 2012), Pune, India (2012)
7. CODE: Visual Analytics, `http://code-research.eu/visual-analytics`
8. Tummarello, G., Delbru, R., Oren, E.: Sindice.com: Weaving the open linked data. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 552–565. Springer, Heidelberg (2007)
9. Huynh, D.F., Karger, D.R.: Parallax and Companion: Set-based Browsing for the Data Web. In: WWW Conference. ACM (2009)
10. Cheng, G., Wu, H., Gong, S., Ge, W., Qu, Y.: Falcons Explorer: Tabular and Relational End-user Programming for the Web of Data. Semantic Web Challenge (2010)
11. OpenRefine, `https://github.com/OpenRefine`
12. LODRefine, `http://code.zemanta.com/sparkica/`
13. W3C: RDF Data Cube Vocabulary, `http://www.w3.org/TR/vocab-data-cube/`

Event Matching Using Semantic and Spatial Memories

Majed Ayyad

IT Department (DISI),
University of Trento,
Via Sommarive 14, Trento, I-38123
ayyad@disi.unitn.it

Abstract. We address the problem of real-time matching and correlation of events which are detected and reported by humans. As in Twitter, facebook, blogs and phone calls, the stream of reported events are unstructured and require intensive manual processing. The plethora of events and their different types need a flexible model and a representation language that allows us to encode them for online processing. Current approaches in complex event processing and stream reasoning focus on temporal relationships between composite events and usually refer to pre-defined sensor locations. We propose a methodology and a computational framework for matching and correlating atomic and complex events which have no pre-defined schemas based on their content. Matching evaluation on real events show significant improvement compared to the manual matching process.

1 Motivation and Problem

In recent years a special attention was given to streamed events and stream reasoning [1] [2][13]. A special type of noisy data streamed for real-time reasoning are events which are detected and reported by humans to actionable knowledge bases in multi-tier responding agencies through different services such as Twitter, facebook, phone calls, Microblogs and other similar sources. A common example on this scenario is the stream of incoming phone calls to the operation room of civil police as depicted in Fig. 1. In a standard operation room, operators only register incoming calls, where a second tier of commanders evaluate these calls, support them, if possible, with other information probed from news, blogs and web pages before taking any actions. The second tier is only interested with events that are valid for processing in a time-window. For every new event, they continuously evaluate it against all events in the past time-window in order to find similar clusters of events.

The general main two continuous queries that could be registered on the stream of calls are : **Query 1.** "Compare each incoming event with all previous events logged during the last 5 minutes, then cluster similar events before taking any decision". For the example given in Fig. 1, the query could be translated to "Are these three events the same?". **Query 2.** " Compare each incoming event with all previous events logged during the last hour… Then predict potential new events ".

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 707–711, 2013.

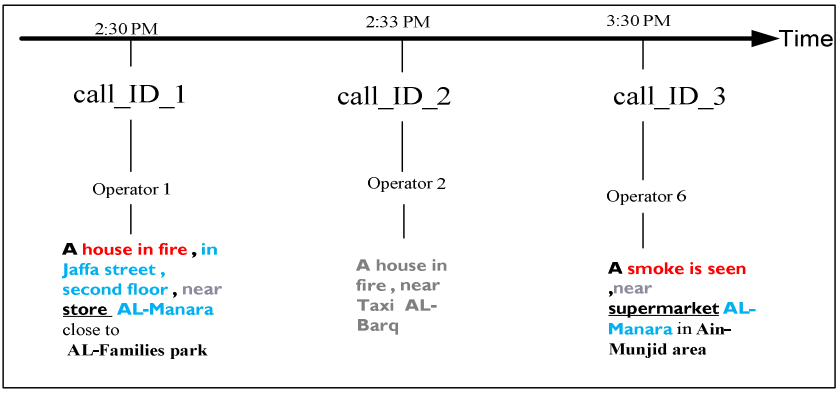


Fig. 1. Calls to the police operation room

To generalize the scenario, and given a stream of events $\{e_1,...e_n\}$ where the structure of e_i, motivated by Davidson convention[4], is of the format $\exists e$ (**Event**(e) $\wedge$ **Agent**(e; an agent) $\wedge$ **Recipient** (e; a recipient) **Time**(e; a time) $\wedge$ **Place**(e; a location) $\wedge$ **Instrument** (e; an instrument)). This format which is illustrated in Fig. 2 also serves as the upper Ontology for events

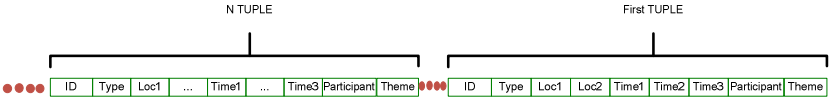


Fig. 2. Stream of non-equal event tuples

The main questions to be answered are :

1. Given a set of atomic events find the similarity between these events in real-time. Similarity is computed as 3-tuples $< e_1, e_2, R >$, where R is expressed as equivalence $(\equiv)$, partially-matched $(\subset)$, and mismatch $(\perp)$.
2. Given a set of occurring events $\{e_1,...e_n\}$ and other historical occurrences, find or infer what pattern of events is occurring.

2 State of the Art

Many approaches followed a content based event matching using different methods which could be summarized as follows **(a) Information retrieval**: [5] use information retrieval techniques for computing events similarity where the event context is treated as a document and the tuple attribute values correspond to document terms. A similar approach was used by The Entity Name System (ENS) [6][7]. **(b) Machine-learning algorithms:** [8] uses machine-learning algorithms to classify events using three groups of features: statistical features, keyword features and word context features. [8]

demonstrated that through event mining, it is possible to detect the location and time for earthquake events by exploiting the real-time nature characteristic of Twitter. The main disadvantage of this method is the need for a large number of events for training, but once learned this method could be used to create models for events correlation.**(c) Predicate-based matching:** The content-based event matching problem was intensively studied in publish-subscribe infrastructure. Where an event to satisfy a subscription, every predicate in the subscription should be matched by some pair in the event [8]. The main disadvantage of predicate-based matching is that predicates should be pre-defined in advance. **(d) Pattern matching (Rete):** The Rete algorithm [9], originally used for production rule systems, is an efficient solution to the facts-rules pattern matching problem. The basic Rete algorithm was extended to accommodate for temporal operators [10][11]. Our approach learns from rete network, but instead of building a network from rules, we build a network from the Ontology and spatial locations.

3 Proposed Approach and Methodology

Our methodology to match and correlate events is based on the content of these events. The methodology approaches the problem from a representational as well as a computational viewpoint as shown on Fig 3. The framework consists of the following components :

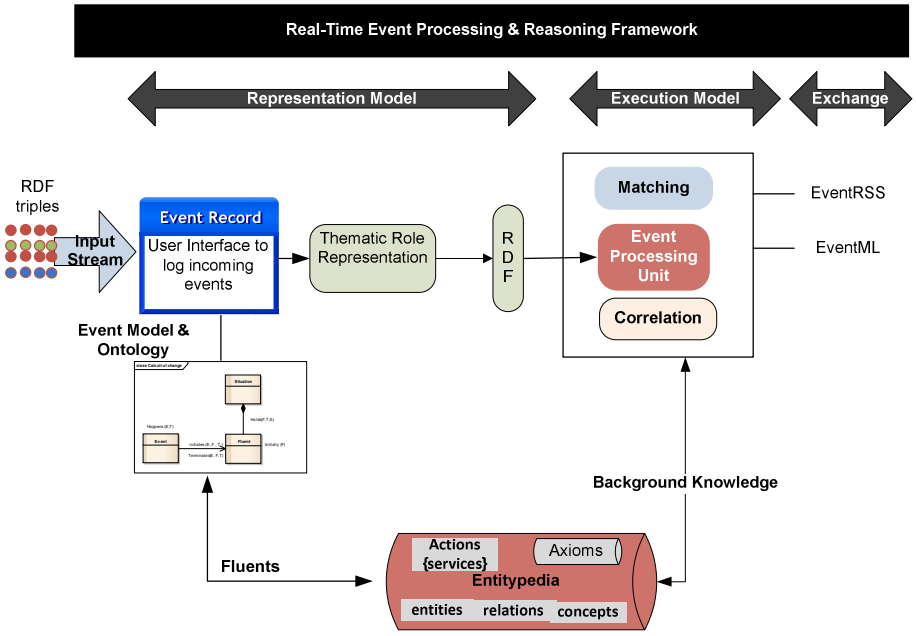


Fig. 3. Real-time event processing framework

(a) **Event Detection and Logging – Controlled GUI :** The first stage, after event detection, starts by logging the event using a controlled natural language graphical user interface designed to capture temporal and space properties of an event based on pre-defined ontology and model. To build such a model we depend on analyzing the linguistic, Ontological and semantic properties of events and treated events as 4 dimension entities.

(b) Thematic Role Model Builder : The thematic role model aims at representing each event in a form that allows us to correlate and match using the thematic roles of events. Each thematic roles describes the "mode of participation" in an event for each argument of a predicate [3].

(c) **Event Model : Semantic and Spatial Graph Representation:** We use two graphs called semantic and spatial memories that are appropriate to perform semantic matching and spatial reasoning about the streamed events. Semantic and spatial memories are built from a central knowledge base of linked entities called Entitypedia[1]. we propose typed composite graphs with inheritance and containment to specify the event structures. After building the network, events asserted from the stream are used to activate these memories at runtime

(d) **Event Query Language :** Clusters of events could be viewed at different granularity based on the typed graphs and their containment relationships.

4 Initial Results and Conclusions

At this stage, we have collected a sufficient number of entities and event types. We collected entities of different types (person, organization and location) from real-life databases. So far we analyzed the meta-data and attributes used by 11 municipalities, 3 Ministries, and two private sector organizations to identify the main entities, their attributes and their instances. We collected 4,358,569 from one country. We designed a preliminary user interface based on the event upper Ontology. For relations between locations, we use the region connection RCC8[12] for qualitative spatial representation and reasoning. For the matching algorithms, we took all the locations in one city and built the RCC8 relationships between these locations. A proof-of-concept prototype for the matching problem was implemented and tested. The initial results show the ability of the system to match hundreds of events efficiently. The set of events that the system couldn't match are collected in a conflict memory. The efficiency of the matching algorithm depends on the number of entities used to build the event networks.

To evaluate the performance of the classification algorithm, we are interested in the algorithm's ability to correctly predict or separate the classes of matched events, partially matched events or non-matched events. To calculate precision and recall we need a ground truth dataset. This data set is under development from multiple sources. During the last six months, events are logged manually on the system from phone

[1] http://entitypedia.org/

calls and two other online news. Crowdsourcing annotations will be used to label events. Disagreement between annotators on event types, spatial and temporal relationships will be evaluated to enhance the parameters of the algorithm.

5 Remaining Work

Still we are working on the optimization of the matching algorithm, specially how to apply different strategies when new token is passed to the event network . Techniques to validate the event ontology and locations path consistency is under consideration. The query language for event matching and correlation with different operators so the end user can be able to examine and fine-tune the obtained results.

References

1. Ceri, S., Della Valle, E., van Harmelen, F., Stuckenschmidt, H.: It's a Streaming World! Reasoning upon Rapidly Changing Information, November/December 2009, vol. 24(6), pp. 83–89 (2010)
2. Anicic, D., Fodor, P., Rudolph, S., Stojanovic, N.: EP-SPARQL: a unified language for event processing and stream reasoning. In: Proceedings of the 20th International Conference on World Wide Web, Hyderabad, India, March 28-April 01 (2011)
3. Carlson, G.N.: Thematic Roles and the Individuation of Events. In Events and Grammar 70, 35–52 (1998), Key: citeulike:3137321
4. Davidson, D.: The individuation of events, p. 179 (1985)
5. Kwon, Y., Lee, W.Y., Balazinska, M., Xu, G.: Clustering Events on Streams Using Complex Context Information. In: Proc. ICDM Workshops, pp. 238–247 (2008)
6. Bouquet, P., Stoermer, H., Bazzanella, B.: An Entity Name System (ENS) for the Semantic Web. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 258–272. Springer, Heidelberg (2008)
7. Stoermer, H., Rassadko, N., Vaidya, N.: Feature-Based Entity Matching: The FBEM Model, Implementation, Evaluation. In: Pernici, B. (ed.) CAiSE 2010. LNCS, vol. 6051, pp. 180–193. Springer, Heidelberg (2010), http://www.springerlink.com/content/t784745m2841n52j/
8. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake twitter users: Real-time event detection by social sensors. In: WWW (2010)
9. Forgy, C.L.: Rete: A fast algorithm for the many pattern/many object pattern match problem. Artificial Intelligence 19, 17–37 (1982)
10. Berstel, B.: Extending the RETE Algorithm for Event Management. In: Proc. of 9th Int. Symp. on Temporal Representation and Reasoning (TIME 2002), pp. 49–51. IEEE Computer Society (2002)
11. Walzer, K., Breddin, T., Groch, M.: Relative temporal constraints in the Rete algorithm for complex event detection. In: Proc. of 2nd Int. Conf. on Distributed Event-Based Systems, DEBS 2008, pp. 147–155. ACM (2008)
12. Randell, D.A., Cui, Z., Cohn, A.G.: A Spatial Logic Based on Regions and Connection. In: 3rd International Conference on Knowledge Representation and Reasoning (KR 1992), pp. 165–176. Morgan Kaufmann (1992)
13. Barbieri, D., Braga, F., Ceri, F., Valle, S., Grossniklaus, M.: Querying RDF Streams with C-SPARQL. ACM SIGMOD Record 39(1), 20–26 (2010), http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.172.9010 (retrieved)

Incremental SPARQL Query Processing

Ana I. Torre-Bastida

Tecnalia Research & Innovation
Parque Tecnológico Edif 202
Zamudio, 48170 - Vizcaya
isabel.torre@tecnalia.com

Abstract. The number of linked data sources available on the Web is growing at a rapid rate. Moreover, users are showing an interest for any framework that allows them to obtain answers, for a formulated query, accessing heterogeneous data sources without the need of explicitly specifying the sources to answer the query. Our proposal focus on that interest and its goal is to build a system capable of answering to user queries in an incremental way. Each time a different data source is accessed the previous answer is eventually enriched. Brokering across the data sources is enabled by using source mapping relationships. User queries are rewritten using those mappings in order to obtain translations of the original query across data sources. Semantically equivalent translations are first looked for, but semantically approximated ones are generated if equivalence is not achieved. Well defined metrics are considered to estimate the information loss, if any.

Keywords: Semantic Web, Linked Open Data Sources, query reformulation, query rewriting, ontology mapping.

1 Problem Statement and Research Question

The Linked Open Data (LOD) initiative has made available to the users a large number of data sources from various domains such as education, life sciences, government data, literature, geography and others. Two commonly used approaches for query processing in this context are: 1) to query the different data sources independently, one by one; or 2) to integrate first the data sources into a local centralized warehouse and then to process queries in a centralized way on the warehouse. Both approaches present relevant problems such as the user needed expertise following the first approach and the scalability problems that arise in the second one. In this scenario an alternative approach is appearing, the so called federated approach, in which a query is formulated and its answer is obtained from different sources but with the distinguishing feature that the technical details associated to the distributed query answering process are transparent to the user. The work developed in this thesis is placed in this approach, but our system will have the added feature that the user does not need to have specific knowledge of the language in whichthe different data sources are modeled. We summarize our research question as the following one: *How*

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 712–716, 2013.

can we assist to the user with querying heterogeneous data sources, without the need to be an expert on the ontologies with which they are modeled and returning incremental and satisfactory results for the user?

Consider the following scenario, a music student formulates the following query to a multimedia local source: recording of "Sonata giocosa" by "J. Rodrigo" played by "Marco Socias". Possibly, due to the limited number of records of that source, the answer to that query is empty. Then, the user clicks on not satisfied and requests the system to reformulate the question about the same source. Transparently to the user, the system reconstructs the query as: recording of "Sonata giocosa" by "J. Rodrigo" played by anybody. This time the answer received is a recording of the requested piece from the polish guitarist "Marcin Dylla". The user clicks again on not satisfied and on this occasion asks the system to consult a new source. The system selects another relevant source or the user can select the source from a set provided by the system. In this case, the user leaves the decision in the hands of the system and it choices a source consisting of cd records and reformulates the query as: cd including "Sonata giocosa" and featuring "Marco Socias". This time the answer is the cd with title "Elogio de la guitarra" where "Marco Socias" plays the requested piece. Notice that in those last cases the semantics of the original query has been changed.

According to my proposal, the user formulates a query expressed with her preferred vocabulary, waits for an answer and asks for more answers if she is not satisfied with those received. Then, the system does its best to satisfy the user. If semantically equivalent translations of the original query are not achievable on different sources, the system proceeds with approximate translations with the hope to find satisfying answers for the user. The system is able to measure the incurred loss of information with the approximate translation, using metrics from the field of information retrieval, such as precision and recall.

The novel contribution that I consider is: *An innovative query approach that provides the answers by accessing different data sources, expressed with different vocabularies, in an incremental way guided by the user. Source mappings are used for issuing translations of the original query and a measure of loss of information incurred in the intended translation is provided in the case that it occurs.*

2 State of the Art

The Sparql query processing over heterogeneous data sources is an extensive research field in the Semantic Web community. Currently, many systems (DarQ[6], FedX[8]) deal with query federation on heterogeneous datasources of the Web of Data[1]. But the federated approach has a fundamental difference with ours, this is the need for the users to know the ontologies with which are described the datasets and write the query in their model. Our approach is more flexible and useful to the user who only knows his dataset domain and languages, being the system responsible of rewrite the query in terms of the ontologies of other additional interesting datasets.

[1] Web of Data - (`http://richard.cyganiak.de/2007/10/lod/`)

Our study is therefore closer to the works that are focused on SPARQL query rewriting and reformulation. Although we present significant innovations in a domain such as the semantic web, in which has hardly developed studies on this topic. Makris et. al.[5] is quite close to our approach. A formal model for RDF triple patterns rewriting is defined. A quite expressive specific mapping language based on Description Logics constructs is defined and used for the query rewriting. Nevertheless, the rewriting of triple patterns is not dependant on mapping relationships (i.e. equivalence or subsumption). These relationships affect only the evaluation results of the rewriten query over the target ontology. Therefore, they do not take into account the estimation of loss in precision or loss in recall. Moreover, it is not clear what is done when there are not enough mapping expressions to rewrite every term of the source query.

On query relaxation field, there are studies like Hurtado et al. [3], where a new clause of SPARQL, called "RELAX", is introduced for make queries more flexibles by a logical relaxation of the conditions enclosed by the clause. This approach is far from our study, because they are not focused on translating the entire query and extend it with other data sources, but in generalize some conditions of it into the same dataset.

Outside the areas of query rewriting or relaxation, Herzig's article [2] presents similar objectives to ours, regarding the goal of query reusing for consult additional datasets. They make a ERM(Entity relevance model) that contains the structure and content of the results needed to answer a query and thus it can be used to transfer the query to other datasets. It has the disadvantage of allowing only the query of entities.

Finally a differentiating aspect of our system is the measure of the loss of information. For compute it we adapt the approach presented by Salton [7] to estimate the information loss when a term is substituted by an expression. We use the metrics precision and recall originating from Information retrieval [9], [1]. There are other metrics like similarity [4], distance between two ontology concepts, that we are studying to adapt too to our approach.

3 Proposed Approach

My purpose is to exploit RDF-ied sources, being they native RDF Linked Open Data sources or having an RDF scheme wrapping with non RDF data source (e.g. relational database with appropriate RDF scheme mapping). Once the original SPARQL query is received, a SPARQL query engine is launched on the by default dataset. After receiving the answer, there is the possibility to ask for more answers. In that case the original query can be sent to different data sources that share the vocabulary used in the query. But if that chance is not available or its answers are not enough, then a query rewriting process begins. Different choices are possible depending on the user decision: 1) to rewrite the original query (slightly changing its semantics) over the same source but looking for different answers to those previously obtained, 2) to try to rewrite the query using another related source with different vocabularies according to the knowledge

managed by our system, and 3) to ask to the user to select another source from a list offered by our system.

All of the choices take advantage of semantic relationships, already existing and accessible by our system, associated to the terms appearing in the original query. Term semantic relationships include but are not necessarily limited to synonymy, hyponymy, and hyperonymy (for instance, consider meronymy). Those relationships may be taken from repositories such as VoID[2] linksets or tools like WordNet[3] or can be described into the RDF datasets. Changing a term for a related one, derives in a change of semantics. The challenge is to be able to appropriately measure that change in order to assist the user when informing with the answers.

In a first attempt query rewriting can be approached term by term. Then, when all the terms of the original query are rewritten we say we have a complete translation (notice that it may also incurred in a semantic change). When there are terms in the original query without associated semantic relationships, we say we have a partial translation. A significant challenge is to manage how to cope with such a scenario. Different approaches are possible. For instance, try to find a translation for the union of its registered hyponyms, or try with the conjunction of its registered hyperonyms. In any case, measures for precision and recall for the query translation must be developed. Using such metrics a user is allowed to establish a threshold for the admitted loss in precision or loss in recall estimated for the received answers. For example, if the user defines a limit of 20% the system must guarantee that the amount of unwanted (loss in precision) or missed data (loss in recall) in the future answers presented to the user is kept always below 20% of the information showed. Moreover, the rewriting approach can be enhanced by allowing the rewriting of query expressions (instead of only single terms).

4 Methodology and Schedule

Our research can be scheduled into three phases.

In the *first phase* I have analyzed related works in the field of SPARQL query engines as well as works that consider query approaches on the database area, taking into account the query rewriting and relaxation techniques. Once I identified their contributions and weaknesses I defined a global architecture of my proposal with an specification of the functionalities of the modules that constitute that architecture. In the *second phase* I am concentrating my efforts on providing an innovative solution for the following two aspects:

- **Rewriting Process**. I am developing an algorithm that tries to rewrite the query in order to get a complete translation of it, and if that is not possible in order to get a partial translation. Different strategies are possible to search for translations.

[2] VoID - (`http://www.w3.org/TR/void/`)
[3] WordNet - (`http://wordnet.princeton.edu/`)

– **Definition of Metrics** that allows to estimate the information loss when semantic equivalence of the original query is not preserved.On this stage, we review the different metrics from Information Retrieval and their literature. Later we adapt the selected metrics to our approach.

In the *third phase* implementations for all those processes will be deployed and proper experimentation will be performed to test the approach.

5 Conclusion

In this paper we propose an approach for the federated query processing of heterogeneous Linked Data sources, based on the rewriting of the initial user query into new queries formulated in terms of the target data sources. To perform this task we are developing a new translation algorithm that uses ontology mapping and query rewriting techniques. Our final aim is to enrich the answer in an incremental manner with data obtained by querying each time to a different datasource, measuring the possible loss of information if semantic changes are detected in the reformulated query.

Acknowledgements. This work is supported by the TIN2010-21387-CO2-01 project.

References

1. Euzenat, J.: Semantic precision and recall for ontology alignment evaluation. In: Proc. 20th International Joint Conference on Artificial Intelligence (IJCAI), pp. 348–353 (2007)
2. Herzig, D., Tran, T.: One query to bind them all. In: COLD 2011, CEUR Workshop Proceedings, vol. 782 (2011)
3. Hurtado, C.A., Poulovassilis, A., Wood, P.T.: Query relaxation in rdf. In: Spaccapietra, S. (ed.) Journal on Data Semantics X. LNCS, vol. 4900, pp. 31–61. Springer, Heidelberg (2008)
4. Maedche, A., Staab, S.: Measuring similarity between ontologies. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS (LNAI), vol. 2473, pp. 251–263. Springer, Heidelberg (2002)
5. Makris, K., Gioldasis, N., Bikakis, N., Christodoulakis, S.: Ontology mapping and sparql rewriting for querying federated rdf data sources. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM 2010, Part II. LNCS, vol. 6427, pp. 1108–1117. Springer, Heidelberg (2010)
6. Quilitz, B., Leser, U.: Querying distributed rdf data sources with sparql. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 524–538. Springer, Heidelberg (2008)
7. Salton, G.: Automatic Text Processing: The Transformation, Analysis, and Retrieval of. Addison-Wesley (1989)
8. Schwarte, A., Haase, P., Hose, K., Schenkel, R., Schmidt, M.: Fedx: Optimization techniques for federated query processing on linked data. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 601–616. Springer, Heidelberg (2011)
9. Van Rijsbergen, C.J.: Information Retrieval. Evaluation, `http://www.dcs.gla.ac.uk/Keith/Chapter.7/Ch.7.html`

Knowledge Point-Based Approach to Interlink Open Education Resources

Xinglong Ma

Design, Engineering & Computing, Bournemouth University, Poole, UK
xma@bournemouth.ac.uk

Abstract. With more and more Open Education Resources (OER) courses being recognised and acknowledged by global learners, an emerging issue is that learners' self-efficacy is often affected by the lack of interaction between peers and instructors in their continuous self-learning process. This paper proposes a low-level Knowledge Point-based approach to serve application layers to enhance the interaction during the self-learning. This is achieved through taking advantage of Semantic Web and Linked Data techniques to annotate and interlink OER fragments which can later be reused and interoperated more conveniently.

Keywords: Linked Data, Open Education Resources, Annotation, Knowledge Point, Media Fragment, Self-Learning.

1 Introduction

An increasing number of universities and organisations are now participating in carrying forward the development of OER since MIT launched the OpenCourseWare (OCW[1]) initiative in 2001. With the multimedia based OER information (video, audio, digitalised textbooks and documents, etc.), global learner can freely access and schedule their self-learning. However, the process becomes more monotonous and unexciting by missing traditional interactive classroom. This often results that most learners are struggling to catch up with the whole curriculum and complete the course. To address this issue, a Knowledge Point-based approach is proposed to semantically annotate and interlink OER fragments rather than a collection of OER material, which provides the support of flexible reuse and interoperation of OER to serve learning applications.

This paper is organised as follows. First, the research problem is described. Second, current Linked Data and annotation technologies for OER are discussed. Third, the proposed Knowledge Point approach is introduced. Last, the schedule and related methodologies are briefed.

[1] http://ocw.mit.edu/index.htm

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 717–721, 2013.

2 Motivation and Research Questions

Without giving timely feedback and assessment, online learners often feel less motivated compared to learners in the traditional classroom learning [1, 2]. In terms of OER, massive open online courses (MOOCs), such as Khan Academy[2]and Coursera[3], are trying to improve this situation by providing more interactive environment with quizzes during or after class. However, these quizzes, which are usually predefined and arbitrary, cannot be easily reused and interoperated with open access. In most situations, once an OER material is published, no on-going supplements are maintained and served although it is essential for self-learning.

To date, most OER data are collected in distributed repositories, such as OCW, OER Commons[4], Merlot[5], where data are annotated by different metadata mechanisms (e.g. IEEE LOM [6], ADL SCORM [7]) and retrieved by individual web APIs/services [3].

This PhD project will research on how to reuse and interoperate isolated OER and in which way these OER can be more openly and flexibly accessed to promote interaction in self-learning.

3 State of the Art

Semantic web and Linked Data technologies have recently been exploited and applied into the field of the technology enhanced learning (TEL) to improve the learning performance and enable the reuse and interoperation of OER data. Following the Linked Data principles [4], URIs are used to name the OER data, which can be unambiguously identified. With the aid of URIs, the corresponding OER data and relevant interlinked data can be dereferenced, which are machine-readable and repurposed to serve the dynamic assessment system to enhance interaction with reused and interoperated OER data.

In [5-7], researchers presented Linked Data based approaches to automatically and dynamically generate learning assessments via DBpedia[8], publishing Wikipedia information on the web. However, these approaches do not reuse existing huge and diverse OER data. Instead, they highly rely on the knowledge from Wikipedia.

On the other hand, OER provided with Linked data facilitates the process of reuse and interoperation, which has been implemented by a few universities, such as Open

---

[2] https://www.khanacademy.org/

[3] https://www.coursera.org/

[4] http://www.oercommons.org/

[5] http://www.merlot.org/merlot/index.htm

[6] IEEE Learning Object Metadata (LOM) http://ltsc.ieee.org/wg12/

[7] Advanced Distributed Learning (ADL) Sharable Content Object Reference Model (SCORM) http://www.adlnet.gov/capabilities/scorm

[8] http://dbpedia.org/About

University[9], University of Southampton[10] and University of Oxford[11]. However, most OER data are still distributed in different OER repositories. Then a lot of researches have been done to integrate and generalise the metadata, such as OAI-PMH[12], which are registered by 1888 repositories [8], and ontology-based approaches[9-11]. However, these technologies are limited because the annotated data cannot be dereferenced [12] and these annotations are for a block or collection of OER rather than a single or certain part of OER material. Therefore more advanced technologies are required to enrich the OER data with Linked Data, such as textual analysis, text mining, information extraction and natural languages processing (NLP).

In another way, the annotation can be clinged to the certain section of an individual OER material to avert NLP kind of complex approaches, which can be implemented for videos and audios by using Media Fragments[13]. In [13], it applies the Media Fragments and NERD[14] to annotate the YouTube video fragments with Linked Data. Yet, for OER, there are more types of open data.

4 Proposed Approach

In this section, it proposes the concept of Knowledge Point (KP) and an annotation approach that refines annotation granularity and is based on the LOD Cloud to semantically annotate variety of OER materials.

In Fig. 1, the left-side shows the distributed and heterogeneous OER Repositories, which store multifarious of OER data, such as videos, documents, etc. After processing "Knowledge Point Annotation", the single material wrapped in a block or collection can be virtually "cut into multiple fragments". And a single fragment can be annotated by more than one KP which are acquired or extracted from manifold datasets of LOD Cloud[15]. Based on the KP Annotation mechanism, the RDF with Linked Data can be used in conjunction with higher-level educational applications.

Characteristics of KP include:

- Fully Compliant to best practice: As a member of web of data, KP complies with the Linked Data principles.
- Independent and atomy: KP can be operated independently and will not be affected by other KP.
- On-demand Fragmentation: An individual OER material can be fragmentised on the basis of KP and an OER fragment can be attributed by manifold KPs.
- KP and OER fragments intersupplement: KP annotates the OER fragment while the OER fragment explains the KP.

[9] http://data.open.ac.uk
[10] http://data.soton.ac.uk
[11] http://data.ox.ac.uk
[12] Open Archives Initiative – Protocol for Metadata Harvesting http://www.openarchives.org/pmh/
[13] Media Fragments URI 1.0 http://www.w3.org/TR/media-frags/
[14] Named Entity Recognition and Disambiguation http://nerd.eurecom.fr/
[15] Linked Open Data Cloud http://linkeddata.org/

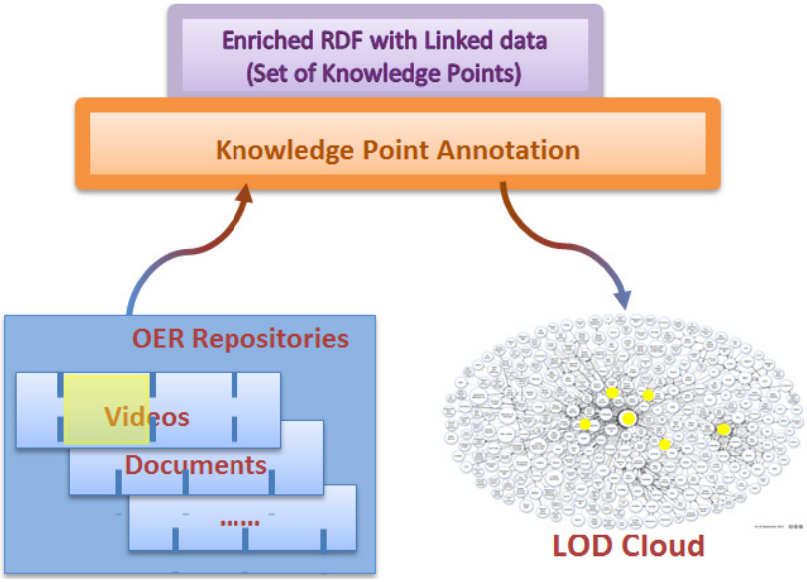


Fig. 1. Knowledge Point-Based Approach to Interlink OER (Left-side OER are virtually cut into multiple fragments. Left certain fragment with yellow can be annotated by the right-side yellow knowledge points distributed over LOD Cloud.)

There are a few challenges that could be seen in this proposed approach which also need to be address in the development. For example, accessing a single file over the distributed OER repositories and blocked file conveniently; compating all OER formats; extracting suitable KPs from LOD Cloud and fragmentising the OER data automatically and precisely.

5 Scheduled Methodology

- **January 2013 - January 2014:**Propose a lightweight mechanism to describe and annotate the heterogeneous OER in fragment level

— January – April: Review the currently lightweight annotation technologies (e.g. Media Fragment URI[16], Tagging, Folksonomy, etc.) and mainstream OER repositories tools and technologies (e.g. DSpace[17], Eprints[18], OAI-PMH, etc.).
— April - June: Find a possible approach on how to annotate the diverse OER materials distributed in different repositories which provide different access methods.
— June - October: Prototype this approach and test its efficiency and performance.
— September – January 2014: Evaluate this approach and improve it.

[16] http://www.w3.org/TR/media-frags/
[17] http://www.dspace.org/
[18] http://www.eprints.org/

- **February 2014 – July 2014:** Research on how to extract KPs from general lexical resources (e.g. DBpedia) and domain-specific ones (e.g. GeoNames[19]).
- **July 2014 - November 2014:** Using the proposed KPs extraction mechanism to extend the above prototype.
- **October 2014 - May2015:** Repeat testing, evaluating and improving to enhance this KP-based approach.
- **September 2015**: Finish and submit the PhD dissertation.

References

1. Rovai, A.P., Barnum, K.T.: On-line course effectiveness: An analysis of student interactions and perceptions of learning. The Journal of Distance Education/Revue de l'Éducation à Distance 18, 57–73 (2007)
2. Gikandi, J.W., Morrow, D., Davis, N.E.: Online formative assessment in higher education: A review of the literature. Comput. Educ. 57, 2333–2351 (2011)
3. Dietze, S., Sanchez, S., Ebner, H., Yu, H.Q., Giordano, D., Marenzi, I., Nunes, B.P.: Interlinking educational Resources and the Web of Data – a Survey of Challenges and Approaches. Electronic Library and Information Systems 47 (2013)
4. Berners-Lee: Design Issues, `http://www.w3.org/DesignIssues/LinkedData.html`
5. Foulonneau, M.: Generating educational assessment items from linked open data: The case of dBpedia. In: García-Castro, R., Fensel, D., Antoniou, G. (eds.) ESWC 2011. LNCS, vol. 7117, pp. 16–27. Springer, Heidelberg (2012)
6. Bratsas, C., Kontokostas, D., Eftychiadou, A., Kontokostas, D., Bamidis, P., Antoniou, I.: Semantic Web Game Based Learning: An I18n approach with Greek DBpedia. In: 2nd International Workshop on Learning and Education with the Web of Data, Lyon, France, April 17 (2012)
7. Waitelonis, J., Ludwig, N., Knuth, M., Sack, H.: WhoKnows? Evaluating linked data heuristics with a quiz that cleans up DBpedia. Interactive Technology and Smart Education 8, 236–248 (2011)
8. Open Archives Initiative: `http://www.openarchives.org/Register/BrowseSites`
9. Mikroyannidis, A., Lefrere, P., Scott, P.: An Architecture for Layering and Integration of Learning Ontologies, Applied to Personal Learning Environments and Cloud Learning Environments. In: 2010 IEEE 10th International Conference on Advanced Learning Technologies (ICALT), pp. 92–93 (2010)
10. Piedra, N., Chicaiza, J., Lopez, J., Martinez, O., Caro, E.T.: An approach for description of Open Educational Resources based on semantic technologies. In: 2010 IEEE Education Engineering (EDUCON), pp. 1111–1119 (2010)
11. Rocha Amorim, R., Rabelo, T., Amorim, D.: Open Educational Resources Ontology. In: II International Symposium on OER: Issues for Globalization and Localization (2012)
12. Haslhofer, B., Schandl, B.: The OAI2LOD Server: Exposing OAI-PMH metadata as linked data. In: Proceedings of WWW 2008 Workshop Linked Data on the Web (2008)
13. Li, Y., Rizzo, G., Troncy, R., Wald, M., Wills, G.: Creating enriched YouTube media fragments With NERD using timed-text. In: 11th International Semantic Web Conference, November 11-15 (2012)

[19] `http://www.geonames.org/`

A Linked Data Reasoner in the Cloud

Jules Chevalier

LT2C, Télécom Saint-Etienne, Université Jean Monnet,
10 rue Tréfilerie, F-4200 France
jules.chevalier@univ-st-etienne.fr

Abstract. Over the last decade, the paradigm of Linked Data has gained momentum. It is possible to leverage implicit knowledge from these data using a reasoner. Nevertheless, current methods for reasoning over linked data are well suited for small to medium datasets, and they fail at reaching the scale of the Web of Data. In this PhD thesis, we are interested in how distributed computing in the Cloud can help a linked data reasoner to scale. We present in this paper the early state of this thesis.

Keywords: Linked Data, Reasoning, Web of Data, Cloud Computing.

1 Research Questions

As Weiser predicted, computers have weaved themselves into the fabric of everyday life so that they are now indistinguishable from it. From personal computers to cars and televisions, all of these objects are now powerful computers generating more and more information. In many applications, the Semantic Web helps in changing this information into knowledge and linking it with other pieces of knowledge on the Web. Apart from their explicit knowledge, linked data contain implicit knowledge that can be leveraged using a reasoner. Reasoning is a complex process, and current solutions aim at reasoning at the scale of the Web of Data. That is why we need more powerful reasoners, scalable enough to make inference over very large datasets. So far, distributing and parallelizing this process over a cluster of computers seems the most adapted solution. Cloud Computing appears like an interesting environment for parallel inferencing. Elasticity is a primary characteristic of the Cloud, as it is composed of more or less heterogeneous clusters of commodity servers. Actually, Cloud providers APIs make it possible to scale up and down the number of dedicated Virtual Machines (VMs) that an application needs. A large-scale reasoner is an application presenting a profile that fits Cloud Computing. It would have computation bursts when new linked data arrive, depending on the amount of data and the number of new derived triples. Once triples are derived, they could be materialized, and then the reasoner no longer needs a large amount of VMs. The research question is therefore to propose a Cloud-ready linked data reasoner, whose architecture makes it possible to reason over a large scale corpus in a distributed way, and where scalability increases (resp. decreases) dynamically as the reasoning process is running (resp. no longer running). This research question also includes

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7882, pp. 722–726, 2013.

bandwidth optimization, which is part of the costs to run a service in the Cloud. In the following, we present previous work about distributed and parallelized reasoning and confront them to Cloud-hosted environments.

2 State of the Art

Until now, three works held our attention. These works are representatives of current solutions for distributed inferencing.

2.1 WebPie

In WebPie[8], the distribution is done thanks to the MapReduce paradigm[1], under the Hadoop framework. MapReduce is a very efficient paradigm for batch processing. In Webpie, each inference rule is a job. Jobs are executed one after each other, but this execution is distributed over a cluster. WebPie works with two logic fragments: RDFS[1] and OWL Horst[3]. The results show that quickly, over four cores, the gain of a new core is massively decreasing, against a logarithmic curve. [7] fixes some issues that optimises the reasoner implementation, improving its performance and completeness. But despite these upgrades, the results still suffer from the same issue. [4] critics this points in details. In short, while MapReduce is a handful paradigm which allows to set a distributed system implementing only two functions and that was popularized by its Hadoop-related eco-system, it is however not very adapted for reasoning. Actually, splitting the data in hermetic cores generates duplicates and therefore introduces unnecessary loops between jobs. This implies a higher bandwidth payload to exchange more batches of triples than necessary. Convergence is longer to reach, in a non linear way, as the number of triples increases. Although no theoretical evidence are provided, in practice a threshold close to four VMs limits the scalability.

2.2 MapResolve

[5] highlights the main drawback of the MapReduce paradigm : each *worker* must wait every other's end. This obviously slows down the computation speed, and decreases the project performance. Inspired by WebPie and other MapReduce works, they propose a reasoning solution over more expressive logic fragments. Despite their extensions, this follow-up proposal fails to provide significant improvements over WebPie in terms of performance.

2.3 Parallel Inferencing for OWL Knowledge Bases

For partitioning inference, we have two solutions: split the rules (that is what WebPie and MapResolve do), or split data. [6] proposes three methods to split data: graph partitioning, hash partitioning, or domain-specific partitioning, and

[1] A final recommendation from the W3C, `http://www.w3.org/TR/rdf-schema/`

a last technique to split rules depending on the rule-dependency graph. The authors also propose a parallelized reasoning algorithm based on existing reasoners. Unlike MapReduce-based solutions, data are not randomly splitted, with the aim to avoid duplicates and core communications. Unfortunately, these optimisations are not sufficient. Data are partitioned into hermetic cores, which still generates loops and duplicates.

2.4 Analysis

Among the three approaches we studied, approaches to build concurrent reasoners are divided into two categories :

Distributed : Both WebPie and MapResolve are based on MapReduce, which is a framework for distributed computing. Data partitioning in [6] is a distributed approach.

Parallel : In the case of rule-partitioning, [6] proposes a parallel approach.

In distributed computing each computing unit has its own private memory whereas in parallel computing all computing units access a shared memory. Due to the very own nature of the reasoning process, where rules can be interdependant, i.e. a directed graph, data cannot be splitted to be processed independently, which is a requirement of the MapReduce paradigm. To circumvent this issue, authors of the three approaches try to split data in order to minimize the overhead that will be implied by reprocessing data after a graph update (which occurs at each inference). This introduces loops and an overhead of bandwidth consumption that prevent scalability with more than half a dozen nodes. Surely the momentum gained by MapReduce for a few years, and the ease of implementation have oriented the authors towards this approach. The state of the art solutions do allow to handle more linked data than a single node could have done before. However, we believe that parallel processing could be an interesting paradigm to foster large scale linked data reasoning.

3 Proposed Approach

After studying existing solutions, we have initiated some features of our own solution for the case of a Cloud-hosted linked data reasoner.

1. Shared memory for a full parallel solution
2. Sort axioms by relevance instead of existing fragments
3. Stream compliant reasoner

3.1 Parallel Processing : Shared Memory

The main difficulty to design a parallel reasoner over the cloud is to efficiently implement a shared memory among numerous VMs. This problem, of the utmost practical interest, has been tackled by several approaches, especially for the Java language. Solutions such as Jelastic, Terracotta, HazelCast, Coherence present features that could be suitable to implement a concurrent reasoner.

3.2 Axioms Sorted by Relevance for the Web of Data

All axioms of description logic are not used with the same frequency. Figure 1 presents a rank of logic axioms as actually used in practise on the Web of Data. Using this histogram, instead of reasoning over defined fragments of description logic, we first reason over the most used axioms, to fit the use made by the Web of Data. This method favours the most-used concepts instead of grouping them by fragments.

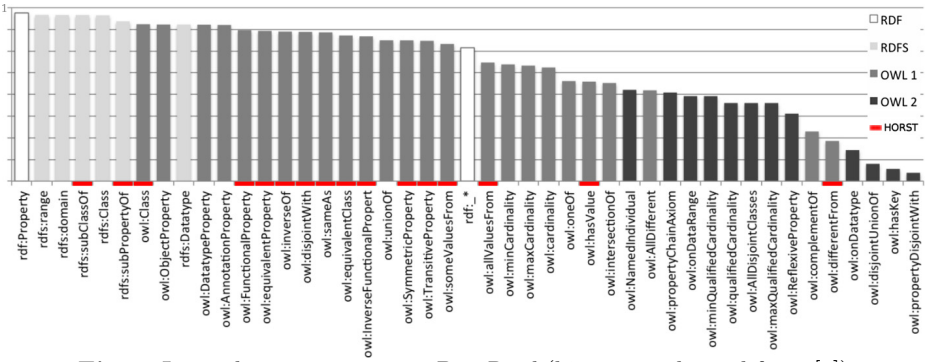


Fig. 1. Logic description axioms PageRank(histogram derived from [2])

A parameter **n** determines how many fragments are taken into account. We would be able to optimize **n** with respect to the time of reasoning that is acceptable for the application that requires the reasoning.

3.3 Stream-Based Architecture

Our last targeted feature is the ability to fire new triples to other nodes as soon as they are created by a rule. This prevents the use of the MapReduce paradigm to split the load. Instead of waiting the entire process of a MapReduce batch of data, each newly created axiom would be sent to all nodes that is expected to leverage new knowledge from this triple. This configuration tends to be as close as possible to pseudo real time reasoning. It could also be more compatible with incremental reasoning strategies of stream-oriented applications to be in semantic sensors networks. It implies to be able to connect the node in the architecture with respect to the dependency graph if rules in the considered linked data logic fragment. This depends on **n**.

4 Planned Research Methodology and Schedule

Deploy WebPie and Reproduce Results. The first point will be to deploy a WebPie version in our private Cloud. Thanks to this, we will be able to reproduce [8] experiments on different datasets, from the smallest to the biggest. This will let us have a baseline to compare our own experimentation results.

Propose and Implement Our Stream Reasoner. The second step will be the proposal of our reasoner. It is for now an archetype which needs to be completed. We must precise its core strategies (rule and data partition especially) and to implement it.

Compete against WebPie Results. When a first implementation will be finished, we would deploy it on our private Cloud, and run tests with the same datasets as for WebPie.

- **May 2013** - State of the art internal report.
- **October 2013** - WebPie deployment and tests
- **February 2014** - Proposal of our Cloud-hosted linked data reasoner.
- **May 2014** - First implementation of our reasoner.
- **November 2014** - 'Stable' version deployed on our private Cloud.
- **January 2015** - Evaluation campaign and interpretation of results.
- **April 2015** - Writing the PhD thesis.

Acknowledgement. Frédérique Laforest, Christophe Gravier and Julien Subercaze, in charge of this thesis.

OpenCloudware, funded by the French Fonds national pour la Société Numérique (FSN), and is supported by Pôles Minalogic, Systematic and SCS.

References

1. Dean, J.: MapReduce: Simplified data processing on large clusters. Communications of the ACM, 1–13 (2008)
2. Glimm, B., Hogan, A., Krötzsch, M., Polleres, A.: OWL: Yet to arrive on the Web of Data? CoRR, abs/1202.0 (2012)
3. Ter Horst, H.J.: Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary. Web Semantics: Science, Services and Agents on the World Wide Web 3(2-3), 79–115 (2005)
4. Patel-Schneider, P.: Comments on webpie. Web Semantics: Science, Services and Agents on the World Wide Web 15(3) (2012)
5. Schlicht, A.: Mapresolve. Web Reasoning and Rule Systems, 1–6 (2011)
6. Soma, R., Prasanna, V.K.: Parallel Inferencing for OWL Knowledge Bases. In: 2008 37th International Conference on Parallel Processing, pp. 75–82 (September 2008)
7. Urbani, J., Kotoulas, S., Maassen, J.: WebPIE: A Web-scale parallel inference engine using MapReduce. Web Semantics: Science, 59–75 (2012)
8. Urbani, J., Oren, E.: RDFS/OWL reasoning using the MapReduce framework. Science, 1–87 (2009)

Author Index