John Davies
Marko Grobelnik
Dunja Mladenic (Eds.)

Semantic Knowledge Management

Integrating Ontology Management, Knowledge Discovery, and Human Language Technologies

Springer

Semantic Knowledge Management

John Davies • Marko Grobelnik
Dunja Mladenic
Editors

Semantic Knowledge Management

Integrating Ontology Management, Knowledge Discovery, and Human Language Technologies

Springer

Editors

Dr. John Davies
BTexact Technologies
5/12 Orion
Ipswich, IP5 3RE
Adastral Park
United Kingdom
john.nj.davies@bt.com

Dr. Marko Grobelnik
Jožef Stefan Institute
Jamova 39
SI-1000 Ljubljana
Slovenia
Marko.Grobelnik@ijs.si

Dr. Dunja Mladenic
Jožef Stefan Institute
Jamova 39
SI-1000 Ljubljana
Slovenia
Dunja.Mladenic@ijs.si

ISBN: 978-3-540-88844-4 e-ISBN: 978-3-540-88845-1
DOI: 10.1007/978-3-540-88845-1

Library of Congress Control Number: 2008937504

ACM Classification: H.3, H.4, I.2

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

Contents

Contributors

Krasimir Angelov
Ontotext Lab, Sirma Group Corp., Bulgaria
krasimir.angelov@ontotext.com

David Baxter
Cycorp, Inc., Austin, TX 78731
baxter@cyc.com

V. Richard Benjamins
iSOCO – Intelligent Software Components, Madrid, Barcelona, València
rbenjamins@isoco.com

Kalina Bontcheva
Department of Computer Science, University of Sheffield, UK
K.Bontcheva@dcs.shef.ac.uk

Tom Bösser
Kea-pro GmbH, Tal, 6464 Spiringen, Switzerland
tb@kea-pro.net

Mercedes Blázquez
iSOCO – Intelligent Software Components, Madrid, Barcelona, València
mblazquez@isoco.com

Stephan Bloehdorn
Institute AIFB, University of Karlsruhe, D-76128 Karlsruhe, Germany
bloehdorn@aifb.uni-karlsruhe.de

Pompeu Casanovas
Institute of Law and Technology (IDT), Autonomous University of Barcelona
pompeu.casanovas@uab.es

Núria Casellas
Institute of Law and Technology (IDT), Autonomous University of Barcelona
nuria.casellas@uab.es

Jesús Contreras
iSOCO – Intelligent Software Components, Madrid, Barcelona, València
jcontreras@isoco.com

Brian Davis
Digital Enterprise Research Institute, Galway, Ireland
brian.davis@deri.org

John Davies
Next Generation Web group, BT Research, Ipswich, IP5 3RE, UK
john.nj.davies@bt.com

Alistair Duke
Next Generation Web group, BT Research, Ipswich, IP5 3RE, UK
alistair.duke@bt.com

Blaž Fortuna
Jožef Stefan Institute, 1000 Ljubljana, Slovenia
Blaz.Fortuna@ijs.si

Jasmin Franz
Empolis GmbH, 67657 Kaiserslautern, Germany
jasmin.franz@empolis.com

Adam Funk
Department of Computer Science, University of Sheffield, UK
a.funk@dcs.shef.ac.uk

Miha Grčar
Jožef Stefan Institute, 1000 Ljubljana, Slovenia
Miha.Grcar@ijs.si

Marko Grobelnik
Jožef Stefan Institute, 1000 Ljubljana, Slovenia
Marko.Grobelnik@ijs.si

Peter Haase
Institute AIFB, University of Karlsruhe, D-76128 Karlsruhe, Germany
haase@aifb.uni-karlsruhe.de

Frank van Harmelen
Department of Computer Science, Vrije Universiteit Amsterdam, the Netherlands
Frank.van.Harmelen@cs.vu.nl

Jörg Heizmann
Institut AIFB, Universität Karlsruhe (TH), D-76128 Karlsruhe, Germany
heizmann@aifb.uni-karlsruhe.de

Zhisheng Huang
Department of Computer Science, Vrije Universiteit Amsterdam, the Netherlands
huang@cs.vu.nl

Nicholas J. Kings
Next Generation Web group, BT Research, Ipswich, IP5 3RE, UK
nick.kings@bt.com

Atanas Kiryakov
Ontotext Lab, Sirma Group Corp., Bulgaria
atanas.kiryakov@ontotext.com

Ilian Kitchukov
Ontotext Lab, Sirma Group Corp., Bulgaria
ilian.kitchukov@ontotext.com

Bryan Klimt
Cycorp, Inc., Austin, TX 78731
bklimt@cyc.com; klimt@cmu.edu

Markus Krötzsch
Institut AIFB, Universität Karlsruhe (TH), D-76128 Karlsruhe, Germany
kroetzsch@aifb.uni-karlsruhe.de

Yaoyong Li
Department of Computer Science, University of Sheffield, UK
Y.Li@dcs.shef.ac.uk

Elke-Maria Melchior
Kea-pro GmbH, Tal, 6464 Spiringen, Switzerland
melchior@acit.net

Dunja Mladenić
Jožef Stefan Institute, 1000 Ljubljana, Slovenia
Dunja.Mladenic@ijs.si

Marta Poblet
Institute of Law and Technology (IDT), Autonomous University of Barcelona
marta.poblet@uab.es

Borislav Popov
Ontotext Lab, Sirma Group Corp., Bulgaria
borislav.popov@ontotext.com

David Schneider
Cycorp Inc., Austin, TX 78731
daves@cyc.com

Rudi Studer
Institute AIFB, University of Karlsruhe, D-76128 Karlsruhe, Germany
studer@aifb.uni-karlsruhe.de

York Sure
Institute AIFB, University of Karlsruhe, D-76128 Karlsruhe, Germany
sure@aifb.uni-karlsruhe.de

Ian Thurlow
Next Generation Web Research, BT Research, Ipswich IP5 3RE, UK
ian.thurlow@bt.com

Ralph Traphöner
Empolis GmbH, 67657 Kaiserslautern, Germany
ralph.traphoener@empolis.com

Joan-Josep Vallbé
Institute of Law and Technology (IDT), Autonomous University of Barcelona
pep.vallbe@uab.es

Johanna Völker
Institute AIFB, University of Karlsruhe, D-76128 Karlsruhe, Germany
voelker@aifb.uni-karlsruhe.de

Denny Vrandečić
Institut AIFB, Universität Karlsruhe (TH), D-76128 Karlsruhe, Germany
vrandecic@aifb.uni-karlsruhe.de

Ting Wang
Department of Computer Science and Technology, National University
of Defense Technology, Changsha, Hunan, China
tingwang@nudt.edu.cn

Paul Warren
Next Generation Web Research, BT Research, Ipswich IP5 3RE, UK
paul.w.warren@bt.com

Moritz Weiten
Ontoprise GmbH, D-76227 Karlsruhe, Germany
m.weiten@seeburger.de

Michael Witbrock
Cycorp Inc., Austin, TX 78731
witbrock@cycorp.eu

Chapter 1
Introduction to Semantic Knowledge Management

John Davies, Marko Grobelnik, and Dunja Mladenić

Despite its explosive growth over the last decade, the Web remains essentially a tool to allow humans to access information. The *Semantic Web* will extend the web's capability through the increased availability of *machine-processable* information.

Currently, Web-based information is based primarily on documents written in HTML, a language useful for describing the visual presentation of Web pages through a browser. HTML and today's Web, however, offer only very limited ways of describing the *content* itself. So, for example, you can specify that a given string should be displayed in a large bold font but you cannot specify that the string represents a product code or a product price.

As described in the following chapters, Semantic Web Technology aims to address this shortcoming using the descriptive languages RDF and OWL, and the data-centric, customizable markup language XML. These technologies, which are standards of the W3C[1] (WorldWideWeb Consortium), allow rich descriptions of the content of Web documents. These machine-processable descriptions in turn allow more intelligent software systems to be written, automating the analysis and exploitation of Web-based information.

This book presents a framework, methods, and tools for the integration of Ontology Management, Knowledge Discovery, and Human Language Technologies in the context of *Semantic Knowledge Management*, by which is meant the use of semantic technology for improved management of tangible knowledge assets. The book is in four parts but before describing its structure, we introduce the three key underlying technologies.

Ontology Management, as described in Chap. 2, focuses on ontologies as a key technology enabling semantic interoperability and integration of data, information, and processes. Ontology Management infrastructures are needed for the increasing development of semantic applications especially in the corporate semantic web, that is, the application of semantic technologies in an enterprise

J. Davies(✉)
Next Generation Web group, BT Research, Ipswich, IP5 3RE, UK.
e-mail: john.nj.davies@bt.com,

[1] http://www.w3c.org

J. Davies et al. (eds.), *Semantic Knowledge Management*,

environment. More specifically, the key requirements of such an infrastructure are identified as integration of heterogeneous knowledge sources, automatic content extraction and classification, dealing with inconsistent information, mediating between different ontologies, and support for structured queries against metadata and documents. An architecture supporting these requirements is presented and research and tools addressing each individual requirement discussed.

The main objective of *Knowledge Discovery*, as described in Chap. 3, is to automatically extract information from large volumes of data in a scalable way and, as we will see, thereby addresses a number of key tasks that arise in Semantic Knowledge Management. Knowledge Discovery can be defined as a process which aims at the extraction of interesting information from data in large databases. Knowledge discovery adopts the methods from Machine Learning and Data Mining which provide techniques for data analysis with various knowledge representations and large amounts of data, in addition to which methods developed in statistical learning and pattern recognition have a role to play.

Human Language Technology, as described in Chap. 4, involves the automatic analysis, mining, and production of natural language. Here we focus on information extraction as one of the important aspects of Human Language Technology when processing large amounts of text for Semantic Knowledge Management. Information extraction takes unseen texts as input and produces fixed-format, unambiguous data as output. It involves processing text to identify selected information, such as particular named entities or relations among them from text documents.

As mentioned above, the book is in four parts: Part I: Technology; Part II: Integration Aspects of Knowledge Management, Knowledge Discovery, *and* Human Language Technologies; Part III: Applications; Part IV: Case Studies.

Part I describes the three technologies outlined above: ontology management, knowledge discovery, and human language technologies. Part II addresses *integration* aspects of the three technologies via the OntoStudio environment for engineering ontologies and a platform supporting shared ontology knowledge management. Part III presents several *applications* ranging from semantically enhances search/browse and knowledge sharing through natural language generation and semantic document summarization to ontology generation from social networks, visualization of temporal spaces, and semantic media wiki. Finally, and most importantly, Part IV describes *case studies* where the presented methods and tools are applied on real-world data including telecommunication domain, legal domain, and evaluation of semantic applications.

Acknowledgment The work reported in this book is a result of the research project under ICT Programme of the European Community SEKT (IST-IP-2003-506826). The successful collaboration of partners in this project, named “Semantically Enabled Knowledge Technologies,” was enabled by the support of the Commission of European Communities. Further information is available at http://www.sekt-project.com/.

Chapter 2
Ontology Management

Stephan Bloehdorn, Peter Haase, Zhisheng Huang, York Sure, Johanna Völker, Frank van Harmelen, and Rudi Studer

Abstract Ontologies are considered a key technology enabling semantic interoperability and integration of data and processes. We are now entering a phase of knowledge system development, in which ontologies are produced in larger numbers and exhibit greater complexity than before. Ontology management infrastructures are needed for the increasing development of semantic applications especially in the corporate semantic web, which comprises the application of semantic technologies in an enterprise environment. This chapter presents a generic system architecture for an ontology management infrastructure that supports the construction of semantic applications. Such an ontology management infrastructure needs to be generic and flexible in its design, comprehensive in its functionality in order to meet the requirements of a wide spectrum of semantic applications. Key component functionalities of an ontology management system are discussed, including ontology mapping, ontology learning, debugging reasoning with inconsistent ontologies, and (structured and unstructured) query answering. An ontology management system architecture which can flexibly build upon existing datas sources and connect to other applications and which – as such – also forms the basis for a wide range of semantically enabled applications is described.

2.1 Introduction

Ontologies are considered a key technology enabling semantic interoperability and integration of data and processes. We are now entering a phase of knowledge system development, in which ontologies are produced in larger numbers and exhibit greater complexity. Ontology management infrastructures are needed for the increasing development of semantic applications especially in the corporate semantic web, which comprises the application of semantic technologies in an enterprise environment.

S. Bloehdorn (✉)
Institute AIFB, University of Karlsruhe, D-76128 Karlsruhe, Germany
e-mail: bloehdorn@aifb.uni-karlsruhe.de

J. Davies et al. (eds.), *Semantic Knowledge Management*,

In this chapter, we present a generic system architecture for an ontology management infrastructure that supports the construction of semantic applications. Such an ontology management infrastructure is expected to be generic and flexible in its design, comprehensive in its functionality in order to meet the requirements of a wide spectrum of semantic applications. To illustrate typical requirements of a semantic application, we start with a short scenario of semantic integration and search:

Bob works as technology analyst for an IT company. His daily work includes research on new technological trends, market developments and competitors of his clients. Bob's company maintains a digital library that gives access to a repository of internal surveys and analysis documents. The company also has a license with an academic research database which is accessed via a separate interface. Depending on his work context, Bob uses the topic hierarchies, the fulltext search functionalities or metadata search facilities provided by the two libraries to get access to the relevant data. However, Bob is often annoyed by the differing topic hierarchies and metadata schemes used by the two libraries as well as by a cumbersome syntax for posing metadata queries.

Questions which Bob might be interested in could include:

1. What articles were published by William Arms in "Communications of the ACM"?
2. Who wrote the book "Digital Libraries"?
3. What article deals with Grid Computing?
4. What are the topics of the paper "The future of web services"?
5. Which conference papers were classified as "emerging technology trends 2007"?
6. Which articles are about "Intellectual Capital"?

This scenario can be seen a prototypical scenario in which ontologies may be used for an integration of corporate information sources to enable semantic search for everyday research tasks. From this initial scenario and the above questions, we derive the following requirements that need to be met by an ontology management infrastructure:

Integration of Heterogeneous Knowledge Sources: In general, answering questions as the above might however require uniform access to different sources or repositories. However, a common limitation is that different knowledge sources use different taxonomies and different metadata schemes to describe the stored content, which requires a user to switch between different representations depending on the backend system searched. A particular challenge is the integration of structured knowledge sources (e.g., document metadata) and unstructured sources (e.g., document fulltexts). We would like to access heterogeneous knowledge sources via a single, unified interface that integrates different metadata schemes as well as different topic hierarchies. In Sect. 2.3 we illustrate how the integration of heterogeneous knowledge sources can be realized using ontology mappings.

Automatic Content Extraction and Classification: Questions 3 and 4 might require fine-grained access to the topics of articles. Hereby, with topics we mean items of some classification scheme to which documents are related. Though in many cases articles are classified, we can not expect that all relevant categories are actually explicitly stated. Thus, some support for automatically capturing the content of new documents added to the library seems necessary. Typically, the content of a knowledge source is not static, but changes over time: New documents come in, but also documents may be removed from the document base. We here require means to automatically extract relevant knowledge and to classify the new content. To address this requirement, we present in Sect. 2.4 techniques of incremental ontology learning.

Dealing with Inconsistent Information: In many cases, the information derived from diverse sources leads to inconsistencies. This is especially the case if information is derived using automatic knowledge acquisition tools such as wrappers, information extraction systems, or tools for automatic or semi-automatic ontology learning that aim at the semi- or even fully automatic extraction of ontologies from sources of textual data. Ontology-based applications which rely on learned ontologies have to face the challenge of reasoning with large amounts of inconsistent information resulting from automatic ontology generation systems. In Sect. 2.5 we introduce approaches that allow to process inconsistent ontologies.

Support for Structured Queries Against Metadata and Documents: With current interfaces to knowledge repositories, users either pose keyword-based queries on document fulltexts or abstracts or they use metadata search facilities to perform document retrieval. In general we are interested in receiving as answers facts from the knowledge base rather than only relevant documents.

We thus require the capability to pose structured queries to the underlying digital library which can be evaluated against the articles metadata as contained in the knowledge base. In general, we would like to allow the retrieval of facts besides the retrieval of documents. For metadata queries, current interfaces either offer preselected attribute fields (which are interpreted as conjunctive queries over a attribute-specific fulltext search) or they require some kind of formal query language.

In order to provide intuitive access and not to impose the burden on the user of learning a formal query language, we would like to allow the user to use an intuitive query language, ideally arbitrary natural language queries. In our running scenario, Bob should thus be able to ask the questions directly in the way they are stated in the scenario above. How such query answering can be realized, is discussed in Sect. 2.6.

Finally, we discuss Related Work in Sect. 2.7 and conclude with an outlook in Sect. 2.8.

2.2 Ontology Management Architecture

In the last decade, a number of competitive ontology management systems have been designed and implemented. Most naturally, many of these systems have a similar setup, i.e. they are built out of a similar set of basic components, each of which is responsible for a bundle of core functionalities. Despite these similarities, the systems also differ in many details such that ontology languages or query languages supported. In this chapter, we do not intend to present a fixed architecture, but rather an architecture that contains the most typical components of ontology management systems. In some way, the architecture is inspired by existing ontology management systems such as KAON2[1] or OntoBroker.[2]

Figure 2.1 illustrates a typical ontology management system architecture. In the following, we shortly summarize the main components. The core of the overall architecture is the ontology model module. The ontology model, typically an in-memory representation, maintains a set of references to the ontologies, their respective contents and corresponding instance data sources. From a logical perspective, it can be seen as a set of (uninterpreted) axioms. The ontology model can typically be modified by means of an API or via well-defined interfaces. Ontology editors and ontology learning systems usually access and modify the contents of the ontology model. As a separate component, the ontology metadata store maintains metadata information about the ontologies themselves, for example author, version and compatibility information.

While the ontology model (or parts of it) is usually kept in-memory during runtime, the underlying layer is responsible for persistence and storage functionalities. Often, these functionalities will be internal serialization and import/export. However, it may also include connectors to external data sources, e.g. classical relational databases that are interpreted in the context of the ontology model they are embedded in.

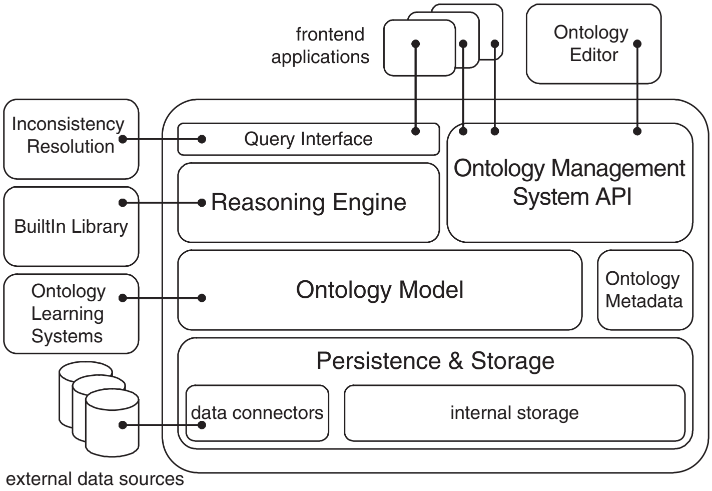


Fig. 2.1 Ontology management system architecture

[1] See http://kaon2.semanticweb.org/

[2] See http://www.ontoprise.de/content/e1171/e1231/index_eng.html

The reasoning engine operates on top of the ontology model, giving the set of axioms an interpretation and thus deducing new facts from the given primitives. Obviously, the behaviour of the reasoning engine will depend on the supported semantics, e.g. Description Logics in the case of OWL reasoning engines. The interaction with the reasoning engine from outside takes place either by means of a defined API or by means of a query interface. Again, the query interface needs to support a standardized query language, e.g. SPARQL for OWL/RDF data. The basic reasoning capabilities based on the ontology model can be further extended by means of external Builtins. Builtins can be seen as external, black-box type components that compute the extensions of certain properties externally. This mechanism increases the flexibility for knowledge representation by e.g. allowing complex arithmetic calculations, comparisons of arbitrary data types and so on. As a further component, the inconsistency resolution module can interact with the query engine and the reasoning engine to return meaningful answers to certain queries, even if the underlying knowledge base is inconsistent.

Based on this architecture, advanced front-end applications can be designed that interact with the ontology management system. As an example, consider the case of a natural language query interface that allows the translation of queries posed in everyday natural language into structured SPARQL queries that are in turn answered by the query interface.

2.3 Ontology Mapping

In large distributed information systems, the individual data sources are often described based on different, heterogeneous ontologies. To enable interoperability across these data sources, it is necessary to specify how the data residing at a particular node corresponds to data residing at another node. This is formally done using the notion of a mapping. There are mainly two lines of work connected to the problem of ontology mapping: (1) identifying correspondences between heterogeneous ontologies, (2) representing these correspondences in an appropriate mapping formalism and using the mappings for a given integration task. We here briefly discuss each of these individual aspects.

2.3.1 Mapping Discovery

The first task of identifying correspondences between heterogeneous ontologies is also referred to as mapping discovery or ontology matching. Today, there exist a number of tools that support the process of ontology matching in automated or semi-automated ways. A good survey of such ontology matchers is provided in (Shvaiko and Euzenat 2005). Instead of presenting individual tools, we here describe generic process for ontology matching, as first it was first presented for

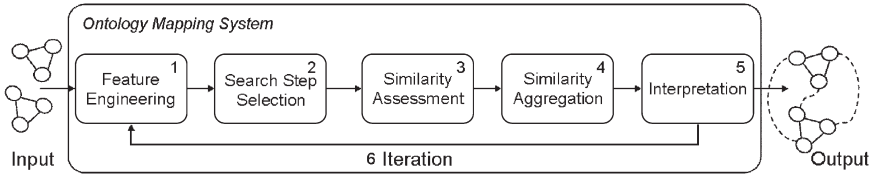


Fig. 2.2 Ontology matching process

the FOAM system in (Ehrig and Sure 2004). Other ontology matching approaches can be described in terms of this process as well. Figure 2.2 illustrates the six main steps of the generic process.

Input: Input for the process are two or more ontologies, which need to be matched with one another. Additionally, it is often possible to enter preknown (manual) matches. They can help to improve the search for matches.

1. **Feature Engineering:** The role of feature engineering is to select relevant features of the ontology to describe specific ontology entities, based on which the similarity with other entities will later be assessed. For instance, the matching process may only rely on a subset of OWL primitives, possibly only the taxonomy, or even just the linguistic descriptions of entities. For each feature, a specific matcher based on some similarity measure will be assigned.
2. **Search Step Selection:** The derivation of ontology matches takes place in a search space of candidate matches. This step may choose to compute the similarity of certain candidate element pairs and to ignore others in order to prune the search space.
3. **Similarity Computation:** For a given description of two entities from the candidate matches this step computes the similarity of the entities using the selected features and corresponding similarity measures.
4. **Similarity Aggregation:** In general, there may be several similarity values for a candidate pair of entities, e.g., one for the similarity of their labels and one for the similarity of their relationship to other entities. These different similarity values for one candidate pair have to be aggregated into a single aggregated similarity value. Often a weighted average is used for the aggregation of similarity values.
5. **Interpretation:** The interpretation finally uses individual or aggregated similarity values to derive matches between entities. The similarities need to be interpreted. Common mechanisms are to use thresholds or to combine structural and similarity criteria. In the end, a set of matches for the selected entity pairs is returned.
6. **Iteration:** The similarity of one entity pair influences the similarity of neighboring entity pairs, for example, if the instances are equal this affects the similarity of the concepts and vice versa. Therefore, the similarity is propagated through the ontologies by following the links in the ontology. Several algorithms perform an iteration over the whole process in order to bootstrap the amount of structural knowledge. In each iteration, the similarities of a candidate alignment are

recalculated based on the similarities of neighboring entity pairs. Iteration terminates when no new alignments are proposed.

Output: The output is a representation of matches and possibly with additional confidence values based on the similarity of the entities.

Current ontology matchers often represent mappings merely as a set of equivalences between simple ontology elements. While this is useful for elementary mappings, it is not sufficient for more advanced integration tasks involving ontology mappings, as we discuss in the following.

2.3.2 Mapping Representation

In contrast to the area of ontology languages where the Web Ontology Language OWL has become a de facto standard for representing and using ontologies, there is no agreement yet on the nature and the right formalism for defining mappings between ontologies. We here briefly discuss general aspects of formalisms for mapping representation.

What do mappings define? We can see mappings as axioms that define a semantic relation between elements in different ontologies. Most common are the following kinds of semantic relations:

Equivalence ($\equiv$): Equivalence states that the connected elements represent the same aspect of the real world according to some equivalence criteria. A very strong form of equivalence is equality, if the connected elements represent exactly the same object.

Containment ($\subseteq$): Containment states that the element in one ontology represents a more specific aspect of the world than the element in the other ontology. Depending on which of the elements is more specific, the containment relation is defined in the one or in the other direction.

Overlap (o): Overlap states that the connected elements represent different aspects of the world, but have an overlap in some respect. In particular, it states that some objects described by the element in the one ontology may also be described by the connected element in the other ontology.

In some approaches, these relations are supplemented by their negative counterparts. The corresponding relations can be used to describe that two elements are *not* equivalent, *not* contained in each other or *not* overlapping (or disjoint) respectively.

What do mappings connect? Most mapping languages allow defining the correspondences of mappings between ontology elements of the same type, e.g. classes to classes, properties to properties, etc. These mappings are often referred to as simple, or one-to-one mappings. While this already covers many of the existing mapping approaches, there are a number of proposals for mapping languages that rely on the idea of view-based mappings and use semantic relations between queries to connect models, which leads to a considerably increased expressiveness.

For example, the approach of (Haase and Motik 2005) allows describing correspondences between conjunctive queries over the source and target ontologies. The expressiveness of conjunctive queries corresponds to that of the well-known select-project-join queries in relational databases. Two typical approaches to specify mappings are the *global-as-view* (GAV) approach, where elements of the target are described in terms of queries over source, and the *local-as-view* (LAV) approach, where elements of the source are described in terms of queries over target.

Possible applications such mappings are manifold, including for example data transformation and data exchange. However, most commonly mappings are applied for the task of *ontology-based information integration*, which addresses the problem of integrating a set of local information sources, each described using a local ontology. Queries are expressed against a global, integrated ontology, which provides a unified view of the local ontologies. For details about query answering algorithms in such integration systems, we refer the reader to (Haase and Motik 2005).

2.4 Incremental Ontology Learning

Being an important means of data interchange and integration ontologies have become a solid basis for an increasing number of knowledge-based applications. Depending on the particular requirements their size ranges from a few to a few thousands of classes, and reasoning applications demand for more and more complex axiomatizations. Since building ontologies therefore is a task which is often difficult and very time consuming, it is desirable to automatize the construction of ontologies as far as possible.

Ontology learning aims at the automatic generation of ontologies from unstructured or semi-structured documents by machine learning, or natural language processing techniques. Whereas the learned ontologies mostly consist of concepts, instances, as well as taxonomic and non-taxonomic relationships recently a few approaches towards learning more expressive ontologies have been developed (e.g., Völker et al. 2007a). Common to all of these approaches is the need for timely adaptation of ontologies over time. That is, as the underlying data sources and the knowledge represented by them changes the learned ontologies have to be modified to reflect the new knowledge.

An efficient way of such data-driven ontology evolution is often referred to as incremental ontology learning. Frameworks like, for example, Text2Onto (Cimiano and Völker 2005) are able to suggest ontology updates based on changing lexical evidence and so-called document pointers, i.e. references linking individual ontology elements to fragments of text or other resources. Keeping document pointers allows for tracing results back to the data, but also enables applications to access parts of the document content by means of the ontology. Since a learned ontology can be considered as some kind of abstract, formal model of the knowledge contained in the underlying documents ontology learning can be a suitable means to support question answering tasks involving formal queries to unstructured, mostly textual data.

Obviously, as soon as more expressive constructs, in particular any type of negation (e.g., disjointness, Völker et al. 2007b), or cardinality constraints are introduced into learned ontologies the adaptation of ontologies over time may lead to logical inconsistencies. Similar problems arise if multiple ontologies are generated from related resources, and have to be integrated into one overall ontology by ontology mapping or merging techniques. In order to address these issues, future environments for automatic or semi-automatic ontology engineering will require methodologies for integrating ontology learning and reasoning, particularly including techniques for debugging or consistent ontology evolution (Haase and Völker 2005).

2.5 Processing of Inconsistent Ontologies

Dealing with inconsistencies is an important issue in the Semantic Web because of its scalability, distribution, and multi-authorship. The classical entailment in logics is explosive: any formula is a logical consequence of a contradiction. Therefore, conclusions drawn from an inconsistent ontology by classical inference may be completely meaningless.

There are two main ways to deal with inconsistent ontologies. One is to diagnose and repair inconsistencies. Another approach is to apply a non-standard reasoning method to obtain meaningful answers in the presence of inconsistency. The former is called the approach of *debugging* inconsistent ontologies, whereas the latter is called the approach of reasoning with inconsistent ontologies. In this section, we will overview both approaches of processing inconsistent ontologies. In particular, we will examine them by introducing the work has been done in the SEKT project.[3]

2.5.1 Debugging Inconsistent Ontologies

2.5.1.1 Framework and Tool

Debugging inconsistent ontologies is to diagnose and repair them. There exist several different approaches for debugging inconsistent ontology. One is to explain inconsistency by pinpointing. That can be done by a top-down approach with axiom pinpointing, or by a bottom-up approach by the support of an external reasoner. Another one is model-based diagnosis approach, which uses the traditional diagnosis approaches in the AI community (Reiter 1987).

Schlobach and Cornet (2003) proposes a non-standard reasoning service for debugging inconsistent terminologies. The work is further conducted in the SEKT

[3] http://www.sekt-project.com/

project (Schlobach and Huang 2007). In Description Logic, a concept is unsatisfiable if its interpretation is an empty set. An ontology is incoherent if it contains an unsatisfiable concept. Axiom pinpointing means identifying debugging-relevant axioms, where an axiom is relevant if an incoherent ontology becomes coherent once the axiom is removed or if, at least, a particular, previously unsatisfiable concept turns satisfiable. In order to obtain the pinpoints, Schlobach and Cornet introduce the notions of MIPS and MUPS. MIPS are incoherent sets of terminological axioms which can, possibly, not be traced back to the unsatisfiability of a particular concept name. If we are interested in the causes for unsatisfiability of a particular concept we need a more fine-grained notion, called minimal unsatisfiability-preserving sub-TBoxes (MUPS) of a TBox and a particular concept. In general there are several of these sub-TBoxes and we select the minimal ones, i.e., those containing only axioms that are necessary to preserve unsatisfiability.

DION is a system that uses the bottom-up approach for debugging inconsistent. DION uses an informed bottom-up algorithm to calculate MUPS/MIPS with the support of an external DL reasoner. The main advantage of the bottom-up approach is that it can deal with any DL-based ontology supported by an external reasoner. DION is designed to be a simple API for a general debugger with inconsistent ontologies. It supports extended DIG requests from other ontology applications, including OWL and DIG.[4] This means that DION can be used as an interface of an inconsistent ontology debugger as it supports the functionality of the underlying reasoner by just passing requests on and provides reasoning functionalities if needed. Therefore, the implementation of DION will be independent of those particular applications or systems. The prototypes of DION are available at the DION website.[5]

2.5.1.2 Experiment and Example

In the previous section of this chapter, we described the efforts of the Text2Onto team in learning expressive ontologies. Often the potential of ontologies cannot be fully exploited, because information such as disjointness of concepts is not made explicit, and as a consequence, possible modeling errors remain undetected. There are different methods to estimate disjointness of two concepts, e.g. based on taxonomic overlap, semantic similarity, linguistic patterns etc., which can be automatically combined by a machine learning classifier. Given a gold standard, this classifier would be trained to predict whether an unseen pair of two concepts is disjoint or not. Obviously, the problem of this approach is that a gold standard is needed, i.e. an agreed-upon collection of pairs of disjoint and non-disjoint concepts. Several experiments to construct such a gold standard have been conducted.

[4] http://dl.kr.org/dig/

[5] http://wasp.cs.vu.nl/sekt/dion

In these experiments, a sub-ontology PROTON is selected. Each concept pair is randomly assigned to six different people – three from each of two groups: the first one consisting of ontology experts, the second is composed of undergraduate students without profound knowledge in ontological engineering. Each of the annotators was given between 385 and 406 pairs along with natural language descriptions of the classes whenever those were available. Furthermore, they computed the majority votes for all the mentioned above datasets by considering the individual taggings for each pair. Adding the created disjointness statements to PROTON results in inconsistent PROTON ontologies. For example, there are 24 unsatisfiable concepts in the PROTON ontology with the disjointness created by 50% votes by Students.

Some of the most important findings from these experiments are that disjointness seems to be very difficult to understand for humans, and that it can be captured surprisingly well by automatic means. More surprising is that the agreement among the experts was not much higher than the agreement among the students. Another finding of the experiments was that even disjointness axioms introduced by full agreement of all annotators can, if included in the ontology, lead to incoherence, i.e. logical contradictions. To investigate the cause of this contradiction we applied DION to debug the newly constructed ontologies.

For the unsatisfiable concepts we calculate the minimal subsets of the ontology (MUPS) in which the concept is still unsatisfiable. Basically, this means that these sub-ontologies still contain a logical contradiction. It is interesting to observe that 100% of experts and students agree with the following axioms, however, they are incoherent:

{Reservoir → Lake, Lake → WaterRegion, Reservior → HydrographicStructure, HydrographicStrure → Facility, Disjoint(WaterRegion, Facility)}

The conclusion Reservoir → WaterRegion follows from the first two axioms, whereas Reservoir → Facility follows from the third and the fourth axioms. However, it contradicts with the axiom that WaterRegion and Facility are disjoint. This case shows that inconsistency occurs much more easily than people expect.

From these MUPS we calculate MIPS. These MIPS contain at least one “error”, they are orthogonal to classical “diagnoses”. They are not unique. We observe that the ontology including PROTON and only disjointness axioms agreed by all annotators only contains three unsatisfiable concepts. The situation is much worse in the case when disjointness axioms are added to PROTON, which were agreed upon by three experts. In this case, an analysis using a debugging tool such as DION can even be more useful to determine the cause of a logical error. Moreover, confidence or relevance information acquired during the ontology learning process might help to select the axiom, which is most likely to be incorrect.

2.5.2 Reasoning with Inconsistent Ontologies

2.5.2.1 Framework and Tool

Reasoning with inconsistent ontologies is to simply avoid the inconsistency by applying a non-standard reasoning method to obtain meaningful answers.

This approach is more suitable for the setting in the web area. For example, in a typical Semantic Web setting, one would be importing ontologies from other sources, making it impossible to repair them, and the scale of the combined ontologies may be too large to make repair effective.

The general task of a system of reasoning with inconsistent ontologies is: given an inconsistent ontology, return *meaningful* answers to queries. Huang et al. (2005) develops a general framework of reasoning with inconsistent ontologies, in which relevance based selection functions are used to obtain meaningful answers, where meaningfulness means that the answer is supported by a selected consistent sub-ontology of the inconsistent ontology. The main idea of the framework is: given a selection function, which can be defined as either syntactic or semantic relevance, we select some consistent sub-theory from an inconsistent ontology. Then we apply standard reasoning on the selected sub-theory to find meaningful answers. If a satisfying answer cannot be found, the relevance degree of the selection function is made less restrictive thereby extending the consistent sub-theory for further reasoning.

A linear extension strategy is proposed for selecting consistent sub-ontologies. We have proved that a reasoner using a linear extension strategy is never over-determined, may be undetermined, is always sound, and is always meaningful (Huang et al. 2005). A reasoner using a linear extension strategy is useful to create meaningful and sound answers to queries. The disadvantage of the linear strategy is that it may lead to an inconsistency reasoner that is undetermined. There exist other strategies that can improve the linear extension approach, for example, by backtracking and heuristics evaluation.

Selection functions play the main role in the framework of reasoning with inconsistent ontologies. Such a selection function determines which consistent subsets of an inconsistent ontology should be considered in its reasoning process. The general framework is independent of the particular choice of selection function. The selection function can either be based on a syntactic approach, or based on semantic relevance like for example in computational linguistics as in Wordnet or based on semantic relevance which is measured by the co-occurrence of concepts in the Web, which information is provided by search engines like Google.

PION is a system of reasoning with inconsistent ontologies, which is developed based on this framework. PION provides various selection functions and extension strategies to deal with inconsistent ontologies. PION supports syntactic-relevance-based selection functions and Google-based semantic relevance selection functions. The prototype of PION is available at its website.[6]

2.5.2.2 Experiment and Example

Huang and van Harmelen (2006) reports several experiments of PION. The tests show that the syntactic relevance approach works well with real-world inconsistent

[6] http://wasp.cs.vu.nl/sekt/pion

ontologies. An explanation for this could be that the syntactic relevance approach mimics our intuition that real-world truth is generally preserved best by the argument with the shortest number of steps; and whatever process our intuitive reasoning uses, it is very likely that it would somehow privilege just these shortest path arguments.

For example, the syntactic relevance approach works very well for the penguin example in which *Birds* are specified as *FlyingAnimals* and *Penguins* are specified as *Birds* which cannot fly. In this example, the reasoning path from *Penguin* to *Not Fly* is shorter than that from *Penguin* to *Fly*.

However, it does not work very well on the MadCow example, in which *Cows* are specified as *Vegetarians* whereas *MadCows* are specified as *Cows* which eats brains of sheep. Under the syntactic relevance based selection function, the reasoner returns that the "accepted" answer to the query "is *MadCow* a *Vegetarian*?" This counter-intuitive answer results from the main feature of the syntactic relevance approach, because it always prefers a shorter relevance path when a conflict occurs. In the MadCow example, the path "mad cow-cow-vegetarian" is shorter than the path "mad cow-eat brain-eat bodypart-sheep are animals-eat animal-NOT vegetarian". Therefore, the syntactic relevance-based selection function finds a consistent sub-theory by simply ignoring the fact "sheep are animals". The problem results from the unbalanced specification between *Cow* and *MadCow*, in which *Cow* is directly specified as a vegetarian whereas there is no direct statement "a *MadCow* is not a *Vegetarian*".

The semantic relevance approach provides an alternative to solve the limitation of the syntactic relevance approach. Cilibrasi and Vitanyi (2007) introduce the notion of Normalized Google Distances (NGD) to measure the co-occurrence of two keywords over the Web. NGD is introduced in PION as a measurement of semantic relevance between two concepts for its selection functions. For example, in the MadCow ontology, the Google-based semantic distance between *MadCow* and *Sheep* is shorter than that between *MadCow* and *Grass*, because *NGD(MadCow, Sheep) = 0.621* and *NGD(MadCow, Grass) = 0.723*. Namely, eating brains of sheep is considered as a more distinctive feature of MadCow than eating grass, based on the semantic information over the Web. Therefore, it provides an alternative to avoid the counter-intuitive answer if *MadCow* is specified as a *Cow* which eats sheep whereas *Cow* is specified as an animal which eats grass by default in the MadCow ontology. Huang and van Harmelen (2006) report several experiments of PION with various selection functions. The tests show that the semantic relevance based selection function plays better than the syntactic relevance selection function in reasoning with inconsistent ontologies.

2.6 Query Answering

Query answering is an important tool in ontology management. We here consider queries are logical conjunctive queries against the ontology. Query answering amounts to a reasoning process over the knowledge sources according to the semantics of the underlying ontology language OWL. A query language of particular importance is SPARQL, which is currently being standardized by the W3C as the standard query language for the Semantic Web. SPARQL was originally designed as RDF query language. As every OWL ontology can be encoded in an RDF graph, SPARQL can serve – at least syntactically – as an OWL query language.

The SPARQL query language is based on matching graph patterns. The simplest graph pattern is the triple pattern. Triple patterns in queries additionally may contain variables in the subject, predicate or object positions. Combining triples gives a basic graph pattern, where an exact match to a graph is needed to fulfil a pattern.

Queries are composed of two main building blocks, the SELECT or CONSTRUCT clause and the WHERE clause. The SELECT or CONSTRUCT clause identifies the variables to appear in the query results, they thus correspond to the distinguished variables of a conjunctive query. The WHERE clause identifies the triple pattern to match. As an example, consider the following SPARQL query which asks for articles that are associated with the topic "Intellectual Capital"?:

```
SELECT ?x WHERE {
?x rdf:type Article.
?x hasSubject ?y.
?y rdfs:label ?z.
match(?z, "Intellectual Capital")
}
```

For the answering of queries, we follow a virtual integration approach that does not require a full integration of all knowledge sources: The actual data still resides in the individual knowledge sources, and the mapping between the ontology and the knowledge sources is only used at runtime to retrieve the facts that are actually relevant for answering a query. In a query, predicates can be used that require access to different knowledge sources. The evaluation of these predicates is automatically *pushed down* to the respective knowledge sources. For example, this could be a relational query in the case of a relational database, or to a fulltext index in the case of the fulltext match predicate *"match"* over the topic hierarchy contained in the above query whereby the latter is implemented as a so-called *built-in*.

2.6.1 Extending Ontologies with Built-ins

Ins many situations it is desirable to let the reasoning engine work with knowledge that can be deduced from the available knowledge by using external components without relying on the reasoning within the semantics of the underlying logic. Built-ins are an approach to address this requirement. Built-ins can be seen as special purpose predicates defined on individuals or datatypes within the ontology. In contrast to conventional predicates, the extension of these predicates, each of which is associated with a particular software component, is not maintained within the ontology model but is computed on-the-fly during query evaluation.

Consider the following examples:

- A predicate *"stringmatch"* defined on string data types which computes the matching between two strings according to some notion of "match" (e.g., whether the strings differ only in their capitalization).
- A predicate *"classify"* defined on a string and an individual that returns the result of a text classifier against a previously trained category with the corresponding model being stored within the second argument.
- A predicate *"similarity"* that returns the similarity between two argument individuals according to a specified similarity measure.[7]

2.6.2 Expressive Natural Language Interfaces

Certainly, conjunctive queries in a query language like SPARQL cannot expected to be expressed directly by an end user. To address this problem, different approaches have been proposed to automatically translate queries in formalisms conceptually more adequate for end users (e.g., keyword queries, or queries in natural language) into formal queries based on logical languages. For example, ORAKEL (Cimiano et al. 2007) is a natural language interface which translates natural language queries to structured queries formulated with respect to a given ontology. This translation relies essentially on a compositional semantic interpretation of the question guided by two lexica: a domain-specific and a domain-independent lexicon. As the name suggests, the domain-independent lexicon is a priori encoded into the system and captures the domain-independent meaning of prepositions, determiners, question pronouns etc., which typically remain constant across domains. The domain-specific lexicon needs to be created by a *lexicon engineer* which is assumed to create the lexicon in various cycles. The end users then are able to directly interact with the application via natural language questions, which are translated into SPARQL queries by the ORAKEL system. The underlying mechanism however is hidden from the users.

2.7 Discussion and Related Work

Ontology management technologies are naturally connected to a wide variety of other areas in the field of semantically enabled knowledge management technologies. In this section, we shortly sketch relevant connections to two prominent areas which have been exploited as part of the SEKT project: knowledge discovery and natural language processing.

[7] E.g. consider the KAON2 similarity project at http://ontoware.org/projects/kaon2similarity/

2.7.1 Relation to Knowledge Discovery

The relation between ontology management and knowledge discovery is obviously bi-directional. On the one hand, knowledge discovery techniques, in particular text mining, can be exploited to generate ontological structures. On the other hand, ontologies can support the knowledge discovery process. For the first case, a recent approach in the SEKT project has used techniques for the scalable discovery of topic ontologies from text documents based on Latent Semantic Indexing and KMeans clustering) (c.f., Fortuna et al. 2006a, 2006b). The resulting system, OntoGen, offers supports to the user during the construction process by suggesting topics and analyzing them in real time. Following a somewhat different paradigm, machine learning techniques can be used to process data contained in the ontology, as in the above example of a text classification built-in. From the other perspective, the structure of ontologies has been used to exploit the implicit similarity of the concept nodes in text mining tasks (c.f., Bloehdorn et al. 2006).

2.7.2 Relations to Natural Language Processing (NLP)

As ontologies formally reflect conceptualizations of domain knowledge, there is a natural relation to applications that deal with the processing of natural language, which can be seen as the most flexible, though informal knowledge representation paradigm. The connections are present in both directions. One the one hand, applications that deal with natural language can rely on ontological knowledge structures to process unstructured language input. The ontologies can contain valuable background information that can help to interpret the input, allow for sanity checks of the results. Further the ontologies can provide the formal framework to represent the results of the NLP analysis as for example in the case of information extraction (IE), where the extracted entities are typically annotated against an existing ontology. On the other hand, NLP techniques can enable new functionalities for the ontology management system, e.g. allow for natural language queries to the ontology and knowledge base as in the above example of the ORAKEL system. Most naturally, ontology learning techniques (Sect. 2.4), that deal with the transformation of textual input into formal knowledge structures rely on NLP techniques.

2.7.3 Conclusion and Outlook

We have presented the main building blocks of an ontology management system. Guided by the scenario of information integration in a digital library scenario, we have looked at the overall architecture, mapping, ontology learning, built-ins and query answering functionalities. Instantiations of the named components have been developed within the EU integrated project “Semantic Knowledge Technologies (SEKT)”. The guiding scenario has been prototypically implemented within one of the case studies in the SEKT project, i.e. the BT digital library – a detailed description of which is given in Chap. 14.

The results of SEKT and in particular the results on ontology management are further developed in various projects. The EU integrated project NeOn[8] aims at extending the existing infrastructure such that one is able to deal with networked ontologies, i.e. multiple, distributed but interconnected ontologies. Further issues addressed by NeOn include managing the dynamics and evolution of ontologies and providing support for collaborative development of networked ontologies. As another example, the EU integrated project XMedia[9] addresses the issue of knowledge management with ontology management systems in complex distributed environments. A particular focus of the project are effective and efficient paradigms for knowledge retrieval within ontologies which allow users to define and parameterize views on the available knowledge according to their needs. Further aspects of the project are techniques for knowledge fusion and uncertainty representation and handling within ontology management systems.

In summary, integrated ontology management systems provide a flexible means for integrating and querying diverse and heterogeneous knowledge sources. The ontology management system can flexibly built upon existing data sources and connect to other applications but – as such – also forms the basis for a wide range of semantically enabled applications.

Acknowledgement This work was supported by the IST Programme of the European Community under SEKT (IST-IP-2003-506826).

References

Bloehdorn S, Basili R, Cammisa M and Moschitti A (2006). Semantic Kernels for Text Classification Based on Topological Measures of Feature Similarity. Proceedings of the 6th IEEE International Conference on Data Mining (ICDM’06) (Hong Kong, December 18–22, 2006).

Cilibrasi R and Vitanyi P (2007). The Google similarity distance. IEEE/ACM Transactions on Knowledge and Data Engineering, 2007.

Cimiano P and Völker J (2005). Text2Onto – A Framework for Ontology Learning and Data-Driven Change Discovery. Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB), 2005.

Cimiano P, Haase P and Heizmann J (2007). Porting Natural Language Interfaces Between Domains: An Experimental User Study with the ORAKEL System. Proceedings of the 12th International Conference on Intelligent User Interfaces (Honolulu, Hawaii, USA, January 28 – 31, 2007). IUI’07. ACM Press, New York, 180–189.

Ehrig M and Sure Y (2004). Ontology Mapping – An Integrated Approach. Proceedings of the First European Semantic Web Symposium, Heraklion, Greece, 76–91.

[8] http://www.neon-project.org

[9] http://www.x-media-project.org

Fortuna B, Grobelnik M and Mladenić D (2006a). System for Semi-automatic Ontology Construction, Demo at ESWC 2006, June 11–14, 2006, Budva, Montenegro.

Fortuna B, Grobelnik M and Mladenić D (2006b). Background Knowledge for Ontology Construction, Poster at WWW 2006, May 23–26, 2006, Edinburgh, Scotland.

Haase P and Motik B (2005). A Mapping System for the Integration of OWL-DL Ontologies. Proceedings of the First International Workshop on Interoperability of Heterogeneous Information Systems (Bremen, Germany, November 4, 2005). IHIS'05. ACM Press, New York, 9–16.

Haase P and Völker J (2005). Ontology Learning and Reasoning – Dealing with Uncertainty and Inconsistency. Proceedings of the Workshop on Uncertainty Reasoning for the Semantic Web (URSW), 2005.

Huang Z and van Harmelen F (2006). Reasoning with Inconsistent Ontologies: Evaluation, SEKT Deliverable D3.4.2.

Huang Z, van Harmelen F and ten Teije A. (2005). Reasoning with Inconsistent Ontologies. Proceedings of IJCAI'05.

Reiter R (1987). A theory of diagnosis from first principles. Artificial Intelligence, 32(1):57–95.

Schlobach S and Cornet R (2003). Non-Standard Reasoning Services for the Debugging of Description Logic Terminologies. Proceedings of IJCAI'03, Morgan Kaufmann, San Fransisco.

Schlobach S and Huang Z (2007). Inconsistent Ontology Diagnosis and Repair, SEKT Deliverable D3.6.3.

Shvaiko P and Euzenat J (2005). A survey of schema-based matching approaches. Journal on Data Semantics IV, 4 (Lecture Notes in Computer Science, vol. 3730):146–171. Springer, Berlin Heidelberg.

Völker J, Hitzler P and Cimiano P (2007a). Acquisition of OWL DL Axioms from Lexical Resources. Proceedings of the 4th European Semantic Web Conference (ESWC'07) (Lecture Notes in Computer Science), vol. 4519. Springer, Berlin Heidelberg.

Völker J, Vrandečić D, Sure Y and Hotho A (2007b). Learning Disjointness. Proceedings of the 4th European Semantic Web Conference (ESWC) (Lecture Notes in Computer Science), vol. 4519. Springer, Berlin Heidelberg.

Chapter 3
Knowledge Discovery for Semantic Web

Dunja Mladenić, Marko Grobelnik, Blaž Fortuna, and Miha Grčar

Abstract Knowledge Discovery is traditionally used for analysis of large amounts of data and enables addressing a number of tasks that arise in Semantic Web and require scalable solutions. Additionally, Knowledge Discovery techniques have been successfully applied not only to structured data i.e. databases but also to semi-structured and unstructured data including text, graphs, images and video. Semantic Web technologies often call for dealing with text and sometimes also graphs or social networks.

This chapter describes research approaches that are adopting knowledge discovery techniques to address semantic Web and presents several publicly available tools that are implementing some of the described approaches.

3.1 Introduction

Knowledge Discovery is traditionally used for analysis of large amounts of data and enables addressing a number of tasks that arise in Semantic Web and require scalable solutions. Additionally, Knowledge Discovery techniques have been successfully applied not only to structured data, i.e., databases but also to semi-structured and unstructured data including text, graphs, images and video. Semantic Web technologies often call for dealing with text and sometimes also graphs or social networks.

This chapter describes research approaches that are adopting knowledge discovery techniques to address semantic Web and presents several publicly available tools that are implementing some of the described approaches.

D. Mladenić (✉)
Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia,
e-mail: Dunja.Mladenic@ijs.si

J. Davies et al. (eds.), *Semantic Knowledge Management*,

3.2 Research Approaches

Knowledge Discovery can be defined as a process which aims at the extraction of interesting (non-trivial, implicit, previously unknown and potentially useful) information from data in large databases (Fayyad et al. 1996). Knowledge discovery approaches adopt the methods developed in Machine Learning and Data Mining (Mitchell 1997, Witten and Frank 1999, Hand et al. 2001) which provides techniques for data analysis with varying knowledge representations and large amounts of data, and also methods developed in statistical learning (Hastie et al. 2001) and pattern recognition (Duda et al. 2000) contributing data analysis in general.

Semantic Web on the other hand can be seen as mainly dealing with integration of many, already existing ideas and technologies with the specific focus of upgrading the existing nature of web-based information systems to a more "semantic" oriented nature (Grobelnik and Mladenić 2005). In that view knowledge discovery offers approaches that can be adopted and combined to provide for semantic Web technologies. Several groups of problems addressed by semantic Web technologies can be supported by knowledge discovery approaches, as pointed out in Grobelnik and Mladenić (2005) and Tresp et al. (2008). These includes ontology construction and management (for ontology management in general see Chap. 2), incorporation of domain knowledge, handling data that changes over time, capturing semantics of multimodal and multilingual data, supporting human language technologies (for human language technologies in general see Chap. 4) in the context of semantic Web. The rest of this section describes research approaches that can be used for some representative groups of problems addressed by Semantic Web technologies.

3.2.1 Ontology Construction

Methodology for semi-automatic ontology construction as defined in Grobelnik and Mladenić (2006) is inspired by CRISP-DM methodology used in knowledge discovery process. It consists of the following interrelated phases:

1. domain understanding (what is the area we are dealing with?),
2. data understanding (what is the available data and its relation to semi-automatic ontology construction?),
3. task definition (based on the available data and its properties, define task(s) to be addressed),
4. ontology learning (semi-automated process),
5. ontology evaluation (estimate quality of the solutions),
6. refinement with human in the loop (perform any transformation needed to improve the ontology and return to any of the previous steps, as desired).

Some of the above phases heavily involve human but most of them can be supported by knowledge discovery approaches. We discuss each of the phases giving some ideas on research approaches that can or have been used to support it.

Domain understanding is mainly on human but knowledge discovery can support it in different ways. For instance, supporting human in manual searching for information on the domain via information retrieval (van Rijsbergen 1979) or enabling collection of relevant data via focused crawling (Chakrabarti 2002). The idea of the underlying research approaches is to enable efficient search over large amounts of textual data returning a ranked list of relevant documents fitting the user query in case of information retrieval or collect a large amount of data on a given topic in the case of focused crawling. Both can be used to learn more on the domain under consideration.

If there is already available data on the domain, either in the form of database or a collection of items (such as, documents or images) one can use data mining and multimedia mining approaches to obtain better understanding of the domain and data. The underlying idea is to find regularities in the data, detect outliers, etc. Moreover by using machine learning and statistical learning approaches one can model the data and by studying the automatically obtained model gain better understanding of the domain. A valuable approach to data understanding is data visualization (Fayyad et al. 2001). For instance, by applying document visualization it is possible to get an overview of the document collection content (Grobelnik and Mladenić 2002, Steinbach et al. 2000) or relationships between the named entities that appear in the documents (Grobelnik and Mladenić 2004).

Task definition is mainly on human to perform but is heavily supported by the previous two phases. When defining the tasks to be addressed in semantic Web setting one needs to understand the domain at least to some extent and be aware of the available data to set up realistic goals. For instance, if we have available an ontology and instances the task can be fitting the instances into the ontology (commonly referred to as ontology population) or to extend the ontology by new concepts and relations. If only instances are available, we can define a task as ontology learning from scratch.

Ontology learning as a semi-automatic process has been supported by knowledge discovery approaches in several ways. If the task is to extend an existing ontology, one can use document collections as proposed in Agirre et al. (2000) or learn relations between the ontology concepts from documents, as proposed in Cimiano et al. (2004), Maedche and Staab (2001) and Heyer et al. (2001). If the task is to learn an ontology from scratch, one can use unsupervised learning on document collection (Bisson et al. 2000, Reinberger et al. 2004, Fortuna et al. 2005a, Hotho et al. 2003). Section 3.3.2 describes an example tool for semi-automatic ontology construction based on knowledge discovery approaches. More on ontology learning from text can be found in Buitelaar et al. (2005). However, even though textual data is the most frequently used source of data when constructing ontologies, one can use also other kinds of data, such as, records of social communication (i.e., social networks), collections of images or video and even a combination of those (commonly referred to as multimodal data, see Sect. 3.2.4 for details).

Ontology evaluation is an important step in validating quality of the existing ontology and comparing different approaches. It involves human, but can be to great extent supported by knowledge discovery approaches. For instance, ontology population supported by knowledge discovery approach can be used for ontology grounding enabling the domain expert to check how the data fits into the constructed ontology (Mladenić and Grobelnik 2007) exposing potential anomalies/drawbacks in the ontology. Automatic evaluation of ontology using knowledge discovery approach as proposed in Brank et al. (2007a) enables evaluation of an ontology that includes instances of the ontology concepts. The approach is based on the golden standard paradigm and its main focus is to compare how well the given ontology resembles the golden standard in the arrangement of instances into concepts and the hierarchical arrangement of the concepts themselves. In that approach no assumptions are made regarding the representation of instances, only that it is possible to distinguish one instance from another and that the ontology is based on the same set of instances as the golden standard. This means that the approach can be used in multimodal setting, where instances of the ontology can be of different modalities (see Sect. 3.2.4 for details).

Refinement of the ontology with human in the loop assumes that we can go back to any of the previous phases and refine it. Knowledge discovery techniques can be used here to support human in identifying the needed refinements and make the process of performing the needed changes as simple as possible.

3.2.2 Domain Knowledge

Domain knowledge captured in different forms can be incorporated in semantic Web applications. For instance, to provide semantically enabled search and browse (for general setting see Chap. 7) one can use knowledge discovery approaches to take into account information about the user interests via constructing a user profile as described in Sect. 3.3.3 and past activities via analysis of frequent patterns (a typical knowledge discovery task). One can also take the data and present it to the user in the context of a specified domain using knowledge discovery approaches to map the data on the context. Notice that the data that the user is browsing as well as the context can be represented in different way. An approach to contextualizing ontology is proposed in Grobelnik et al. (2007), where the data is an ontology and the context is represented as a collection of ontologies.

3.2.3 Dynamic Data

Data that in semantic Web applications often have time component and the property of that data that it changes over time may be important for the user. Different knowledge discovery approaches address a problem of dynamic data, for instance offering automatic identification of concept drift in the data, visualization of data changes over time (see Chap. 12) or ontology construction from data stream (Grobelnik et al. 2006). Data changes over time can be connected to different applications, for instance, observing changes of research area over time, one may want to see how competences of individuals or organizations change over time (Jörg et al. 2006) or who their collaboration network changes over time. Having a large social network changing over time, such as LinkedIn or Facebook, one may want to monitor the changes using knowledge discovery methods as proposed in Ferlež et al. (2008). Moreover, not only the underlying data but also the data models often change through time. For instance ontologies change via a process largely done manually by human editors. Knowledge discovery approaches can be used to automatically predicting when structural changes will occur in a given ontology (Brank et al. 2007b).

3.2.4 Multimodal and Multilingual Data

Knowledge discovery approaches can be used on multimodal data consisting of different data types including databases, text, images, video, graphs. The general idea is to preprocess the data and represent it in a way appropriate for further analysis with knowledge discovery approaches. For instance, a social network of researchers capturing paper co-authorship can be transformed into a graph and visualized as such or, transformed further into sparse vector representation of a graph (Mladenić and Grobelnik 2005). Chapter 10 describes an example application of ontology generation from social network that is based on sparse vector representation. Software mining (Grčar et al. 2007).

3.3 Tools

Knowledge Discovery methods have been used by several tools that support semantic Web and knowledge management. We present several of them, all partially developed under the same umbrella of Semantically Enabled Knowledge Technologies and refined through their usage in several real-world applications (Mladenić and Grobelnik 2007).

3.3.1 OntoClassify

OntoClassify (Grobelnik and Mladenić 2005) is a tool for ontology population that is using knowledge discovery methods for efficient population of a large topic ontology. The underlying idea is to generate a model for each ontology concept enabling efficient estimation of a level to which a new instance is matching the concept. OntoClassify has been developed following the idea of automatically classifying Web pages into WWW directory as described in Mladenić (1998) and evaluated on Yahoo! Directory of Web pages.

OntoClassify was first used for populating WWW directories (e.g., DMoz Open directory) and annotating textual resources such as, digital library resources by the ontology of British Telecom digital library (Mladenić and Grobelnik 2007). OntoClassify has been recognized as a useful building block for advanced semantic Web applications and incorporated in several system including construction of ontology from data stream (Grobelnik et al 2006), semi-automatic ontology construction (Fortuna et al. 2006b) and contextualizing ontologies (Grobelnik et al 2008). It was used to model several large ontologies including AgroVoc and ASFA (relevant for the Food and Agricultural Organization of the UN), EuroVoc (EU legislation), Cyc (common-sense knowledge) and DMoz directory.

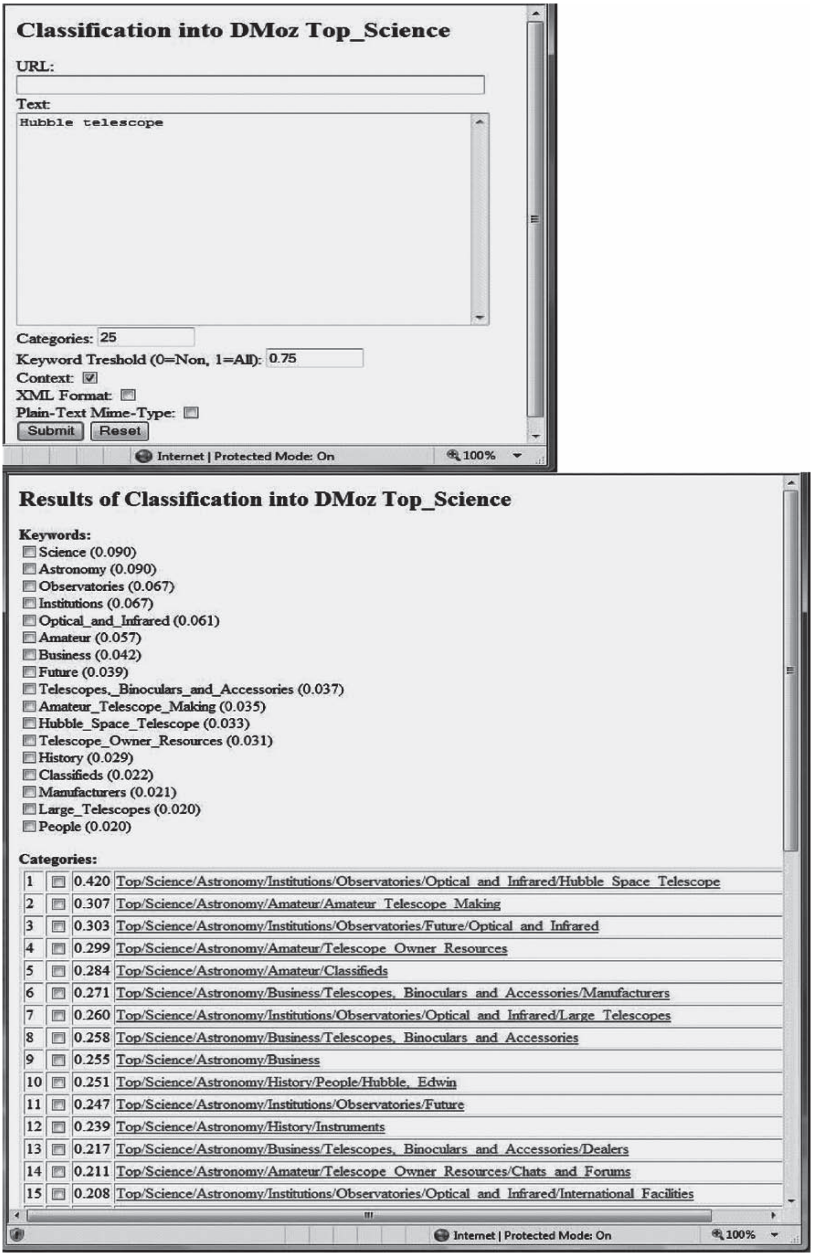
OntoClassify can be accessed as a Web application or as Web service <http://alchemist.ijs.si/ASP.NETv1.1/DMozClassify/OntoService.as mx> with a model generated for DMoz directory. Figure 3.1 shows a simple interface to OntoClassify as a Web application running on Science part of Open directory. From the results returned by the system we can see that "hubble telescope" can be annotated by the following keywords (in descending order of relevance): Science, Astronomy, Observatories, Institutions, Optical_and_Infrared, Amateur, Business, Future, Telescopes,_Binoculars_and_Accessories, Amateur_Telescope_Making, Hubble_Space_Telescope, etc. At the same time it can be assigned to the following categories (concepts from DMoz_Science).

OntoClassify enables entering relevant URL in addition to providing textual description of the instance to be annotated or assigned to an ontology. Figure 3.2 illustrates the results of providing URL of NASA Web page on the Hubble space telescope *http://hubble.nasa.gov/* in addition to providing the text "Hubble telescope." We can see that compared to the results in Fig. 3.1 the returned keywords and categories are slightly changed towards NASA concepts that are present in DMoz Open directory.

3.3.2 OntoGen

OntoGen (Fortuna et al. 2006b) is a semi-automatic and data-driven ontology editor focusing on editing of topic ontologies (a set of topics connected with different types of relations). The system combines text-mining techniques with an efficient user interface to reduce both: the time spent and complexity for the user (see Fig. 3.3 for illustration of the system interface). In this way it bridges the gap between complex ontology editing tools and the domain experts who are constructing the ontology and not necessarily having skills of ontology engineering. The two main characteristics of the system are that it is semi-automatic and data-driven.

By semi-automatic we refer to the interactive part of the system that aids the user during the ontology construction process. It suggests: concepts, relations between the concepts, names for the concepts, automatically assigns instances to



Classification into DMoz Top_Science

URL:

Text:

```
Hubble telescope
```

Categories: 25

Keyword Treshold (0=Non, 1=All): 0.75

Context: ☑

XML Format: □

Plain-Text Mime-Type: □

Results of Classification into DMoz Top_Science

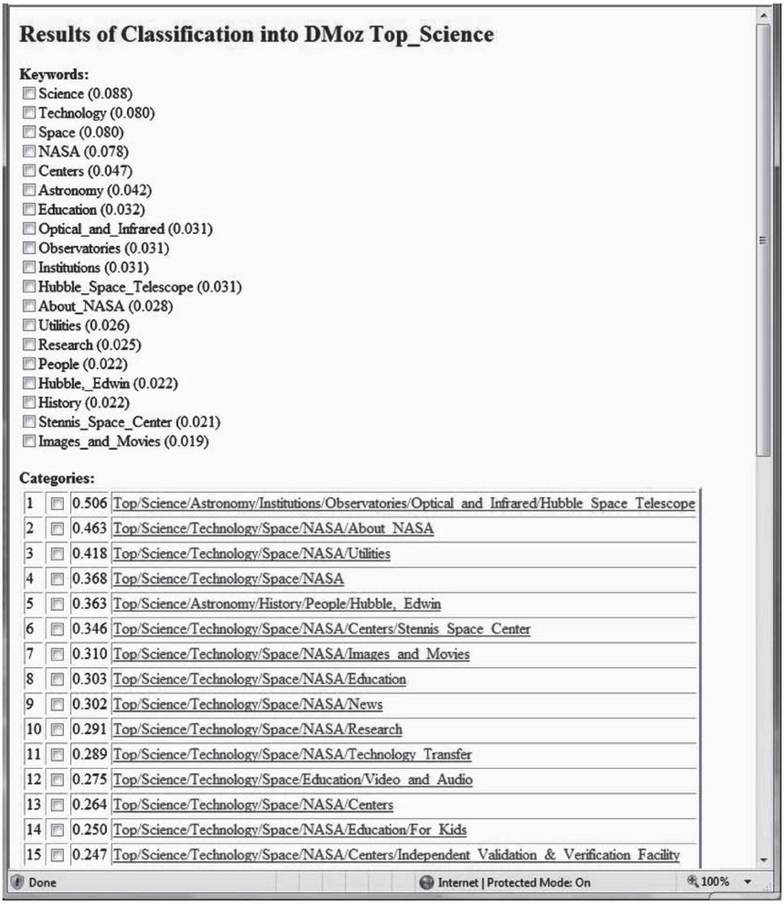
Keywords:

□ Science (0.090)
□ Astronomy (0.090)
□ Observatories (0.067)
□ Institutions (0.067)
□ Optical_and_Infrared (0.061)
□ Amateur (0.057)
□ Business (0.042)
□ Future (0.039)
□ Telescopes,_Binoculars_and_Accessories (0.037)
□ Amateur_Telescope_Making (0.035)
□ Hubble_Space_Telescope (0.033)
□ Telescope_Owner_Resources (0.031)
□ History (0.029)
□ Classifieds (0.022)
□ Manufacturers (0.021)
□ Large_Telescopes (0.020)
□ People (0.020)

Categories:

1	□	0.420	Top/Science/Astronomy/Institutions/Observatories/Optical and Infrared/Hubble Space Telescope
2	□	0.307	Top/Science/Astronomy/Amateur/Amateur Telescope Making
3	□	0.303	Top/Science/Astronomy/Institutions/Observatories/Future/Optical and Infrared
4	□	0.299	Top/Science/Astronomy/Amateur/Telescope Owner Resources
5	□	0.284	Top/Science/Astronomy/Amateur/Classifieds
6	□	0.271	Top/Science/Astronomy/Business/Telescopes, Binoculars and Accessories/Manufacturers
7	□	0.260	Top/Science/Astronomy/Institutions/Observatories/Optical and Infrared/Large Telescopes
8	□	0.258	Top/Science/Astronomy/Business/Telescopes, Binoculars and Accessories
9	□	0.255	Top/Science/Astronomy/Business
10	□	0.251	Top/Science/Astronomy/History/People/Hubble, Edwin
11	□	0.247	Top/Science/Astronomy/Institutions/Observatories/Future
12	□	0.239	Top/Science/Astronomy/History/Instruments
13	□	0.217	Top/Science/Astronomy/Business/Telescopes, Binoculars and Accessories/Dealers
14	□	0.211	Top/Science/Astronomy/Amateur/Telescope Owner Resources/Chats and Forums
15	□	0.208	Top/Science/Astronomy/Institutions/Observatories/Optical and Infrared/International Facilities

Fig. 3.1 OntoClassify running on Science part of Open directory (DMoz Top_Science). Interface over the Web where the user has entered “Hubble telescope” requesting the top 25 most relevant categories and the results of OntoClassify



Results of Classification into DMoz Top_Science

Keywords:

- □ Science (0.088)
- □ Technology (0.080)
- □ Space (0.080)
- □ NASA (0.078)
- □ Centers (0.047)
- □ Astronomy (0.042)
- □ Education (0.032)
- □ Optical_and_Infrared (0.031)
- □ Observatories (0.031)
- □ Institutions (0.031)
- □ Hubble_Space_Telescope (0.031)
- □ About_NASA (0.028)
- □ Utilities (0.026)
- □ Research (0.025)
- □ People (0.022)
- □ Hubble,_Edwin (0.022)
- □ History (0.022)
- □ Stennis_Space_Center (0.021)
- □ Images_and_Movies (0.019)

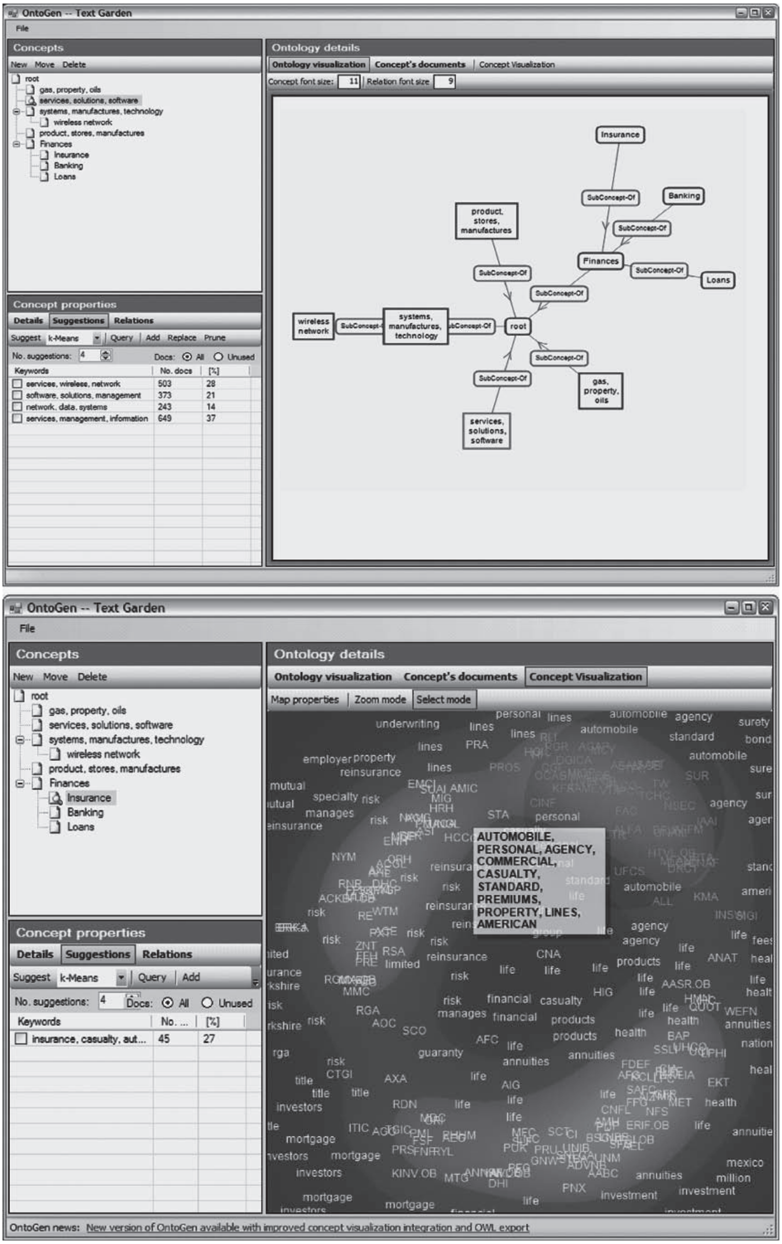
Categories:

1	□	0.506	Top/Science/Astronomy/Institutions/Observatories/Optical and Infrared/Hubble Space Telescope
2	□	0.463	Top/Science/Technology/Space/NASA/About NASA
3	□	0.418	Top/Science/Technology/Space/NASA/Utilities
4	□	0.368	Top/Science/Technology/Space/NASA
5	□	0.363	Top/Science/Astronomy/History/People/Hubble, Edwin
6	□	0.346	Top/Science/Technology/Space/NASA/Centers/Stennis Space Center
7	□	0.310	Top/Science/Technology/Space/NASA/Images and Movies
8	□	0.303	Top/Science/Technology/Space/NASA/Education
9	□	0.302	Top/Science/Technology/Space/NASA/News
10	□	0.291	Top/Science/Technology/Space/NASA/Research
11	□	0.289	Top/Science/Technology/Space/NASA/Technology Transfer
12	□	0.275	Top/Science/Technology/Space/Education/Video and Audio
13	□	0.264	Top/Science/Technology/Space/NASA/Centers
14	□	0.250	Top/Science/Technology/Space/NASA/Education/For Kids
15	□	0.247	Top/Science/Technology/Space/NASA/Centers/Independent Validation & Verification Facility

Fig. 3.2 Results of OntoClassify running on Science part of Open directory (DMoz Top_Science when the user has entered URL of NASA Web page on the Hubble space telescope and instance name "Hubble telescope"

the concepts and provides a good overview of the ontology to the user through concept browsing and various kind of visualization. At the same time the user is always full in control of the systems actions and can fully adjust all the properties of the ontology by accepting or rejecting the system's suggestions or manually adjusting them. This lets the user to establish a trust towards the system in a way that he has a full control over all the modifications to the edited ontology.

By data-driven we refer to the fact that most of the aid provided by the system is based on the underlying data provided by the user typically at the beginning of the ontology construction. The data reflects the structure of the domain for which



Concept font size: 11 | Relation font size 9

Suggest: k-Means

No. suggestions: 4 Docs: ● All ○ Unused

Keywords	No. docs	[%]
□ services, wireless, network	503	28
□ software, solutions, management	373	21
□ network, data, systems	243	14
□ services, management, information	649	37

Suggest: k-Means

No. suggestions: 4 Docs: ● All ○ Unused

Keywords	No. ...	[%]
□ insurance, casualty, aut...	45	27

OntoGen news: New version of OntoGen available with improved concept visualization integration and OWL export

Fig. 3.3 The main window of the system (upper) and the topic visualization (lower)

the user is building ontology. The system supports automatic extraction of instances (used for forming concepts) and co-occurrences of instances (used for relations between the concepts) from the data.

The tool combines several knowledge discovery approaches including handling textual data and representing it as vectors, automatic discovery of concepts, adding topics to the ontology, concept naming, incorporating domain knowledge, data visualization, addition of new instances. The rest of this section briefly describes each of the approaches.

For the *representation of documents* we use the well established bag-of-words document representation, where each document is encoded as a vector of term frequencies and the similarity of a pair of documents is calculated by the number and the weights of the words that these two documents share.

The central parts of OntoGen are the methods for *discovering concepts* from a collection of documents. OntoGen uses Latent Semantic Indexing (LSI) (Deerwester et al. 1990) and k-means clustering (Jain et al. 1999). LSI is a method for linear dimensionality reduction by learning an optimal sub-basis for approximating documents' bag-of-words vectors. The sub-basis vectors are treated as topics. k-means clustering is used to discover topics by clustering the documents' bag-of-words vectors into *k* clusters where each cluster is treated as a topic. The user interaction with the system is via a graphical user interface (GUI). When the user selects a topic, the system automatically suggests its potential subtopics. This is done by LSI or k-means algorithms applied only on the documents from the selected topic. The number of suggested topics is supervised by the user. User then selects the subtopics s/he finds reasonable and the system adds them to the ontology as subtopics of the selected topic.

Alternative way of *adding topics to the ontology* is by using Active Learning. In this case the user bootstraps the classification model, which classifies as positive the documents, belonging to the topics. The user starts the process by describing the topics using a query. Then the system selects a set of documents for which it is most unsure if they belong to the topic and asks the user for the correct answer. This is done until the user is satisfied with extracted topic. We use the SVM based active learning method originally proposed in Tong and Koller (2000).

Integrated into the system are two methods to help the user with understanding what the content of the extracted topics is and provide help at *naming* the topics. The first method, based on keyword extraction using centroid vectors, extracts descriptive keywords – most relevant keywords from the documents from the topic. The second method, keyword extraction using Support Vector Machine (SVM) (Joachims 1999), extracts distinctive keywords – the keywords that separate the topic from its close neighbors in the topic ontology.

The topic suggestion methods presented above heavily rely on the weights associated with the words – the higher the weight of a specific word the more probable that two documents are similar if they share this word. The weights of the words are commonly calculated by the so called TFIDF weighting (Salton 1991). In Fortuna et al. (2006a) we argue that this provides just one of the possible views on the data and propose an alternative word weighting that also takes into account the *domain knowledge* which provides the user's view on the documents. We integrated this method into the data loading functions of the system.

The system for *visualization* of large collections of documents presented in Fortuna et al. (2005) is tightly integrated with the OntoGen system. The user can visualize selected topic, by means of topic map. There is also feedback loop from the visualization, allowing the user to select a portion of a map (a topic) and adding it to the ontology as a subtopic of the visualized topic.

In order to support *addition of new instances* to the ontology (ontology population) we use the approach proposed in Grobelnik et al. (2006), but instead of using k-nearest neighbors classifier in each of the concepts we use the concept's SVM linear model for classification of new instances into the existing ontology. The system shows to the user all the concepts that the instance belongs to together with the level of certainty for instance belonging to the concept. Note that a new instance can be classified into more than one leaf concept.

3.3.3 SEKTbar

User profiling is an important part of the Semantic Web as it integrates the user into the concept of Web data with machine-readable semantics. In this chapter, user profiling is presented as a way of providing the user with his/her interest-focused browsing history. We present a system – SEKTbar – that is incorporated into the Internet Explorer and maintains a dynamic user profile in a form of automatically constructed topic ontology.

3.3.3.1 Architecture and Functioning of the System

The system provides a dynamic user profile in a form of topic ontology (similar to the one presented by Kim and Chan 2003). After a page is viewed, the textual content is extracted and stored as a text file. A collection of such text files (from now on simply termed *pages*) is maintained in two folders. The first folder holds *m* most recently viewed pages (*the short-term interest folder*). In our experiments, *m* is set to 5. The second folder contains the last *n* viewed pages, where *n > m (the long-term interest folder)*. In our experiments, *n* is set to 300. When a page is first visited, it is placed into both folders. Eventually it gets pushed out by other pages that are viewed afterwards. A page stays in the long-term interest folder much longer than in the short-term interest folder (hence the terms long- and short-term), the reason for this being a much higher number of new pages that need to be viewed for the page to be pushed out of the long-term interest folder.

To build a user profile, we first take the pages from the short-term interest folder and compute their TFIDF vector representations of the textual content, ignoring the order of words (thus such vectors are also termed bags of words), as introduced in Salton (1990). Each vector component is calculated as the product of Term Frequency (*TF*) – the number of times a word *W* occurs in the page – and Inverse Document Frequency (*IDF*). Prior to transforming pages into vectors, stop-words are removed and stemming is applied. After vectors are obtained, the centroid of short-term interest pages is computed by averaging corresponding TFIDF vectors component-by-component. This process combines the short-term interest pages, regardless of their count, into one single construct – the short-term interest centroid.

The long-term interest pages are treated slightly differently from the short-term interest pages. We first perform the bisecting k-means clustering (i.e., a variant of hierarchical clustering) over the long-term interest TFIDF vectors. This clustering method is computationally efficient and was already successfully applied on text documents (Steinbach et al. 2000). At start, all the pages form the root cluster which is first divided into two child clusters (hence the term bisecting clustering). The same procedure is repeated for each of the two newly obtained clusters and recursively further down the hierarchy. We perform the splitting until the size of the clusters (i.e., the number of pages the cluster contains) is smaller than the predefined minimum size (usually set to 10% of the initial collection size). During the clustering process, the similarity between two vectors is computed as the cosine of the angle between the two vectors.

The result of the clustering is a binary tree (in this text termed topic ontology), with a set of pages at each node. Later on, for each node a centroid is computed in the same way as for the short-term interest pages.

The root of the topic ontology holds the user's general interest while the leaves represent his/her specific interests. By our understanding the term general interest is not synonymous with *long-term interest* and in that same perspective the term *specific interest* is not a synonym for *short-term interest*. While the terms *long-term* and *short-term* (i.e., *recent* or *current*) *interest* emphasize the chronological order of page-views, this is not the case with the terms *general* (i.e., *global*) *interest* and *specific interest. General interest* stands for all the topics the user is – or ever was – interested in, while the term *specific interest* usually describes one more-or-less isolated topic that is – or ever was – of interest to the user.

By using the cosine similarity measure, we are able to compare the centroid at each node to the short-term interest centroid. In other words, we are able to map the user's current interest to the topic ontology. The mapping reveals the extent to which a node (i.e., a set of pages) is related to the user's short-term interest. By highlighting nodes with the intensity proportional to the similarity score, we can clearly expose the topic ontology segments that are (or are not, for that matter) of current interest to the user.

Due to the highlighting the user can clearly see which parts of the topic ontology are relevant to his/her current interest. He/she can also access previously visited pages by selecting a node in the hierarchy which is visualized in the application window. This can be explained as the user's interest-focused Web browsing history, the interest being defined by the selected node. The whole process is illustrated in Fig. 3.4.

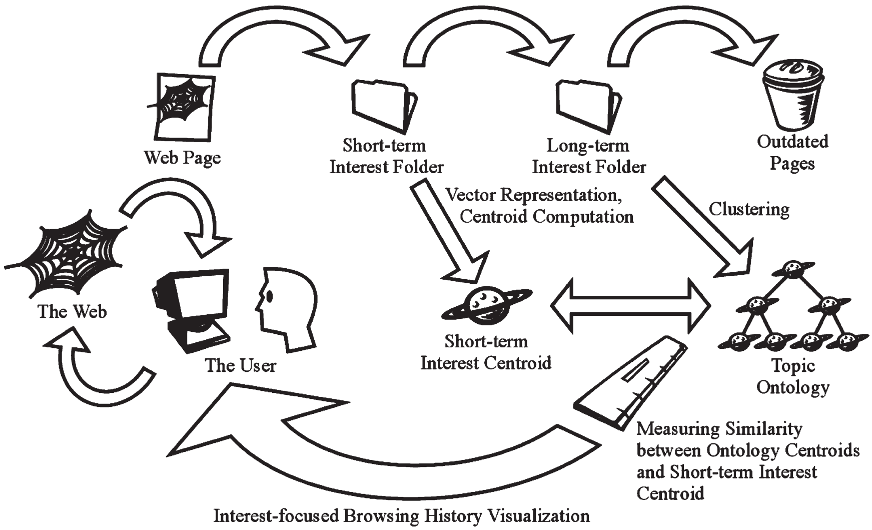

Web Page
Short-term Interest Folder
Long-term Interest Folder
Outdated Pages
Vector Representation, Centroid Computation
Clustering
The Web
The User
Short-term Interest Centroid
Topic Ontology
Measuring Similarity between Ontology Centroids and Short-term Interest Centroid
Interest-focused Browsing History Visualization

Fig. 3.4 The Interest-focused Browsing History Architecture

3.3.3.2 Implementation of the System

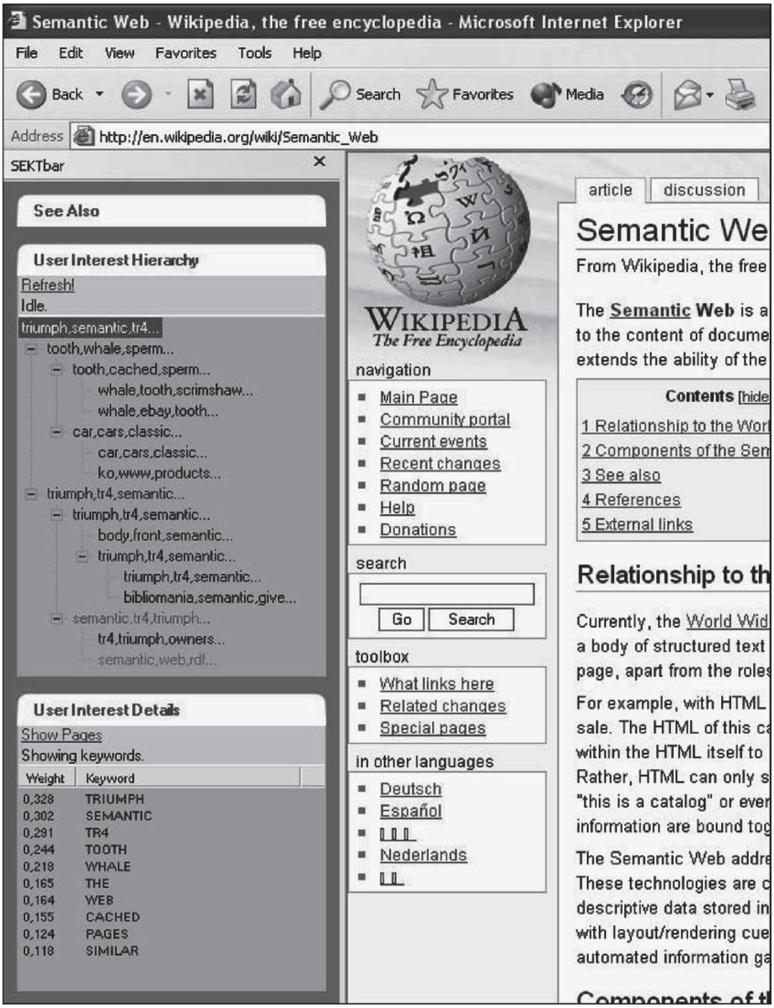
The user profile is visualized on the Internet Explorer toolbar that we developed for this purpose. The user can select a node (i.e., his/her more or less general interest) to see its specific keywords and the associated Web pages.

Generally, an Internet Explorer (IE) toolbar is an extension of the IE's GUI, as well as an application that extends the IE with additional functions. Since it is highly integrated into the IE, a toolbar can also:

1. receive notifications and information about the user's actions in the IE; most notably the user's requests to "navigate to" (the user's requests can be filtered or preprocessed in some other way);
2. access the contents of the currently loaded Web page;
3. apply any kind of changes to the content of the currently loaded page (e.g., highlight links to recommended pages, highlight some parts of the text, etc.);
4. easily access the Web as well as the local computer.

We have developed an IE toolbar to construct and visualize the user's interest-focused browsing history. The toolbar is placed into the left side of the IE's application window. It is divided into two panels, one showing the user's topic ontology and the other showing the most characteristic keywords and the set of pages corresponding to the selected node (see Fig. 3.5). The user can select any page from the list and navigate to that page.

The user's current interests are highlighted in the ontology visualization panel. The color intensity of the highlighting corresponds to the relevance of the node to



http://en.wikipedia.org/wiki/Semantic_Web

User Interest Hierarchy

Refresh!

Idle.

```
triumph,semantic,tr4...
  tooth,whale,sperm...
    tooth,cached,sperm...
      whale,tooth,scrimshaw...
      whale,ebay,tooth...
    car,cars,classic...
      car,cars,classic...
      ko,www,products...
  triumph,tr4,semantic...
    triumph,tr4,semantic...
      body,front,semantic...
      triumph,tr4,semantic...
        triumph,tr4,semantic...
        bibliomania,semantic,give...
    semantic,tr4,triumph...
      tr4,triumph,owners...
      semantic,web,rdf...
```

User Interest Details

Show Pages

Showing keywords.

Weight	Keyword
0,328	TRIUMPH
0,302	SEMANTIC
0,291	TR4
0,244	TOOTH
0,218	WHALE
0,165	THE
0,164	WEB
0,155	CACHED
0,124	PAGES
0,118	SIMILAR

Semantic We

From Wikipedia, the free

The **Semantic Web** is a
to the content of docume
extends the ability of the

Contents [hide

1 Relationship to the Worl
2 Components of the Sem
3 See also
4 References
5 External links

Relationship to th

Currently, the World Wid
a body of structured text
page, apart from the role

For example, with HTML
sale. The HTML of this c
within the HTML itself to
Rather, HTML can only s
"this is a catalog" or eve
information are bound to

The Semantic Web addr
These technologies are c
descriptive data stored in
with layout/rendering cue
automated information g

Components of t

Fig. 3.5 Screenshot of the system's GUI, captured after the user visited several Web pages on different topics. Screenshot shows the topic ontology of the user's interests and the most characteristic keywords from the root cluster. The user's most recent interest is highlighted with red color (the brighter the more relevant)

the user's current interest. The user can thus clearly see which pages that he/she already visited are in the context of his/her current interest.

Acknowledgments This work was supported by the Slovenian Research Agency and the IST Programme of the EC under SEKT (EU FP6 Project IST-IP-2003-506826) and NeOn (IST-2004-27595-IP).

References

Agirre E, Ansa O, Hovy E, Martínez D (2000) Enriching very large ontologies using the WWW. In: Proceedings of the First Workshop on Ontology Learning OL-2000. The 14th European Conference on Artificial Intelligence ECAI-2000. Berlin, Germany.

Bisson G, Nédellec C, Cañamero D (2000) Designing clustering methods for ontology building: The Mo'K workbench. In: Proceedings of the First Workshop on Ontology Learning OL-2000. The 14th European Conference on Artificial Intelligence ECAI-2000. Berlin, Germany.

Brank J, Grobelnik M, Mladenić D (2007a) Automatic evaluation of ontologies. In: Kao and Poteet (eds.) Natural Language Processing and Text Mining, Springer, London: 193–219.

Brank J, Grobelnik M, Mladenić D (2007b) Predicting the addition of new concepts in a topic hierarchy. In: Bohenec et al. (eds) Zbornik 10. mednarodne multikonference Informacijska družba IS 2007, 8.-12. oktober 2007: zvezek A: volume A, (Informacijska družba). Ljubljana: Institut "Jožef Stefan": 181–185.

Buitelaar P, Cimiano P, Magnini B (2005) Ontology Learning from Text: Methods, Applications and Evaluation, Frontiers in Artificial Intelligence and Applications, IOS Press, Amstredam

Chakrabarti S (2002) Mining the Web: Analysis of Hypertext and Semi Structured Data. Morgan Kaufmann, San Francisco.

Deerwester S, Dumais S, Furnas G, Landuer T and Harshman R (1990) Indexing by latent semantic analysis. Journal of the American Society of Information Science, Vol. 41/6: 391–407.

Duda R O, Hart P E and Stork D G (2000) Pattern Classification 2nd edition, Wiley, New York.

Fayyad U, Piatetski-Shapiro G, Smith P and Uthurusamy R (eds.) (1996) Advances in Knowledge Discovery and Data Mining. MIT Press, Cambridge, MA.

Fayyad U, Grinstein G G and Wierse A (eds.) (2001) Information Visualization in Data Mining and Knowledge Discovery, Morgan Kaufmann, San Francisco.

Ferlež J, Faloutsos C, Leskovec J, Mladenić D, Grobelnik M (2008) Monitoring network evolution using MDL. In: Proceedings of the IEEE 24th International Conference on Data Engineering ICDE: 1328–1330.

Fortuna B, Grobelnik M, Mladenić D (2005) Visualization of text document corpus. Informatica 29: 497–502.

Fortuna B, Grobelnik M, Mladenić D (2006a) Background Knowledge for Ontology Construction. WWW 2006, May 23.26, Edinburgh, Scotland.

Fortuna B, Grobelnik M, Mladenić D (2006b) Semi-automatic data-driven ontology construction system. In: Proceedings of the 9th International Multi-conference Information Society IS-2006, Ljubljana, Slovenia.

Grčar M, Grobelnik M, Mladenić D (2007) Using text mining and link analysis for software. In: Raś V, Zbigniew W, Zighed, Djamel A, Tsumoto, Shusaku (eds). ECML PKDD 2007, The 18th European Conference on Machine Learning and the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases, September 21, 2007, Warsaw, Poland. Proceedings of the Third International Workshop on Mining Comlex Data, MCD 2007: 1–12.

Grobelnik M, Mladenić D (2002) Efficient visualization of large text corpora. In: Proceedings of the Seventh TELRI Seminar. Dubrovnik, Croatia.

Grobelnik M, Mladenić D (2004) Visualization of news articles. Informatica Journal, Vol. 28, No. 4: 375–380.

Grobelnik, M, Mladenić, D (2005) Simple classification into large topic ontology of Web documents. Journal of Computing and Information Technology – CIT 13, 2005, 4, 279–285.

Grobelnik M, Mladenić D (2006) Knowledge discovery for ontology construction. In: John Davies, Studer Rudi, Warren Paul, (eds) Semantic Web Technologies. Wiley, Chichester: 9–27.

Grobelnik M, Brank J, Mladenić D, Novak B, Fortuna B (2006) Using DMoz for constructing ontology from data stream. In: Proceedings of the 28th International Conference on Information Technology Interfaces ITI 2006, 19–22 June, Cavtat, Croatia (IEEE Catalog, No. 06EX1244). Zagreb: University of Zagreb, SRCE University Computing Centre: 439–444.

Grobelnik M, Brank J, Fortuna B, Mozetič I (2008) Contextualizing ontologies with ontolight a pragmatic approach. Informatica, Vol. 32, No. 1: 79–84.

Hand DJ, Mannila H. Smyth P (2001) Principles of Data Mining (Adaptive Computation and Machine Learning), MIT Press, Cambridge, MA.

Hastie T, Tibshirani R and Friedman J H (2001) The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer Series in Statistics, Springer, New York.

Jain A K, Murty M N, and Flynn P J (1999) Data Clustering: A Review. ACM Computing Surveys, Vol 31/3: 264–323.

Joachims T (1999) Making large-scale svm learning practical. In: Scholkopf et al. (eds) Advances in Kernel Methods, Support Vector Learning, MIT, Cambridge, MA.

Jörg B, Jermol M, Uszkoreit H, Grobelnik M, Frlež J, Kiryakov A (2006) Analytical information services for the European research area. V: CUNNINGHAM, Paul (ed), CUNNINGHAM, Miriam (ed). Exploiting the knowledge economy: issues, applications and case studies, (Information and communication technologies and the knowledge economy, Vol. 3. IOS Press Amsterdam: 1367–1395.

Kim H R, Chan P K (2003) Learning implicit user interest hierarchy for context in personalization. In Proceedings of the 8th Iinternational Conference on Intelligent user Interfaces IUI, ACM Press, New York: 101–108.

Mitchell T M (1997) Machine Learning. McGraw-Hill.

Mladenić D (1998) Turning Yahoo into an automatic Web-page classifier. In: Proceedings of the13th European Conference on Artificial Intelligence ECAI 98. Wiley, Chichester: 473–474.

Mladenić D, Grobelnik M (2005) Visualizing very large graphs using clustering neighborhoods. In: Morik et al. (eds.) Local Pattern Detection: International Seminar: Dagstuhl Castle, Germany, April 12–16, 2004: revised selected papers, Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence) vol. 3539, State-of-the-art survey. Springer, Berlin Heidelberg 89–97.

Mladenić D, Grobelnik M (2007) Evaluation of semi-automatic ontology generation in real-world setting. In: Proceedings of the 29th International Conference on Information Technology Interfaces, June 25–28, 2007, Cavtat/Dubrovnik, Croatia. *ITI 2007* (IEEE Catalog, No. 07EX1589). University of Zagreb, Zagreb SRCE University Computing Centre: 547–551.

Salton G, Buckley C (1990) Improving Retrieval Performance by Relevance Feedback. Journal of the American Society for Information Science, Vol. 41, 4: 288–297.

Salton G (1991) Developments in automatic text retrieval. Science, Vol. 253: 974–979.

Steinbach M, Karypis G, Kumar V (2000) A comparison of document clustering techniques. In: Proceedings of KDD-2000 Workshop on Text Mining: 109–110.

Tong S, Koller D (2000) Support Vector machine active learning with applications to text classification. In: Proceedings of 17th International Conference on Machine Learning (ICML).

Tresp V, Bundschus M, Rettinger A, Huang Y (2008) Towards Machine Learning on the Semantic Web. In: d'Amato et al. (eds.) Uncertainty Reasoning for the Semantic Web I. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence) vol. 5327, Springer, Berlin Heidelberg.

van Rijsbergen, C J (1979) Information Retrieval. 2nd Edition. Butterworths, London.

Witten I H, Frank E (1999) Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann, San Francisco.

Chapter 4
Human Language Technologies

Kalina Bontcheva, Brian Davis, Adam Funk, Yaoyong Li, and Ting Wang

Abstract A tension exists between the increasingly rich semantic models in knowledge management systems and the continuing prevalence of human language materials in large organisations. The process of tying semantic models and natural language together is referred to as semantic annotation, which may also be characterized as the dynamic creation of bidirectional relationships between ontologies and unstructured and semi-structured documents.

Information extraction (IE) takes unseen texts as input and produces fixed-format, unambiguous data as output. It involves processing text to identify selected information, such as particular named entities or relations among them from text documents. Named entities include people, organizations, locations and so on, while relations typically include physical relations (located, near, part-whole, etc.), personal or social relations (business, family, etc.), and membership (employ-staff, member-of-group, etc.).

Ontology-based information extraction (OBIE) can be adapted specifically for semantic annotation tasks. An important difference between traditional IE and OBIE is the latter's closely coupled use of an ontology as one of the system's resources – the ontology serves not only as a schema or list of classifications in the output, but also as input data – its structure affects the training and *tagging* processes.

We present here two ontology-based developments for information extraction. OBIE experiments demonstrate clearly that the integration of ontologies as a knowledge source within HLT applications leads to improved performance. Another important finding is that computational efficiency of the underlying machine learning methods is especially important for HLT tasks, as the system may need to train hundreds of classifiers depending on the size of the ontology.

K. Bontcheva(✉)
Department of Computer Science, University of Sheffield, UK
e-mail: kalina@dcs.shef.ac.uk

J. Davies et al. (eds.), *Semantic Knowledge Management*,

4.1 Introduction

There is currently a large gap between Knowledge Management (KM) systems and the natural language materials that form 80% of corporate data stores (Perna and Spector 2004). Similarly, Gartner reported in 2002[1] that for at least the next decade more than 95% of human-to-computer information input will involve natural language, and that by 2012 taxonomic and hierarchical knowledge mapping and indexing will be prevalent in almost all information-rich applications. So a tension is present between the increasingly rich semantic models in KM systems and the continuing prevalence of human language materials. The process of tying semantic models and natural language together is referred to as semantic annotation, which may also be characterized as the dynamic creation of bidirectional relationships between ontologies and unstructured and semi-structured documents.

4.2 Research Approaches

Information extraction (IE) takes unseen texts as input and produces fixed-format, unambiguous data as output. It involves processing text to identify selected information, such as particular named entities or relations among them from text documents (Appelt 1999). Named entities include people, organizations, locations and so on, while relations typically include physical relations (located, near, part-whole, etc.), personal or social relations (business, family, etc.), and membership (employ-staff, member-of-group, etc.).

Ontology-based information extraction (OBIE) can be adapted specifically for semantic annotation tasks. An important difference between traditional IE and OBIE is the latter's closely coupled use of an ontology as one of the system's resources – the ontology serves not only as a schema or list of classifications in the output, but also as input data – its structure affects the training and tagging processes.

We present here two ontology-based developments for information extraction: annotation of entities with class labels using machine-learning algorithms modified specifically to take advantage of the hierarchical class structure; and extracting relations using machine-learning techniques on linguistic features identified in the text. The first tool takes advantage of the relations between concepts to apply hierarchical classification learning to semantic annotation; the second applies machine learning from linguistic features to the IE task of relation extraction, which it treats as a problem in identifying instances of ontological properties.

In addition to creating formal knowledge automatically from existing textual content via semantic annotation, human language technology can also

[1] http://www3.gartner.com/DisplayDocument?id = 379859

be used to assist people in the ontology authoring process, due to the fact that creating formal data is a high initial barrier for small organizations and individuals wishing to benefit from semantic knowledge technologies. Part of the solution comes from ontology authoring tools such as Protégé, but these often require specialist knowledge about ontology engineering. Therefore, in the case of naïve users, the definition of a controlled language for formal data description will enable them to author ontologies directly in natural language. Building on controlled language machine translation (MT) work, information extraction (IE) for controlled language analysis may achieve the high levels of accuracy necessary to make the ontology authoring process more widely feasible. For this task we have defined the Controlled Language for Ontology Editing (CLOnE) and implemented the Controlled Language Information Extraction (CLIE) tool which manipulates an ontology according to input sentences in CLOnE.

4.3 Tools

GATE (General Architecture for Text Engineering) (Cunningham et al. 2002) is an infrastructure for developing and deploying software components that process human language. It provides a set of NLP tools including a tokenizer, gazetteer, POS tagger, chunker, parsers, etc.; a system for developing and managing CREOLE (Collection of Reusable Objects for Language Engineering) plug-in components; and a graphical user interface for application development.

Because it is a widely used and recognized open-source toolkit suitable for reproducible experiments and a *de facto* standard for HLT applications, we chose GATE as the development platform for the OBIE (Ontology-Based Information Extraction) and CLIE (Controlled-Language Information Extraction) tools described next.

4.3.1 OBIE for Class Annotation

Conventional IE uses labels that have no specific relation among each other; the learning algorithms treat the labels (e.g., Person, Location) as completely independent of each other. However, because concepts in an ontology are related to each other (at the very least through the subsumption hierarchy), it is beneficial to feed this knowledge into the OBIE algorithms.

As concepts in ontologies are related to each other in a subsumption hierarchy, the cost (or loss) for an instance of a concept X being wrongly classified as belonging to another concept Y is defined as dependent on the relationship between the two concepts. (This cost is denoted $c(X,Y)$.) Consequently one requirement for OBIE is that if the system misclassifies a mention in the text as class Y instead of X, the cost of the misclassification should be as small as possible (e.g., where Y is a superclass of X).

IE systems are traditionally evaluated with the standard metrics – precision, recall and F-measure – computed independently for each category and averaged to obtain the overall performance. These do not, however, take into account the hierarchical structure of our classifications, so we generalize these measures to produce ontologically relevant ones, by defining a cost-based measure which takes into account distances in the ontology (Li and Bontcheva 2007).

Our OBIE learning algorithm is based on the Hieron large-margin learning algorithm for hierarchical classification (Dekel et al. 2004), which exploits the class label structure by learning a Perceptron model for each class while ensuring that the difference between two models is in proportion to the distance between the two classes in the tree. Thus when we misclassify an example as class X instead of Y, X should be close to Y in the hierarchy.

Our OBIE system learns from the training set with repeated learning loops until the improvement (reduction in error from the previous loop) is less than a specified parameter. (In contrast, the original Hieron algorithm allows only one learning loop.)

In order to address specifics of the OBIE task, we also introduce a regularization parameter, similar to that used in Perceptron (Li et al. 2002, 2005), which adds an extra dimension to the feature vector of each training example, so that the training examples become linearly separable even if they were not originally so; thus the Hieron algorithm will stop after a finite number of loops over any training set.

To implement this system, we adapt the OBIE task according to the following principles.

1. For each class in the ontology, two hierarchical classifiers are trained – one to recognize the start token and another for the end token of the class label.
2. The ontology is notionally extended with an additional class directly under the root node to represent irrelevant tokens. Since it is required only for the functioning of the algorithm, this class is not considered in the evaluation metrics.
3. The Hieron algorithm, which is designed for strictly tree-shaped hierarchies, is adapted to deal with graphs where some classes have more than one parent class. (In the Proton[2] ontology, for example, *SportBuilding* is a subclass of both *Building* and *SportFacility*.)

To evaluate OBIE, we used a corpus of almost 300 news articles (divided nearly equally among business, international politics and UK politics) that had been manually annotated according to the Proton ontology. (This corpus was composed internally in the SEKT project.) We pre-processed the corpus with the GATE ANNIE system to take advantage of domain-independent linguistic features (e.g., lemma, POS tag, named-entity tag) as well as the document text itself, and

[2] http://proton.semanticweb.org/

considered the features across a window of 9 tokens (including and centered on each token in question).

Since there are no previously reported results on the corpus we required, we compare our results with the modified Hieron algorithm against two state-of-the-art traditional learning algorithms, SVM and Perceptron, on the same corpus.

Table 4.1 presents the results of this evaluation on all three subcorpora. We note that Hieron generally outperforms the conventional (non-ontology-based) algorithms according to the conventional F-measure, and that it consistently achieves significantly higher performance according to the ontologically oriented F-measure. Further details of our evaluation experiments are given in Li and Bontcheva (2007).

4.3.2 OBIE for Relation Extraction

Until recently, research has focused primarily on the use of IE for populating ontologies with concept instances (Handschuh et al. 2002, Motta et al. 2002). In addition to this, however, many ontology-based applications require methods for the automatic discovery of properties (called relations in the corpus to which we will refer here) of instances. Semantic relations provide richer metadata connecting documents to ontologies and enable more sophisticated semantic search and knowledge access.

The Automatic Content Extraction (ACE) program[3] defines this task as Relation Detection and Characterization (RDC). RDC uses the results of named entity recognition, which detects and classifies entities according to a predefined entity type system. In contrast to previous evaluations, ACE2004 introduced a type and subtype hierarchy for both entities and relations – an important step towards ontology-based information extraction. We have chosen the ACE2004 entity and relation hierarchies as the basis for this work, in order to benefit from the class hierarchy as well as the opportunity for evaluation on the existing corpus.

Table 4.1 Comparison of the performance of Hieron, SVM and Perceptron learning on OBIE: micro-averaged F1 and distance-based Fo1

	F1			Fo1		
	PAUM	SVM	Hieron	PAUM	SVM	Hieron
Bus.	74.1	75.3	82.7	78.8	79.3	91.2
Int.77.1	80.1	83.3	83.0	85.9	91.3	
UK	82.0	82.9	82.5	83.6	84.4	90.1

[3] http://www.nist.gov/speech/tests/ace/

One of the main barriers to applying information extraction in ontology-based applications comes from the difficulty of adapting the algorithms to new domains. In order to overcome this problem, recent research has advocated the use of Machine Learning (ML) techniques for IE.

A number of ML approaches have been used for relation extraction, for example, Hidden Markov Models (HMM) (Freitag and McCallum 2000), Conditional Random Fields (CRF) (Lafferty et al. 2001) and Maximum Entropy Models (Kambhatla 2004). Among those, Support Vector Machines (SVM) (Vapnik 1998) have been particularly successful and tended to outperform other methods.

Zelenko et al. (2003) proposed extracting relations by computing kernel functions between shallow parse trees. Kernels have been defined over shallow parse representations of text and have been used in conjunction with SVM learning algorithms for extracting person-affiliation and organization-location relations. Culotta et al. (2004) extended this work to estimate kernel functions between augmented dependency trees.

Zhou et al. (2005) further introduced diverse lexical, syntactic and semantic knowledge in feature-based relation extraction using SVM. The feature system covers word, entity type, overlap, base phrase chunking, dependency tree and parse tree, together with relation-specific semantic resources, such as country name list, personal relative trigger word list. Their results show that the feature-based approach outperforms tree kernel-based approaches, achieving 55.5% F-measure in relation detection and classification on the ACE2003 training data. This is in fact state-of-the-art performance on this task, as it is a lot harder than other information extraction tasks, for example, recognizing names of people.

SVMs were however originally designed for binary classification, whereas relation extraction is reducible only to a multiclass problem. Following Hsu and Lin's (2002) study of SVMs and multiclass problems, we chose to construct one-against-one combinations of binary classifiers for our experiments.

Using natural language processing (NLP) to produce linguistic features for machine-learning has been shown to benefit IT (Li et al. 2005) so here we use a diverse set of NLP tools running in GATE to generate a set of 94 features, partly adopted from Zhou et al. (2005) but with the addition of POS tags, entity subtypes and classes, entity mention roles and several semantic features (Wang et al. 2005).

We evaluated our method, especially the contributions of the various features, on the ACE2004 training data, taking into consideration only explicit relations between pairs of entity mentions in the same sentence. Following previous work in this field, we assume correct named entity recognition and evaluate only the relation extraction's performance, which we model as multiclass classification. In our comparison of SVM kernels, we found that the linear kernel gave better precision, recall and F1 results than the quadratic and cubic kernels – but not significantly better than the quadratic. The linear kernel is also simpler and more efficient, so we chose it for the subsequent experiments.

We investigated the impact of various features on performance by adding them incrementally from shallow (the word itself) to deep (WordNet features). Using the word feature gave an F1 of 33%, which POS and entity features increased to 50%.

Adding further features took the F1 up to 56.78%, which is comparable to the reported best results (55.5%) of Zhou et al. (2005) on the ACE2003 data. The entity features (including in ACE2004 the subtype and class) produced the best improvement – this is not surprising since the relation between two mentions is closely connected with their entity types. In fact, the entity type alone produces an improvement of 10.29%. More accurate information on mentions will therefore contribute to performance in relation extraction.

We also found an interesting trade-off in feature selection because some of the deep features (chunk, dependency tree, parse tree and SQLF) give only 2% improvement in total but directly affect the required quantity of training data and computational resources for training and application.

We also compared (using the complete feature set) relation extraction at three different levels of the relation hierarchy: subtype (23 relations), type (7 relations) and relation itself (a binary task: whether a relation exists between two mentions). We found significant differences: 71.59% F1 for relation detection, 65.20% for type classification and 56.78% for subtype classification (with precision and recall both following the same trend). Thus it is more difficult to classify at deeper levels because there are fewer examples per class and the classes are more similar.

4.3.3 CLIE for Ontology Editing

Controlled Language Information Extraction (CLIE) is an application which will allow users to design, create, and manage information spaces without knowledge of complicated standards such as XML, RDF and OWL, or ontology engineering tools such as Protégé.[4]

The controlled language IE task has developed a simplified natural language processor that allows the specification of logical data for Semantic KT purposes in normal language, but at the same time attains the high accuracy levels necessary for high reliability in applications. The components are based on GATE's existing cascade of finite state transducers (FSTs) for IE. (Cunningham et al. 2002) CLIE is configured so that it either accepts input as valid (in which case accuracy is generally 100%) or rejects it and warns the user of needed repairs to his syntax. Because the parsing process is deterministic, the usual IE performance measures (precision and recall) are not relevant.

Related work includes notably ACE OWL, a sublanguage of Attempto Controlled English, which is well-known and has been adopted as the controlled language for the EU FP6 REWERSE Network of Excellence[5] (Fuchs et al. 2006).

[4] http://protege.stanford.edu

[5] http://rewerse.net/

A web service is available[6] to demonstrate the Attempto Parsing Engine (APE), which uses a definite clause grammar (augmented with features and inheritance) to parse ACE input to a DRS (Discourse Representation Structure) which can be mapped cleaner to first-order logic and OWL DL (Web Ontology Language Description Logic); REWERSE proposes ACE OWL as a means of writing formal, simultaneously human- and machine-readable summaries of scientific papers (Hoefler 2004, Kaljurand and Fuchs 2006, Kuhn 2006).

To be highly expressive but at the same time unambiguous, ACE OWL imposes a number of restrictions, such as the elimination of plural nouns and the requirement for the user to POS-tag words that are not in the predefined lexicon (with which the user must therefore be familiar) (Kaljurand 2006a, b).

In summary, ACE OWL provides in a sublanguage of English much of the expressivity of OWL DL, but it requires training and practice to write correctly. It is, however, generally easy for humans to read.

In developing CLIE and CLOnE (Controlled Language for Ontology Editing, the input language processed by CLIE), we have also given high priorities to human-readability and the elimination of ambiguity, but we have decided to compromise between expressive power and ease of learning at a different point. CLIE and CLOnE are in particular designed to offer the following advantages.

- CLIE requires only one programming language or runtime environment, Java 1.5.
- CLOnE is a sublanguage of English.
- As far as possible, CLOnE is grammatically lax; in particular it does not matter whether the input is singular or plural (or even in grammatical agreement). For example, the input sentences "Book is a type of document.," "Books are types of document." and "Books are a type of documents." have the same semantics.
- CLOnE is case-insensitive.
- CLOnE can be compact; the user can create any number of classes or instances in one sentence.
- CLOnE is easy to learn by following examples and a few guiding rules, without having to study formal expressions of syntax. (Nonetheless, the current implementation of CLIE uses only 11 syntactic rules).
- Any valid sentence of CLOnE can be unambiguously parsed.

Users can therefore learn CLOnE by reading annotated lists of examples, such as the following short list for manipulating ontology classes (and *subclas*s relationships):

- "There are projects and organizations." Create two new classes, Project and Organization, directly under the top class Entity.
- "Universities and companies are types of organization." The class Organization must already exist. Make classes University and Company direct subclasses of Organization and create the first two classes if they do not already exist. (A class can have more than one direct superclass.)

[6] http://www.ifi.unizh.ch/attempto/tools/

- "Forget that universities and companies are types of organization." Unlink the subclass-superclass relationships. This statement does not delete any classes. (A later example in the full user's guide shows how to delete classes.)

The list we have developed for the current version of CLOnE contains 25 such items under the following headings.

- Manipulating classes
- Manipulating instances
- Deleting classes and instances
- Manipulating object properties
- Manipulating datatype properties
- Adding and removing synonyms
- Clearing the entire ontology
- Using the names of classes and instances
- Using lists of classes and instances
- Sentences that can refer to classes or instances
- CLIE is implemented as a GATE application, a detailed treatment of which is given in Funk et al. (2006).

Formally, the CLOnE language consists of 11 unambiguous syntactic rules, each of which has one or more semantic definitions in terms of operations on the ontology. The semantic definitions are disambiguated with reference to the information already in the ontology.

For example this syntactic rule is used to create and instantiate properties ChunkList Copula Chunk Preposition ChunkList and can be used to parse the following examples:

- "Persons are authors of documents." The first and last chunks name existing classes, so a new property is created between them.
- "Carl Pollard and Ivan Sag are authors of 'HPSG'." The first two and last chunks name existing instances, and a suitable property already exists between their immediate or higher classes (created by the previous example), so property definitions are instantiated between them.
- If some of the chunks do not name existing classes or instances in the ontology, an error is indicated.

The software is described in much greater detail, with a comprehensive user's guide, in the relevant project deliverable (Funk et al. 2006).

4.3.4 Evaluation of CLIE

We carried out a repeated-measures, task-based user evaluation of CLIE in comparison with Protégé; each user carried out a similar list of ontology-editing tasks using both tools and completed several questionnaires to give us quantified subjective evaluations of both tools. The users also gave us quantifiable self-evaluations of their expertise through a "pre-test" questionnaire. Half the users carried out task list A with CLIE then task list B with Protégé, and half carried out A with Protégé then B with CLIE; this equalized the effects of fatigue and learning between the tools across the survey without boring the users by requiring each one to do exactly the same task list twice.

We chose the System Usability Scale (SUS) questionnaire as our principal measure of software usability because it is a *de facto* standard in this field. Although it superficially seems subjective and its creator called it "quick and dirty," it was developed according to the proper techniques for a Likert scale (Brooke 1996). It has also been found to produce the most reliable results on small sample sizes, with recommendation to use at least 12–14 subjects (Tullis and Stetson 2004). As a reference or baseline for interpreting the results, average SUS scores are usually between 65 and 70 (Bailey 2006).

We also recorded the time taken to carry out the task lists, and calculated confidence intervals and Pearson's and Spearman's correlation coefficients on a wide variety of combinations of our measurements (Calder 1996, Hildebrand et al. 1977). We concluded that our subjects found CLIE significantly more usable and preferable than Protégé for the straightforward tasks that we assigned. (Of course we make no claims about the more complicated knowledge engineering work for which Protégé but not CLIE is designed and intended.)

Our subjects also made several interesting and useful suggestions for improvements to CLIE, many of which we are considering for future work on this software. Their suggestions included syntax highlighting, better error messages, spell-checking, an undo function with history, and hinting for names of classes and instances.

4.4 Discussion

Our OBIE experiments on the SEKT corpus have demonstrated clearly that the integration of ontologies as a knowledge source within HLT applications leads to improved performance. Another important finding has been that computational efficiency of the underlying machine learning methods is especially important for HLT tasks, as the system may need to train hundreds of classifiers depending on the size of the ontology. Future work will include experiments with more sophisticated cost measures c(X,Y) in addition to the straightforward distance measure, integration of active learning with the Hieron algorithm, and further evaluation on related tasks such as opinion mining.

Our future work on relation-extraction will include an evaluation of our techniques on a larger scale, that is, with a bigger ontology and integration of automatic named entity recognition and relation extraction (our experiments so far have involved relation extraction from the gold-standard named entity annotations). We are also interested in comparing the performance of other variants of the SVM model (e.g., SVM with uneven margins) and other ML approaches (e.g., CRF and MEM).

The semantic annotations produced by OBIE are used extensively as a way to offer semantic-based access to large document spaces. For example, Duke et al. in this book covers the use of semantic annotations in enhanced search and browse tools, whereas Kiryakov et al., also in this volume, demonstrates how faceted search and popularity trend analysis can be carried out over semantic annotations.

In the area of assistive, language-based ontology authoring, our initial work on CLIE has shown clearly that this approach can indeed help naïve users with ontology authoring tasks. Therefore we have started work on embedding CLIE in wikis and perhaps other user interfaces which will eliminate some of the constraints imposed by running it in the GATE GUI, which is not intended for use for such tasks. In particular, the NEPOMUK[7] project, based at DERI Galway,[8] will integrate CLIE into a web service for round-trip ontology authoring for the Integrated Knowledge Articulation and Visualization Workbench as part of the Social Semantic Desktop. NEPOMUK aims to realize and deploy a comprehensive solution – methods, data structures, and a set of tools – for extending the personal computer into a collaborative environment, which improves the state of the art in on-line collaboration and personal data management.

We are also assisting the EPSRC-funded Poleazy project to use CLIE to provide a controlled natural language interface for editing IT authorization policies (access to network resources such as directories and printers) stored as ontologies. This project will probably involve specialized extensions to CLOnE as well as a *virtuous circle* or round-trip information flow, to be completed by generating CLOnE output from formal information with a *shallow NLG* component whose output could be configured with an XML file (Chadwick and Sasse 2006).

Acknowledgments This research has been funded by grants for KnowledgeWeb (EU Network of Excellence IST-2004-507482), TAO (EU FP6 Project IST-2004-026460) and SEKT (EU FP6 Project IST-IP-2003-506826).

References

Appelt D (1999) An introduction to information extraction. Artificial Intelligence Communications, Vol 12(3): 161–172.

Bailey B (2006) Getting the complete picture with usability testing. Usability updates newsletter, U.S. Department of Health and Human Services.

Brooke J (1996) SUS: a "quick and dirty" usability scale. In: Jordan PW, Thomas B, Weerdmeester BA, McClelland AL (eds) Usability Evaluation in Industry. Taylor and Francis, London.

Calder J (1996) Statistical techniques. In: Sapsford R and Jupp V (eds) Data Collection and Analysis, chapter 9. Open University, UK.

Culotta A, Sorensen J (2004) Dependency tree kernels for relation extraction. In: Proceedings of 42th Annual Meeting of the Association for Computational Linguistics. 21–26 July Barcelona, Spain.

[7] http://www.semanticdesktop.org/

[8] http://www.deri.ie

Cunningham H, Maynard D, Bontcheva K, Tablan V (2002) GATE: A framework and graphical development environment for robust NLP tools and applications. In: Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics.

Chadwick D, Sasse A (2006) The virtuous circle of expressing authorization policies. In: Semantic Web Policy Workshop, Athens, GA.

Dekel O, Keshet J, Singer Y (2004) Large margin hierarchical classification. Proceedings of the 21st International Conference on Machine Learning (ICML-2004), Canada.

Freitag D, McCallum A (2000) Information extraction with HMM structures learned by stochastic optimization. In: Proceedings of the 7th Conference on Artificial Intelligence (AAAI-00) and of the12th Conference on Innovative Applications of Artificial Intelligence (IAAI-00): 584–589, Menlo Park, CA. AAAI Press, Menlo Park, CA.

Fuchs N E, Kaljurand K, Kuhn T, Schneider G, Royer L, Schröder M (2006) Attempto Controlled English and the semantic web. Deliverable I2D7, REWERSE Project.

Funk A, David B, Tablan V, Bontcheva K, Cunningham C (2006) Controlled Language IE Components version 2. SEKT project deliverable D2.2.2.

Handschuh S, Staab S, Ciravegna F (2002) S-CREAM – Semi-automatic CREAtion of Metadata. In: Proceedings of the13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02), Siguenza, Spain.

Hildebrand D K, Laing J D, Rosenthal H (1977) Analysis of Ordinal Data. Quantitative Applications in the Social Sciences. Sage, Beverly Hills, CA.

Hoefler S (2004) The syntax of Attempto Controlled English: An abstract grammar for ACE 4.0. Technical Report IFI-2004.03, Department of Informatics, University of Zurich.

Hsu C-W, Lin C-J (2002) A comparison of methods for multi-class support vector machines. IEEE Transactions on Neural Networks, Vol 13(2): 415–425.

Kaljurand K (2006a) From ACE to OWL and from OWL to ACE (http://rewerse.net/events/annual-meeting-2006/abstracts/i2-rewerse_am2006_kaljurand.pdf), presented at Rewerse Annual Project Meeting 2006 (http://rewerse.net/events/annual-meeting-2006/rewerse-annual-meeting-2006-demos-and-abstracts.pdf), March 2006, Munich, Germany.

Kaljurand K, Fuchs NE (2006), Bidirectional mapping between OWL DL and Attempto Controlled English. In: Fourth Workshop on Principles and Practice of Semantic Web Reasoning, Budva, Montenegro.

Kambhatla N (2004) Combining lexical, syntactic and semantic features with maximum entropy models for extracting relations. In: Proceedings of 42nd Annual Meeting of the Association for Computational Linguistic. 21–26 July Barcelona, Spain.

Kuhn T (2006) Attempto Controlled English as ontology language (http://rewerse.net/events/annual-meeting-2006/abstracts/i2-TobiasKuhn.pdf), presented at Rewerse Annual Project Meeting 2006 (http://rewerse.net/events/annual-meeting-2006/rewerse-annual-meeting-2006-demos-and-abstracts.pdf), March 2006, Munich, Germany.

Lafferty J, McCallum A, Pereira F (2001) Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of 18th International Conference on Machine Learning, Morgan Kaufmann, San Francisco, 282–289.

Li Y, Bontcheva K, Cunningham H (2005) Using Uneven Margins SVM and Perceptron for Information Extraction. In: Proceedings of Ninth Conference on Computational Natural Language Learning (CoNLL-2005).

Li Y, Bontcheva K (2007) Hierarchical Perceptron-like Learning for Ontology-Based Information Extraction. To appear in the 16th International World Wide Web Conference.

Li Y, Zaragoza H, Herbrich R, Shawe-Taylor J, Kandola J (2002) The perceptron algorithm with uneven margins. In: Proceedings of the 9th International Conference on Machine Learning (ICML-2002), 379–386.

Motta E, Vargas-Vera M, Domingue J, Lanzoni M, Stutt A, Ciravegna F (2002) MnM: Ontology Driven Semi-Automatic and Automatic Support for Semantic Markup. In: 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02), 379–391, Siguenza, Spain.

Perna J, Spector A (2004) Introduction to the special issue on unstructured information management. IBM Systems Journal, Vol. 43(3), 447–448.

Tullis TS, Stetson JN (2004) A comparison of questionnaires for assessing website usability. In: Usability Professionals' Association Conference, Minneapolis, MN.

Vapnik V (1998) Statistical Learning Theory. Wiley, New York.

Wang T, Bontcheva K, Li Y, Cunningham H (2005) Ontology-based information extraction. SEKT project deliverable D2.1.2.

Zelenko D, Aone C, Richardella A (2003) Kernel methods for relation extraction. Journal of Machine Learning Research, Vol. 3, 1083–1106.

Zhou G, Su J, Zhang J, Zhang M (2005) Combining various knowledge in relation extraction. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics.

Chapter 5
OntoSTUDIO® as a Ontology Engineering Environment

Moritz Weiten

Abstract The development and operation of semantic applications often involves a number of different technologies such as reasoning, text mining and ontology learning. Infrastructures supporting knowledge engineers and users need to provide the requested capabilities in an integrated but flexible way. Also required are basic functionalities such as the storage and management of semantic data and metadata. After a brief survey of current ontology engineering environments, the OntoStudio extensible environment is described.

5.1 Introduction and Motivation

The development and operation of semantic applications often involves a number of different technologies such as reasoning, text mining and ontology learning. Infrastructures supporting knowledge engineers and users need to provide the requested capabilities in an integrated but flexible way. This also includes basic functionalities such as the storage and management of semantic data and metadata.

Platforms for components offering the respective functionalities can be categorized in terms of “runtime support” or “design time support”. This chapter describes a representative of the latter category, the ontology engineering environment OntoStudio®.

Extensible ontology engineering environments do -in contrast to pure editors- not only provide basic functionality such as the storage, management and interpretation of ontologies. They also offer a platform to build and configure customized tools suited for certain tasks such as the semantic enrichment of textual resources or information integration.

M. Weiten
Ontoprise GmbH, Karlsruhe, Germany
e-mail: m.weiten@seeburger.de

J. Davies et al. (eds.), *Semantic Knowledge Management*,

Having an extensible platform allows component providers to benefit from existing capabilities such as the basic features mentioned above. Providing an integrated environment on the other hand helps to reduce the complexity of semantic technologies and increase productivity.

A range of ontology development tools and a large number of components have emerged in recent years. Some focus on core aspects of ontology development, such as editing, navigation and debugging capabilities in a "language-oriented" way, helping to get most out of standard representations such as OWL. Others, like OntoStudio,® include additional functionalities in a more "task-oriented" way, such as semantic wrapping or lifting of existing data sources.[1]

The following sections describe OntoStudio® as an environment that can be used to develop semantic solutions as well as a platform to create a customized knowledge engineering tool.

5.2 Ontology Engineering Tools: State of the Art

In this section a brief description of the state of the art of ontology engineering tools is given. The description is not meant to replace a complete and detailed survey but to offer an overview. A clear focus of current ontology management tools is to support the development of ontologies with a wide range of editing features.

Typical core features of ontology engineering tools include:

- editing and browsing of ontologies
- visualization of ontologies
- import and export of different ontology formats (languages)
- Some tools offer advanced functionalities, such as:
- extensibility via custom plug-ins
- conversion, integration and annotation of external sources
- debugging of capabilities for axioms and/or rules
- ontology mediation (merging and/or mapping of ontologies)
- graphical editing of axioms and/or rules

Probably the most popular environment is Protégé (Gennari et al. 2002). Protégé 3.2 is the latest version of the Protégé OWL editor (Knublauch et al. 2004), created by the Stanford Medical Informatics group at Stanford University. Protégé is a Java-based open source standalone application to be installed and run a local computer. It enables users to load and save OWL, RDF and Frame-based ontologies, edit and visualize classes, properties and SWRL rules (Horrocks et al. 2004), define logical class characteristics as OWL expressions and edit OWL individuals.

[1] This distinction is of course not an exclusive one but it helps to characterize the current landscape of tools.

Protégé is a hybrid tool, as it supports different ontology languages with different paradigms based on a configurable user interface.

Protégé is an extensible platform as it offers a proprietary framework for plug-ins enabling users to extend the tool. The possible plug-ins include custom widgets as well as additional storage backends. Protégé has gained much popularity over the years and thus has a large user-base. Consequently a large number of plug-ins is available. The standard distribution contains plug-ins for graph-based visualization, import of different formats and many more. Additional plug-ins offer for example ontology merging functionalities. The plug-in-mechanism of Protégé is based on a fixed extension-model. Thus only certain forms of functional extensions and user interface extensions are possible. In addition there is no predefined concept for "extensions of extensions".

The commercial TopBraid Composer™ is based on the Eclipse platform and thus offers a flexible extension model (which is explained later in this chapter in the context of OntoStudio®). TopBraid Composer™ is a modelling tool for the creation and maintenance of ontologies.[2] Historically the development of TopBraid Composer™ has its roots in Protégé OWL.[3] Thus some of the concepts of TopBraid™ are similar to those of Protégé, such as the generation of schema-based forms for data acquisition.

TopBraid Composer™ offers functionalities going beyond the creation and management of OWL/RDF(S) files. This includes the import of databases, XML-Schemas, UML and spreadsheets as well as a basic support for rules. The system supports rules in either the Jena Rules format or SWRL. Both types of rules are executed with the internal Jena Rules engine to infer additional relationships among resources. Rules can be edited with support of auto-completion and syntax checking.

Protégé and TopBraid Composer™ are both representatives of extensible ontology engineering platforms. As such they are rather complex and "heavy weight" tools, although Protégé for example offers a number of wizards for tasks like the initial ontology creation and configuration.

A somewhat "light-weight" representative of an ontology engineering environment is SWOOP (Kalyanpur et al. 2005). SWOOP is an open-source hypermedia-based OWL ontology editor. The user interface design of SWOOP follows a browser paradigm, including the typical navigation features such as history buttons. Offering an environment with a look and feel known from web-browsers, the developers of swoop aimed at a concept that average users are expected to accept within short time. Thus users are enabled to view and edit OWL ontologies in a "web-like" manner, which implies navigation via hyperlinks and also annotation features. SWOOP therefore provides an alternative to web-based ontology tools but offers additional features such as a plug-in-mechanism.

[2] At the time of this writing, the TopBraid Composer™ is available as version 2.2.

[3] As pointed out on the TopBraid website mid of 2006: http://www.topbraidcomposer.com/tbc-protege.html

Among the core features of SWOOP are the debugging features for OWL ontologies, exploiting features of OWL reasoners (in this case Pellet). This includes for example the automatic generation of explanations for a set of unsatisfiable axioms (e.g., for a particular class).

5.3 The Ontostudio® Ontology Engineering Environment

OntoStudio® is a commercial product from ontoprise.[4] It is a the front-end counterpart to OntoBroker®, a fast datalog based F-Logic inference machine. Consequently a focus of the OntoStudio® development has been on the support of various tasks around the application of rules. This includes the direct creation of rules (via a graphical rule editor) but also the application of rules for the dynamic integration of datasources (using a database schema import and a mapping tool).

OntoStudio® is available with a main memory- or database-based model, is therefore scaleable and is thus suitable for modelling even large ontologies. Based on Eclipse, OntoStudio® provides an open framework for plug-in developers. It already provides a number of plug-ins such as a query plug-in, a visualizer and a reporting plug-in supporting the Business Intelligence Reporting Tool (BIRT).

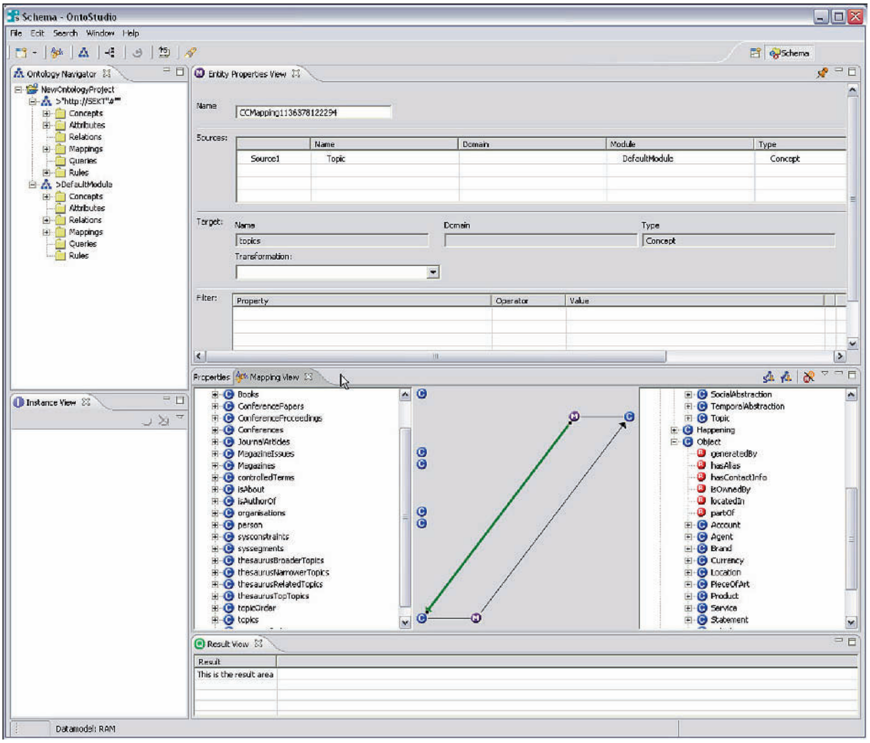
Just like TopBraid Composer™, OntoStudio® is implemented as IDE-application using the Eclipse platform with all the advantages such as the plug-in concept.

OntoStudio® is tightly coupled to F-Logic (resp. its proprietary XML serialization OXML); the import and export from/to OWL/RDF is restricted mainly to concepts which can be expressed in F-Logic. Despite some minor syntactical details the Ontoprise F-Logic dialect conforms semantically to the F-Logic definition (Kifer et al. 1995). Ontoprise is in close contact with the F-Logic forum to work on future versions of F-Logic and further standardization efforts.

OntoStudio® offers a graphical and a textual rule editor as well as debugging features as well as a form-based query-editor. It also includes a graphical editor for the creation and management of ontology mappings including conditional mappings, filters and transformations. Thus OntoStudio® takes advantage of the capabilities of F-Logic regarding rules (such as the support for function symbols) (Fig. 5.1).OntoStudio® targets expert users who want to benefit from the productivity of a graphical tool with wizards, etc.

Basic tasks, like the creation of classes and properties imply a flat learning curve. More advanced tasks require a certain amount of training as well as some background knowledge. This concerns for example the declarative characteristics of logical rules. Those tasks include the graphical and especially the textual creation of rules as well as the creation of advanced mappings.

[4] http://www.ontoprise.com/



Name: CCMapping1136378122294

Sources:

	Name	Domain	Module	Type
Source1	Topic		DefaultModule	Concept

Target:

Name: topics

Domain:

Type: Concept

Transformation:

Filter:

Property	Operator	Value

Result
This is the result area

Fig. 5.1 OntoStudio® (with Mapping-View)

Users with Java™-expertise and a basic knowledge of the Eclipse-platform can extend the tool and use its API to access and modify knowledge bases, create additional graphical components, etc.

5.3.1 OntoStudio as an Eclipse-Based IDE

OntoStudio® is based on the architectural concepts of the Eclipse platform. The Eclipse IDE (integrated development environment) provides both GUI level components as well a plug-in framework for providing extensions to the base platform.

The Eclipse platform itself is highly modular. Very basic aspects are covered by the platform itself, such as the management of modularized applications (plug-ins), a workbench model and the base for graphical components. The real power in the Eclipse platform lies however in the very flexible plug-in concept.

Plug-ins are not limited to particular aspects of the IDE but cover many different kinds of functionalities. For example the very popular Java-development support is not

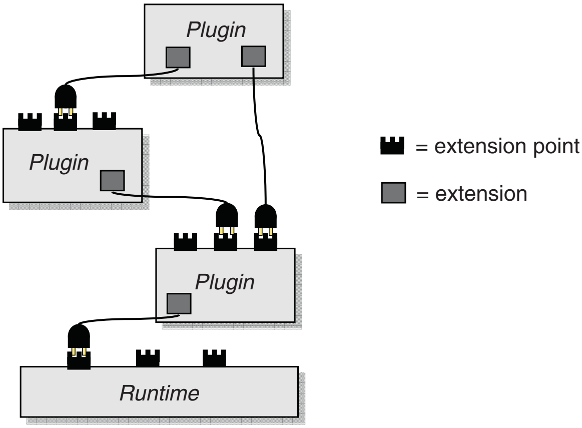


Fig. 5.2 Plug-in concept of Eclipse

provided by the Eclipse platform but by a set of plug-ins. Even functionalities users would consider to be basic (such as the abstraction and management of resources like files or a help system) are realized through plug-ins. This stresses the modular character of Eclipse, which follows the philosophy that “everything is a plug-in”.

A plug-in itself can be extended by other plug-ins in an organized manner. As shown in Fig. 5.2 plug-ins define extension points that specify the functionality which can be implemented to extend the plug-in in a certain way. An extending plug-in implements a predefined interface and registers itself via a simple XML file. In the XML file the kind of extension as well as additional properties (such as menu entries) are declared.

OntoStudio® can be extended by Eclipse-plug-ins in several ways. The OntoStudio® datamodel plug-in provides an API to access and manipulate the ontology model of OntoStudio®. All functionalities that are part of the standard distribution of OntoStudio® are realized as plug-ins which access the datamodel. OntoStudio® also provides a number of extension points. The latter allow extensions of existing components, such as the ontology navigator.

A typical use-case for custom extensions would be an import-filter. An import-filter requires extensions on the level of the ontology-API (writing parsed objects into the OntoStudio® data model) as well as on the GUI-level (extending the “Import” menu and providing graphical wizards).

An important aspect of the extension mechanism of Eclipse is that it supports “extensions of extensions”. This enables OntoStudio® plug-ins like OntoMap® to offer their own extension points. OntoMap® will be presented in the next section.

5.3.1.1 Mediation and Integration: OntoMap® and the Database Import

OntoMap® is a plug-in for the ontology-management platform OntoStudio® that supports the creation and management of ontology mappings via a graphical interface. Mappings can be specified based on graphical representation, using a schema-view of the respective ontologies. Users just have to understand the semantics of the graphical representation (e.g., an arrow connecting two concepts). Thus they do not have worry about the logical representation mappings.

The user of OntoMap® is supported by drag-and-drop functionality and simple consistency checks on property-mappings (automatic suggestion of necessary class-mappings). For concept-mappings[5] constraints can be specified on the available attributes.

OntoStudio® has its own grounding of mappings, based on F-Logic rules. OntoMap® supports simple types of mappings. If more complex mappings are needed (possibly using complex logical expressions or built-ins), the have to be encoded manually. However, OntoMap® still covers a great number of use-cases. The rules that are currently supported include the following *patterns*:

- concept to concept mappings;
- attribute to attribute mappings;
- relation to relation mapings;
- attribute to concept mappings.

Those patterns allow users to apply standard solutions for typical mediation problems in a comprehensive way. Dragging a class of a source ontology onto a target ontology for example triggers a concept-to-concept mapping.

In OntoStudio® queries on stored ontologies can be formed and executed instantly, which gives users the possibility to test the consequence of mappings they have created (or imported).

OntoMap® offers extension points for plug-ins users can implement in order to provide additional mediation features. The most important extension point is the one for mapping patterns. Users can create additional plug-ins which support – possibly complex – mapping patterns as a standard solution for particular mediation problems.

This example shows that through the capabilities of the Eclipse platform OntoStudio®-plug-ins form a modular, highly extensible tool.

5.3.1.2 Other OntoStudio® Plug-ins: Rule Editors, Debugging, Query Formulation, Reporting

The previous section briefly described the OntoMap®-plug-in as a solution for ontology mediation based on the ideas of patterns. The advantage of the latter is the graphical support and the increased productivity given by standardized solutions for typical problems. However, practical cases of ontology mediation might lead to situations where the available patterns are not sufficient.

[5] By the term “concept mappings” refer to mappings from concept to concept.

OntoStudio® offers two plug-ins which give users additional possibilities to target mediation problems in a declarative way: a graphical and a textual rule editor. The graphical rule editor abstracts from details of the underlying representation (of F-Logic rules) and restricts the expressivity supported by the reasoner. It represents rules as graphs and supports drag-and-drop functionality in combination with the ontology navigator. As it is not limited to certain mapping patterns, the graphical rule editor gives end-users additional possibilities (e.g., transitive rules, aggregations, more complex rule bodies/conditions, etc.). The complexity of rule editor is on the other hand higher as the complexity of OntoMap® in the case of simple mapping patterns.

The textual rule editor then requires the highest level of expertise. At the same time offers the full power and flexibility of F-Logic, including function symbols, etc. The latter offers a couple of features, such as syntax highlighting, auto-completion and incremental parsing – all based on the Eclipse editor framework. It has been included recently as there was an increasing interest in "expert features" by end-users targeting demanding problems. This example shows that the modularity and extensibility of OntoStudio® as an Eclipse-based tool allows to drive the development into different directions: towards high productivity and easy-to-use features based on wizards, graphical interfaces, etc. as well as towards "expert features".

OntoStudio® ships with a number of other plug-ins supporting various tasks. This includes –among others– a debugging plug-in for rules, a graphical query formulation tool and a reporting feature based on the BIRT plug-in. The debugger allows users to incrementally analyse their rule-base using a given query. It supports the visualization of dependencies as well as the testing of parts of rule-bodies (single conditions/literals).

The graphical query formulation plug-in allows end users to create and store queries against a given ontology using a form-based interface. Alternatively a simple query interface for the direct formulation of F-Logic or SPARQL queries (Prud'hommeaux and Seaborne 2007) can be used.

Queries can help to test and debug applications. But they can also be used as a view onto data. This possibility is exploited by the reporting plug-in OntoStudio® offers. Users can select given queries and based on those create reports including pie charts, tables, etc.

5.3.1.3 Using OntoStudio® as a Design Time Integration Platform

The previous sections show that one of the main features OntoStudio® apart from its rule and mediation support is the modularity and extensibility. In the EU-funded SEKT project (IP 2003-506826) those characteristics have been exploited in order to establish a platform for the integration of ontology engineering tools. This was part of a dual strategy:

- creating a "runtime platform" for capabilities needed by semantically enhanced applications (such as retrieval solutions);
- creating a "design time platform" for capabilities needed for the creation and maintenance of ontologies and related data and metadata.

OntoStudio® has been selected as a base for the integration of engineering functionalities available for ontology design and associated tasks. The technology of several project partners has been integrated as plug-in. This includes functionalities for information extraction and ontology learning.

5.4 Final Remarks

In this chapter the ontology engineering environment OntoStudio® was introduced. OntoStudio® is one of the few commercial environments available in addition to the open-source tools offered mainly by academic institutions.

The brief overview of ontology engineering tools shows that there are different profiles which serve different needs. OntoStudio® has a clear focus on rules and rule-based information integration. While OntoStudio® offers general-purpose capabilities such as schema-editing functionalities and import-filters for standard formats, it has its strength in the area of semantic information integration and rule-based semantic application development.

Due to its modularity and extensibility OntoStudio® might be seen as a customizable platform rather than a closed tool. OntoStudio® exploits the possibilities of the underlying Eclipse platform, modularizing its functionalities and offering extension points for users.

Acknowledgment This work was supported by the IST Programme of the European Community under SEKT (IST-IP-2003-506826).

References

Gómez-Pérez A, Corcho O, Fernández-López M (2004) Ontological Engineering, Springer, Berlin Heidelberg.

Gennari J, Musen, M A, Fergerson R W, Grosso W E, Crubezy M, Eriksson H, Noy N F, Tu S W (2002) The Evolution of Protégé: An Environment for Knowledge-Based Systems Development.

Ginsberg A, Hirtle D, McCabe F, Patranjan P (2006) RIF Use Cases and Requirements, Working Draft W3C http://www.w3.org/TR/rif-ucr/.

Grau B C, Motik B, Patel-Schneider P (2006) OWL 1.1 Web Ontology Language, XML syntax, W3C Note, http://www.w3.org/Submission/owl11-xml_syntax/.

Horrocks I et al. (2004) SWRL: A Semantic Web Rule Language – Combining OWL and RuleML, W3C Member Submission, http://www.w3.org/Submission/SWRL/.

Kifer M, Lausen G, Wu J (1995) Logical foundations of object-oriented and frame-based languages, Journal of the ACM, Vol. 42, no. 4, ACM Press, New York.

Kalyanpur A, Parsia B, Sirin E, Cuenca B, Grau, Hendler J (2005) Swoop: A Web Ontology Editing Browser, Elsevier's Journal of Web Semantics (JWS), Vol. 4, no. 2.

Knublauch H, Fergerson R W, Noy N F, Musen M A (2004) The Protégé OWL Plug-in: An Open Development Environment for Semantic Web Applications Third International Semantic Web Conference, Hiroshima, Japan.

Mizoguchi R (2004) Ontology Engineering Environments, in Studer, R., Staab, S. (eds), Handbook on Ontologies, Springer, Berlin Heidelberg: 275–298.

Noy N F, Musen M A (2004) Ontology versioning in an ontology management framework, IEEE Intelligent Systems, Vol. 19, no. 4, 6–13.

Palma R, Hartmann J, Gómez-Pérez A (2006) Towards an Ontology Metadata Standard, 3rd European Semantic Web Conference (ESWC), Budva.

Prud'hommeaux E, Seaborne A (2007) SPARQL Query Language for RDF, W3C Working Draft, http://www.w3.org/TR/rdf-sparql-query/.

Sirin E, Parsia B, Grau B C, Kalyanpur A, Katz Y (2007) Pellet: A practical OWL-DL reasoner, Journal of Web Semantics, Vol. 5, no. 2.

Chapter 6
Shared Ontology for Knowledge Management

Atanas Kiryakov, Borislav Popov, Ilian Kitchukov, and Krasimir Angelov

Abstract This chapter focuses on semantic searching at web scale. The solution presented takes advantage of the specific strengths of semantic repositories and the raw power of relational databases, the latter having been developed over decades and capable of handling efficiently large volumes of data with fixed structure (which is the case with the occurrence statistics) and the former allowing for inference and querying on top of formal knowledge. The interactive faceted search capability described is a demonstration how an approach based on these two technologies is more powerful and efficient for certain tasks as compared to traditional search engines.

The contributions of the chapter can be summarized as follows:

- novel indexing schema for semantic search, based on entity occurrence;
- description of a scalable implementation of such indexing;
- advanced faceted search interface, based on co-occurrence.

6.1 Introduction

The enormous scale of the web sets a new challenge – handling this scale, so that the users do not get lost and can efficiently perform their intended tasks. The ranking of documents and providing the most relevant information first is a step towards alleviating this information overload and has already been addressed to some extent by contemporary search engines. Another approach, which has not yet been seen in large scale applications, involves some linguistic and semantic analysis in order to better *understand* the user need and the data on the web and match

A. Kiryakov(✉)
Ontotext Lab, Sirma Group Corp, Bulgaria
e-mail: atanas.kiryakov@ontotext.com

J. Davies et al. (eds.), *Semantic Knowledge Management*,

them at a deeper level. This analysis may result in the production of semantic annotations, as defined in Kiryakov et al. (2005b), and represents an aspect of the annotated resources that provides machine-readable insights into the meaning of the content. Our belief is that the scalable and automatic extraction of semantic annotations for the existing web is the route towards real-world Semantic Web (SW) applications. Such metadata would allow SW applications to provide new ways of searching, navigating, summarizing, and analyzing the web resources. For several years, we have been developing the KIM Platform (Popov et al. 2004) for semantic annotation, indexing and retrieval, which basic functionality is presented in Sect. 6.1.1 below. In a nutshell, it allows for efficient semantic annotation and indexing of documents with respect to named entities (NE). The basic indexing approach is preceded by processing of the text content, which is then indexed with a standard information retrieval (IR) engine, as discussed in Sect. 6.0. Such indexing allows KIM to perform hybrid queries, combining full-text search, structured queries, and inference.

Recently we have extended KIM with a new module called CORE, focusing on co-occurrence of entities and providing for incremental searching based on this information, as well as ranking, tracking trends and popularity timelines of these entities. Having some experience in extracting predefined relations between named entities and understanding the extra investment of effort for defining extraction algorithms for new relation types, we decided to extend KIM in the direction of tracking associative (or soft) relations between entities, based on their co-occurrence in the same block of content (or context). The information gathered in the CORE module is based on the semantic annotations produced by KIM and provides another perspective to the results of the semantic analysis of the content as compared to the semantic repository where all the extracted entity instances are stored with their names, relations, and attributes. CORE focuses on tracking some statistics for the bi-directional relations between entities and documents, while leaving the semantically heavier descriptions of entities to the semantic repository.

The KIM Platform is generally independent of the domain it is applied to, but the concrete configuration context provided by default (ontology, instance base, information extraction modules) is tuned for analysis of international news. As in any domain, if one acquires large amounts of data to work on, interesting co-relations of the domain entities could be monitored, shaping trends and dependencies that are not clear without the automatic assimilation of large volumes of information. If the data is also aligned with particular points in time, timelines of these co-relations or of the popularity of entities could be obtained.

The concrete solution presented here takes advantage of the specific strengths of semantic repositories and the raw power of relational databases. The latter having been developed over decades and capable of handling efficiently large volumes of data with fixed structure (which is the case with the occurrence statistics); the former allowing for inference and querying on top of formal knowledge. The combination of the two in one system is like the unity of physical strength and prudent mind in a person. The interactive faceted search (named CORE search) is a demonstration how an approach based on these two technologies is more powerful and more efficient for certain tasks as compared to traditional search engines.

The contributions of this paper can be summarized as follows:

- novel indexing schema, based on entity occurrence;
- description of a scalable implementation of such indexing;
- advanced faceted search interface, based on co-occurrence.

After this section, the paper continues with an overview of the related work explaining some closely or a bit more remotely relevant efforts. A short description of the environment in the third section sets the basis for the better understanding of the work. The general idea and the conceptual models behind the CORE module are explained in section four. The overall architecture of KIM and some technical details on the annotation and indexing processes are discussed in Sect. 6.5. The faceted search user interface named CORE Search is presented in Sect. 6.6; it provides a demonstration of the capabilities of the CORE module and a pictorial pathway for understanding of the featured functionality. In the last two sections we elaborate some of our intentions for the future and conclude the chapter.

6.1.1 Semantic Annotation in KIM

The main focus of KIM (before the presented extension with the CORE module) was on providing the necessary infrastructure for scalable automatic extraction of named entity (NE) references and descriptions, including attributes and relations, from text. We call both the process and the result semantic annotation. Ontologies and factual background knowledge (held in a semantic repository) are used for analysis of text, implemented as information extraction (IE) application in GATE (see Sect. 6.0). The outcome is twofold:

- extension of the semantic repository with structured data, extracted from the text;
- generation of annotations (metadata) that link the text with the repository.

Figure 6.1 presents the interlinking between the text and the data; each annotation contains an identifier of entity, described in the semantic repository. KIM recognizes both known and unknown NE, instances of predetermined classes, as well as attributes and relationships between them. In addition to the NE, KIM is also extracting and annotating key-phrases, found to be statistically characteristic for the document. Those are stored in the repository as an instance of a special class called *GeneralTerm* and treated as regular entities.

KIM is designed to be able to take advantage of pre-existing structured data and inference – the text-mining algorithms are initialized from a semantic repository, which can handle large amounts of instance data. A major task in the process of semantic annotation is called *identity resolution*: to find whether a particular NE reference denotes an already known NE in the repository and which one. In case the reference has no match in the repository, a new entity description is created and

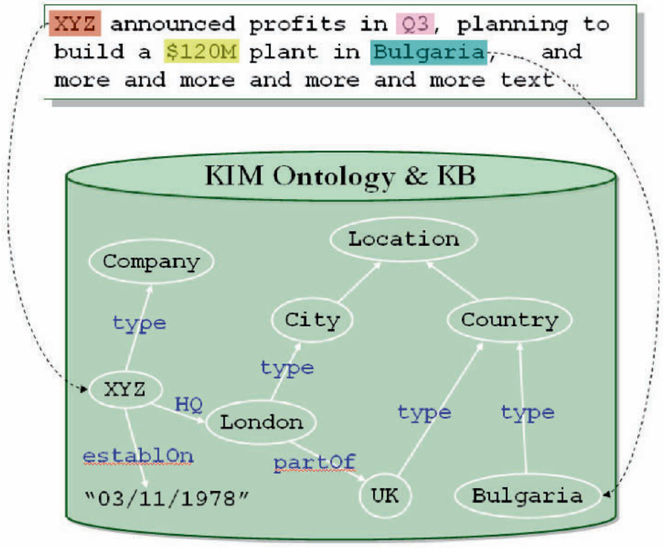

XYZ announced profits in Q3, planning to build a $120M plant in Bulgaria, ... and more and more and more and more text ...
KIM Ontology & KB
Company
Location
City
Country
type
type
XYZ
HQ
London
type
type
establOn
partOf
"03/11/1978"
UK
Bulgaria

Fig. 6.1 Semantic annotation in KIM

stored in the ontology. As a result of the text analysis, KIM can also extend the descriptions of known entities by means of finding new attribute values and relationships.

Both KIM's architecture and ontologies (Terziev et al. 2005) allow for easy domain specific specialization on the principle of modular ontological extensions and pluggable modules for IE.

6.1.2 Inverted Indices and Vector-Space Model for Information Retrieval

Traditional IR engines index text documents with respect to the string tokens mentioned in them. Usually, tokens are considered all the different strings which appear in the text separated by white-space or punctuation. Such are different forms of words, all the spellings of names, abbreviations, numbers, etc. The so-called full-text indexing and search is based on inverted indices which allow for retrieval of documents by tokens mentioned in them. The standard way to formulate a query, that is, to express user's information need, is to provide several tokens, which should appear in the documents of interest. In the Boolean search model, whenever multi-token query is evaluated, the engine uses the inverted indices to check for documents where all of the query tokens appear.

A task that appears very important, when dealing with large sets of documents, is not only to retrieve the documents, but also to order (or rank) them by relevance to the query. For this purpose a model richer than inverted token-document index is necessary. The most popular approach for relevance ranking in IR is the so-called Vector Space Model (VSM). The underlying assumption is that documents are considered as objects which are characterized by the tokens appearing in them. This way, tokens represent a set of features of the documents. It is important that the degree of association is considered, so that some tokens are considered more characteristic for a document than others. The abstraction used to formally model these "weighted" characterization is a geometrical one: documents are represented in a space, where each dimension is associated with a token. Thus if there are 10 million different tokens appearing in the indexed documents, the space used as a model for relevance ranking will have 10 million dimensions. Each document is modeled as a vector, where the number of occurrences of a specific token in it is taken as a coordinate value for the corresponding dimension. The engine represents the query as a vector in the same space and then takes the cosine of the angle between the query and the document as a relevance measure.

In practice, IR engines implement quite a number of normalizations on top of the basic VSM in order to handle issues like very popular terms, documents of varying size, fields of special importance (e.g., title), etc.

6.1.3 Semantic Indexing and Retrieval in KIM

Once the semantic annotations are in place (as discussed in Sect. 6.0), they are used for indexing of the documents. KIM implements a schema which forces a standard IR engine to index the document not only by string tokens, but also by entity identifiers. This is achieved by simple pre-processing of the queries and the documents sent for indexing. Technically, the identifiers of the entities are added as new unique tokens (or features), which appear as new dimensions in the vector-space model.

The immediate benefit is that documents can be retrieved by entities, which represents a sort of conceptual search that abstracts the user from spelling (surface realization) of the references to the entities. For example one can find documents which refer "UK", "United Kingdom", and "U.K." with one simple query, asking for document, characterized by the identifier of UK, as it appears in the semantic repository.

This type of semantic indexing allows also hybrid queries, combining full-text search, structured queries, and inference. Consider for example, the following query:

"telecom company in Europe" "John Smith" director

The information need appears to be for documents concerning a telecom company in Europe, a person called John Smith, and a management position. Note, however, that a document containing the following sentence would not be returned using conventional search techniques:

"*the board of O2 appointed John G. Smith as CTO*"

In order to be able to return this document, the search engine would need to be able to "consider" several semantic relations and apply several inference rules:

- O2 is a mobile operator in UK;
- Mobile operators are a special case of telecom companies;
- UK is part of Europe;
- CTO is a kind of director.

These are precisely the kinds of relations which can be present in the semantic repository – either as pre-loaded background knowledge or as facts extracted from documents. Finally, to match the query, the semantic repository should be able to perform simple inference like: *if O2 is located in UK and UK is part of Europe, then it is also located in Europe*.[1]

6.2 Related Work

Very interesting and relevant work has been done at the Joint Research Center of the EC in Italy by the group of Ralf Steinberger and Clive Best. One of their products is called EMM NewsExplorer and analyses news articles in several languages. The extracted information is presented in a knowledge base and static web pages are produced to ensure stable and fast access for the users. Besides extracting key-phrases and named entities, NewsExplorer calculates clusters of news articles where each cluster represents a single event covered in the news. This analysis is also cross-lingual and each cluster is formed of articles in different languages referring to the same event. It is obvious that searching technology helps users find their way in the contemporary information overflow, but the identification of a single *event* out of many news articles that discuss the same topic brings the usability of NewsExplorer to a new level. JRC also provides timelines of major news on the basis of the sizes of the event clusters. Another interesting aspect is the front-end provided to users to help them navigate through the news. For each news article there are people and organizations that are either related to or associated with the content. Similarly for each entity one can see the articles that refer to it. Generally their ideas, efforts and results are remarkable in several aspects: multilinguality, IE, clustering of news to find out events, and presentation of the news to the end-user.

Rocha et al. (2004) describe a search architecture that applies a combination of spread activation and conventional information retrieval to a domain-specific semantic model in order to find concepts relevant to a keyword based query. The

[1] With thanks to John Davies, who first proposed this example.

essence of spread activation, as applied in conventional textual searching, is that a document may be returned by a query, even if it contains none of the query keywords. This happens if the document is linked to by many other documents which do contain the keywords.

The query being a set of keywords is forwarded to a conventional search engine which assigns a score to each document in its index in the usual way. In addition to a conventional index, the system contains a domain-specific KB, which includes instance nodes pointing to web resources. As usual in RDF, each instance is described in terms of links labeled with properties in compliance with the ontology. The basic assumption is that weightings that express the strength of each instance relation can be derived.[2] Thus the resulting network has, for each relationship, a semantic label and an associated numerical weight. The intuition behind this approach is that better search results can be obtained by exploiting not only the relationships within the ontology, but also the strength of those relationships.

The QuizRDF search engine (Davies et al. 2003) combines FTS with a capability to exploit RDF annotation querying. QuizRDF combines search and browsing capabilities into a single tool and allows document-level RDF annotations to be exploited (searched over) where they exist, but will still function as a conventional search engine in the absence of those annotations.

A document can be described with properties appropriate for the class of its subject. For example a document associated with subject *Employee* can be given attributes *firstName, lastName* (inherited from the *Person* super-class) and position. Full-text indexing is applied not only to the content of the documents, but also the literal values of any relations. This way a keyword search for "*CTO*" will return this document, although this string may not appear in the document.

This datamodel would have been clearer if the documents and the objects they are about were defined as separate RDF resources, for example, *<cv,isAbout,emp1>, <emp1,type,Employee>, <emp1,firstName,"George*">. On the other hand, this model has all the advantages of being more simple and efficient.

TAP (Guha and McCool 2003) is a Semantic Web architecture, which allows RDF(S)-compliant consolidation and querying of structured information. In Guha et al. (2003), two Semantic Web-based search engines are described: ABS – activity-based search and W3C Semantic Search. In both cases TAP is employed to improve traditional search results (obtained from Google) when seeking information in relation to people, places, events, news items, etc. TAP is used for two tasks:

Result augmentation: the list of documents returned by the IR system is complemented by text and links generated from the available background knowledge;

[2] A number of different approaches to this derivation are taken and the authors state that no single weight derivation formula is optimal for all application areas.

Query term disambiguation: the user is given the opportunity to choose the concrete entity she is searching for, than the system attempts to filter the results of the IR system to those referring only this entity. An approach using several statistics for this purpose is sketched in Guha et al. (2003) without details of the implementation.

A system called SemTag is presented in Dill et al. (2004), which performs automatic semantic annotation of texts with respect to large scale knowledge bases available through TAP, solving a task similar to the one solved by the KIM Platform (Popov et al. 2004).

Recently Google released Google Trends, which allows the user to choose keywords and see the search frequency for these in a timeline chart. Based on this timeline, one can see the change in frequency and sometimes news articles associated with a point on that timeline, but without a particular relation to the extremes in that timeline. An interesting feature is that the statistics can be restricted by region and thus track trends specific only for this geographic area or location.

6.3 Description of the Environment

The CORE module is a part of the KIM Platform which consists of an ontology, a world knowledge base, a KIM Server providing API for remote access, embedding, and integration and some front end applications. It is based on robust open-source platforms specialized in three different domains:

- *Ontology management*: the knowledge resources are kept in the SESAME[3]-based OWLIM[4] semantic repository, which provides storage, inference, and query infrastructure on top of RDF(S) or OWL;
- *Text mining*: the GATE[5] platform has been used as a basis for the information extraction (IE) processing and also as framework for the management of content and annotations. It provides the fundamental text analysis technologies, on top of which we have built the semantically aware extensions, specific for the IE of KIM.
- *Information retrieval* (IR): the Lucene[6] IR engine has been adopted to perform indexing, retrieval and evaluation of content relevance with respect to named entities.

The CORE module is extending the combination of the mentioned technologies towards a relational database management systems (RDBMS), taking benefits of

[3] http://www.openrdf.org – one of the most popular RDF repository frameworks.

[4] http://www.ontotext.com/owlim – the fastest and most scalable RDF(S)/OWL semantic repository.

[5] http://www.gate.ac.uk – the "General Architecture for Text Engineering" – the most popular NLP and text-mining platform developed in University of Sheffield.

[6] http://lucene.apache.org/ – the most popular open-source IR engine.

the specific strengths of these engines. The actual implementation featured here is based on Oracle,[7] although the design of the CORE module allows for implementations based on other RDBMS. It takes over some of the functionality present in the semantic repository and the IR engine, enriching it with co-occurrence and ranking of entities.

The corpus used for populating CORE with data exceeds 1 million news articles in English. It includes news from numerous sites like BBC, CNN, AP, AFP, Reuters, etc. The corpus currently covers all major news for the period from 2002 till now. Each news article is associated with a point in time when it has been published. This allows statistics over time to be calculated for the contained entities that were recognized, as well as to track the popularity of these entities.

6.4 Co-Occurrence and Ranking Model

The model of KIM's CORE module differs from the traditional IR indexing approach (see Sect. 6.0) in two ways:

- It considers as features of the documents not all the tokens appearing in them, but rather only the named entities and the key-phrases. This leads to reduced-dimensionality vector-space model.
- While the inverted indices support only retrieval of documents by features, CORE maintains bi-directional indices, which allow also for retrieval of features by documents. This way CORE allows for efficient retrieval of entities, which are mentioned in documents.

The reduction of the dimensionality of the space is a technique used in other advanced IR approaches, for example, LSA (Landauer and Dumais 1997). When implemented "properly" it allows for indexing and retrieval by concepts, a sort of improved feature space, which is a better representation of the meaning of the documents, as compared to the strings of characters appearing in the text. The schema selected by us has the following advantages:

- The selection of the dimensions is cheaper and can be performed in a more dynamic fashion, as compared to statistical methods like LSA. This is because, we do not need heavy computational effort to derive the features – we take the named entities and the key-phrases which come as output of the semantic annotation;
- The reduced dimensionality allows efficient management and scaling of the bi-directional weighted index of CORE. As it is more expensive to maintain than a simple inverted index, such an index is impractical if applied over the entire original feature space;

[7] http://www.oracle.com/ – one of the leading RDBMS.

- It can always be complemented by a standard full-text search index, maintained in parallel. This is the case implemented in KIM.

This new indexing schema allows for efficient handling of several new types of queries. One of them is CORE's capability to track co-occurrences and answer queries focused on entities as an end result like *give me all entities that co-occur with Europe and Iran*. Although we have shown ways of altering a full-text search (FTS) engine to work with respect to named entities (Popov et al. 2004), such queries cannot be processed by these engines for the sole reason of the organization of their indices which point from the indexed terms to the documents but not vice versa. In CORE one could ask for co-occurring entities, their rank or the deviations in their occurrence in time (timelines) conforming to restrictions like time period or a sub-set of the corpus. Reducing the feature space to entities allows for effective tracking of these correlations between entities, time points and documents for large sets of data (millions of documents) in a relational database. Modeling of occurrence tracking, FTS indexing over the content, class hierarchy, and a significant part of the entity description in the database allows for expressive and efficient querying on a combination of these features without the need to intersect result sets from different data sources like an IR engine, a semantic repository, and a database.

As presented in Sect. 6.0, the CORE module allows incremental search and timelines based on extracted semantic information and on entities rank and co-occurrence. It is essential that this is a mixture of both semantic and statistical information. The semantic information is about the entities relations and classification, while the entity rank is purely statistical information. The co-occurrence information has both statistical and hidden semantic meaning. We will show that this mixture allows for a new way of searching and analysis that makes it easier for the user to find a set of documents relevant to her need.

When two entities co-occur in the same context usually this is a clue for a kind of semantic relation between them which may or may not be explicitly expressed. The IE of KIM extracts relations like *Organization* located in *Location; Person* having a *Position* within an *Organization* or associated with a *Location* (e.g., a mayor), but the set of possible relations is unlimited. Even if the user interest is in relations that are not recognized explicitly, still, one can easily find related entities and respective documents based on the associative relations found on the basis of co-occurrence in the same context.

The nature of ranking is statistical, but from that we also can derive semantic information. Let us suppose that we are interested in the relations of *John* with other people. CORE allows asking about the *most popular* people that are mentioned together with John. The term "most popular" means "*with the highest rank in the set of documents that also mentions John*". The co-occurrence of the not so popular people will be considered as a less relevant part of the result set.

Since each document in the corpus has explicit date, we can track trends in time (timelines) for a set of entities based on their occurrence or co-occurrence frequency. Having these timelines, one can find out the important events in the *life* of a given entity following the extreme points of this function.

6.5 System Architecture

We have to manage both semantic and statistical information and this is inefficient or impossible with the usage of either semantic repositories or contemporary IR engines. For storage of semantic information the usage of semantic repository is the natural decision but for processing (grouping, sorting and aggregation) of large volumes of numeric data the modern database systems are much more efficient. KIM employs a semantic repository [based on SESAME and OWLIM (Kiryakov et al. 2005a)] and a relational database (based on Oracle). OWLIM is the fastest and most scalable semantic repository currently known, while Oracle has been the leading database system for a long time. Scalability is a key design goal because we intended to build a system that could handle millions of documents and hundreds of millions of entity co-occurrences.

The KIM Server automatically extracts semantic annotations from unstructured text and associates these with the processed document (Fig. 6.2). Each annotation keeps URI references to the extracted entity. The annotated documents with the entity occurrence information are stored in a RDBMS, while the semantic description of entities is stored in the semantic repository. The usage of two heterogeneous systems inherently causes duplication of some pieces of information. To keep the two in synchrony we have to keep the correspondence between the entity description in the semantic repository and the database.

There are three major types of information stored in the database (Fig. 6.3): entity information, document information and the relation between them (information for the occurrence of entities in the documents).

The entity information is the mapping between its identity in the database and in the semantic repository. Since the database indices are optimized to work with integers while the semantic repository is based on URIs, we have to generate one

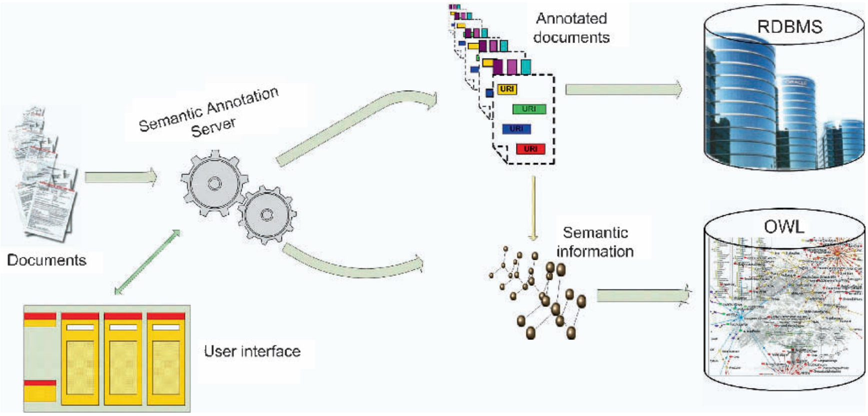


Fig. 6.2 Partial architecture of the KIM platform

unique integer ID for each entity URI. This is the most important part of the entity information since it links its representation in the database to the full entity description in the semantic repository. In some kinds of queries we are interested only in entities of a given class or with a given alias. The entity class and aliases are stored in the semantic repository but for performance reasons we have to keep them in the database too. This is not strictly necessary but it is the only way to make the queries efficient without the need to aggregate query results from different sources.

All documents, from which the semantic information was extracted, are stored in the database. The important features of a document are the content and the document-level metadata, among which - its date. The text is required if you want to combine FTS queries with purely semantic queries. The document date is necessary for the modeling of entity occurrences with respect to time points.

While the above two pieces were related to two disjoint aspects of the data (i.e., documents and entities) the third one is the connection between them – a many-to-many relation between entity and document. Each entity can be referred to in many documents and each document refers to many entities. For each unique pair entity/document the number of occurrences is kept so that we can compute the entity rank. The actual computation of ranking in a given sub-corpus requires aggregation and sorting of the occurrence information in each individual document. The presentation of entity occurrence in the database also allows for querying the co-occurrence of entities in the same context using efficient relational operations.

The current scale of the CORE module has been examined with about a million documents and all the respective entity occurrences. For such amounts of potential results the first step towards successful information retrieval is to limit the results either by FTS keywords, time period, or selecting an entity and incrementally restricting the set of results until a comprehensible scale is achieved. By selecting an entity, one restricts the set of results to the ones that refer to this entity, and the options for further restrictions are based on further selecting one of the entities that co-occur with the selected one.

All further queries are performed only in the scope of the selected sub-corpus. Currently CORE can retrieve the following types of results: a list of all documents contained; count of the documents in the sub-corpus; a list of all or the first N most popular entities referred from a given class; the number of entities referred; rank or popularity timeline for a given entity (or set of entities) within the sub-corpus in focus; ranking timeline for the first N most popular entities in the

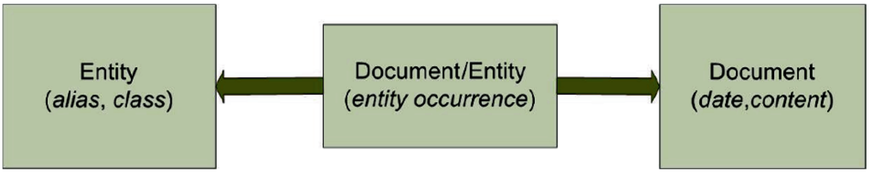


Fig. 6.3 Part of the information stored in the database

(sub-)corpus; time distribution of the documents in the corpus based on daily, monthly or yearly units.

The KIM Platform is a distributed system with Java RMI interface between the server and client sides (Popov et al. 2004). The CORE module API is only a part of the interface provided by the KIM Server. The CORE API is divided into two parts: one for general operations over documents and entities and the other for timeline queries. There is a query object that represents the possible restrictions over the data and it is passed to the different methods of this API which return a list of entities, list of documents or a timeline. The API for timelines requires extra parameters like the time unit and the entity set in focus. All methods in the timelines API return an object that represents the time distribution of the featured documents or entities.

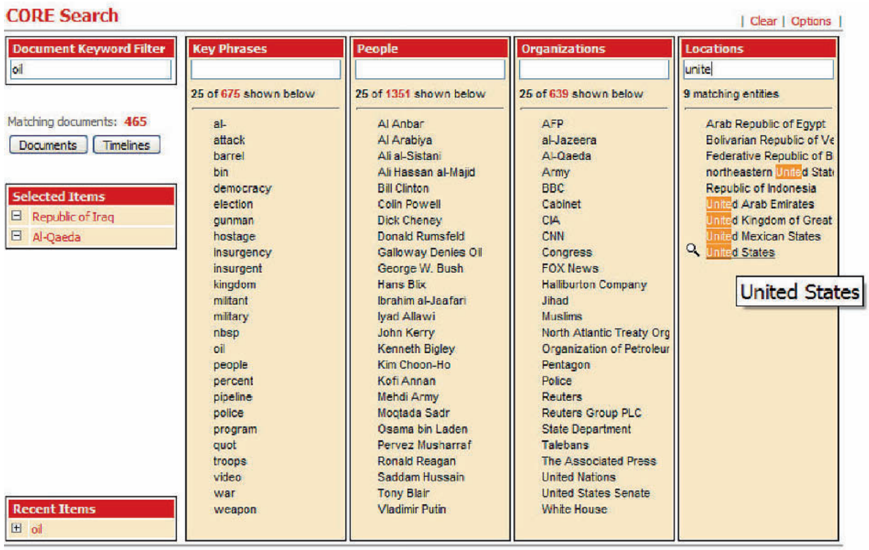
6.6 User Interface

The user interface that gives access to the CORE resources and presents a demonstration of the server functionality allows the user to perform different requests and is visually divided into two parts. The first one is called CORE Search and provides the faceted search functionality, while the second one called Timelines is intended to give the user an easy way to rank entities by popularity and see occurrence trends for specific entities of interest. There is still another piece of functionality that is not separated from these, but is very useful along with CORE Search and Timelines. This is the option to see documents' distribution over time when searching for documents matching certain criteria and given a time period.

CORE Search utilizes most of the functionality CORE was designed for. It also presents some nice features that make it easier and more convenient to narrow down the documents to the desired set. There are generally two ways the user can restrict documents of interest (Fig. 6.4):

- words or phrases can be typed in the *Document Keyword Filter* field to limit documents to those that contain them. This is the standard token/term/key-word-based full text search restriction;
- entities can be filtered by name and added to the *Selected Items* list to restrict the documents to only these in which the selected items appear.

The entities could be selected on the right side of the interface shown in Fig. 6.4, where they are listed in columns by class. To find and add a specific entity to the filter the user can type in a (part of) the entity name in the alias filter above the list for the desired class. This will narrow down the entities in the list and highlight will appear for the filter matches. Afterwards, the user can browse the full list of entities by clicking on the number representing the *entity result count* if the searched one is not displayed in the top ranked entities list. Another option is to click on the magnifying glass icon beside an entity and see the full entity description which will be requested from the semantic repository (Fig. 6.5). The entity description explorer



Document Keyword Filter: oil

Matching documents: 465

Selected Items
- Republic of Iraq
- Al-Qaeda

Recent Items
- oil

Key Phrases

25 of 675 shown below

al-
attack
barrel
bin
democracy
election
gunman
hostage
insurgency
insurgent
kingdom
militant
military
nbsp
oil
people
percent
pipeline
police
program
quot
troops
video
war
weapon

People

25 of 1351 shown below

Al Anbar
Al Arabiya
Ali al-Sistani
Ali Hassan al-Majid
Bill Clinton
Colin Powell
Dick Cheney
Donald Rumsfeld
Galloway Denies Oil
George W. Bush
Hans Blix
Ibrahim al-Jeafari
Iyad Allawi
John Kerry
Kenneth Bigley
Kim Choon-Ho
Kofi Annan
Mehdi Army
Moqtada Sadr
Osama bin Laden
Pervez Musharraf
Ronald Reagan
Saddam Hussain
Tony Blair
Vladimir Putin

Organizations

25 of 639 shown below

AFP
al-Jazeera
Al-Qaeda
Army
BBC
Cabinet
CIA
CNN
Congress
FOX News
Halliburton Company
Jihad
Muslims
North Atlantic Treaty Org
Organization of Petroleur
Pentagon
Police
Reuters
Reuters Group PLC
State Department
Talebans
The Associated Press
United Nations
United States Senate
White House

Locations: unite

9 matching entities

Arab Republic of Egypt
Bolivarian Republic of Ve
Federative Republic of B
northeastern United Stat
Republic of Indonesia
United Arab Emirates
United Kingdom of Great
United Mexican States
United States

United States

Fig. 6.4 CORE search user interface

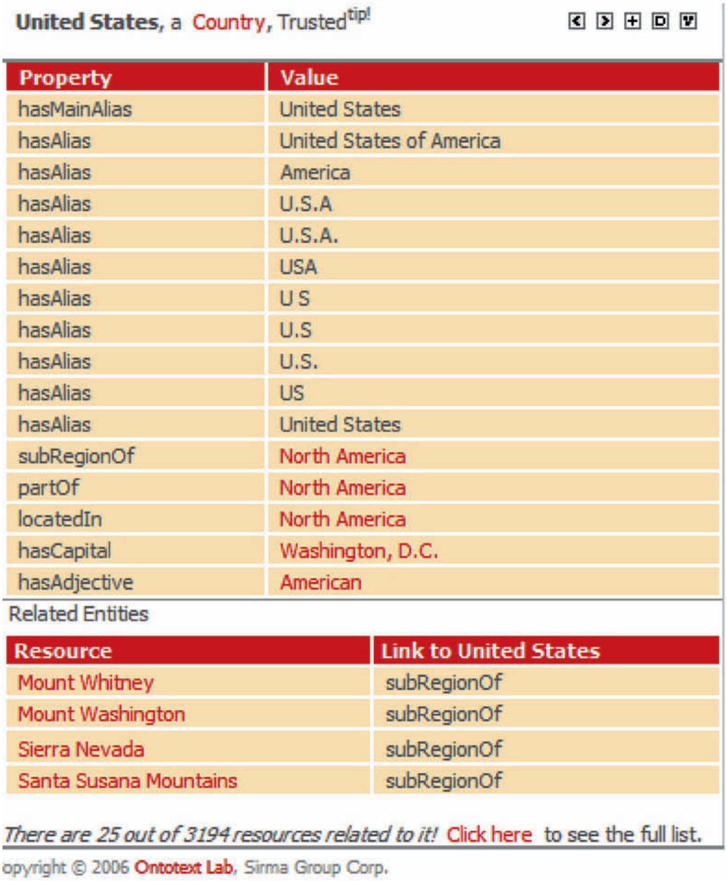
also allows further navigation through the instance base by following the links to the proper class or the related entities.

By selecting an entity from one of the lists it is being transferred to the *Select Items List* and the document set and displayed entities are being refreshed. The new result displays the updated count of documents containing the selected entities. The entity columns are refreshed to show only the entities co-occurring with the selected ones, thus ensuring that the document result can be further narrowed down. As the example in Fig. 6.4 shows, there are 1351 people that appear in the same 465 documents, that were restricted by a keyword and two selected entities.

The list of displayed entities for each class is limited by the obvious space restrictions of the single page layout of the user interface. The visible entities are the top ranking ones given the current co-occurrence context, entity name and FTS restrictions. If the entity of interest is not there, either a more restrictive filter can be entered or the full entity list can be browsed. In Fig. 6.4 it is seen that only the 25 most popular people for the given set are shown out of more than thousand that meet the restrictions.

Selecting too many entities would lead to a situation where the document or entity set has been over-restricted and too small. At this point entities could be removed from the *Selected Items List* and will appear as history but can easily be moved back to the selected list later.

Another characteristic of the CORE Search UI is that it is highly interactive. For instance, the count of matching documents is updated on the run as the user types in tokens in the keyword filter field. Entities are filtered in the same way: as one enters a part of the name of the needed entity, the list of the matching entities is being updated dynamically. This is another demonstration of the extreme efficiency



United States, a Country, Trusted^tip!

Property	Value
hasMainAlias	United States
hasAlias	United States of America
hasAlias	America
hasAlias	U.S.A
hasAlias	U.S.A.
hasAlias	USA
hasAlias	U S
hasAlias	U.S
hasAlias	U.S.
hasAlias	US
hasAlias	United States
subRegionOf	North America
partOf	North America
locatedIn	North America
hasCapital	Washington, D.C.
hasAdjective	American

Related Entities

Resource	Link to United States
Mount Whitney	subRegionOf
Mount Washington	subRegionOf
Sierra Nevada	subRegionOf
Santa Susana Mountains	subRegionOf

There are 25 out of 3194 resources related to it! Click here to see the full list.

opyright © 2006 Ontotext Lab, Sirma Group Corp.

Fig. 6.5 Semantic description of an entity

of the CORE module – effectively, each character entered by the user causes numerous non-trivial queries, which are handled in real time, usually, matching the speed of typing.

The user can also change the number of entity columns to be shown and assign a custom class to each of them from the *Options menu* (top right of Fig. 6.4).

Finally, when narrowing down the document set to a browsable scale, one can view the document result matching the current restrictions. Along with the resulting documents list, one also sees a chart of the distribution of these documents over time (bottom of Fig. 6.8).

If not interested in documents but in the timelines of entity popularity, one can choose to see these using the restricted document set as a basis for their calculation. *Timelines* interface allows trends to be calculated and conveniently viewed and navigated through a chart. The calculation can be performed over all the documents in store or just over some limited document set that came from *CORE Search*.

Selecting a time period for the calculation will also limit those documents. Depending on the length of that period a granularity can be set that ensures the desired level of details for the resulting chart.

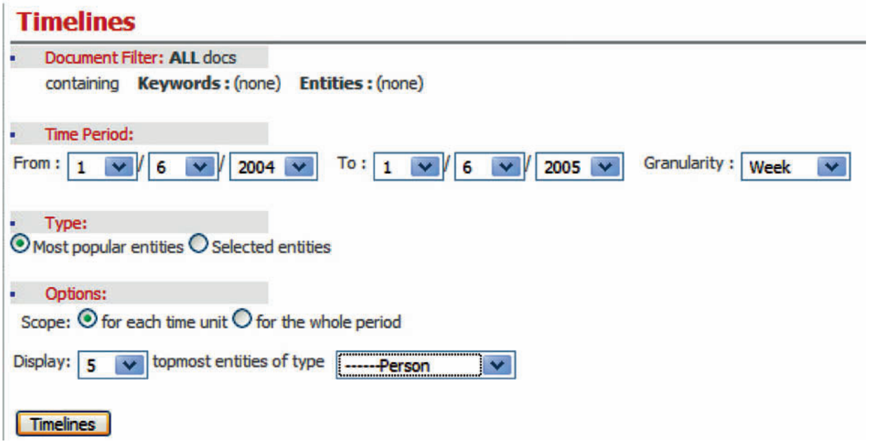
There are two different types of timelines that can be calculated: one can ask for the timeline of the most popular entities of a given class (Fig. 6.6) or view the-distribution over time of the occurrences of specific entities chosen by the user.

When requesting the most popular entities the user may choose their class, the number of entities to be covered by the timeline charts, and the time period granularity. One can also specify if the entities in interest are the ones that have overall top rank across the period or the ones that have a top rank only in some of the granularity time periods. The latter option would include in the timeline entities that are having their "fifteen minutes of fame", while the former would cover only such that are with more stable occurrence frequency throughout the featured period.

There is an option to see a timeline for a set of selected entities, which can be selected from columns like the ones in the CORE Search interface. The difference is that the selected entities do not restrict the document set but are only used as input parameters for the generation of the timeline chart.

In Fig. 6.6, the top 5 people for a given period are being requested, and at the same time their popularity is to be calculated only based on their appearance in the document set limited through the CORE Search interface (Fig. 6.4). The chart in Fig. 6.7 is the resulting timeline for these entities.

Navigation on the timeline chart itself is simple and intuitive. Positioning over a point from the chart shows information about the entity, the time period and the number of occurrences of the entity for that period and the given set of documents. Clicking on that point would redirect to the list of the documents that forms the respective part of the timeline along with a document distribution chart for the period of calculation (Fig. 6.8).



Timelines

Document Filter: ALL docs
containing Keywords : (none) Entities : (none)

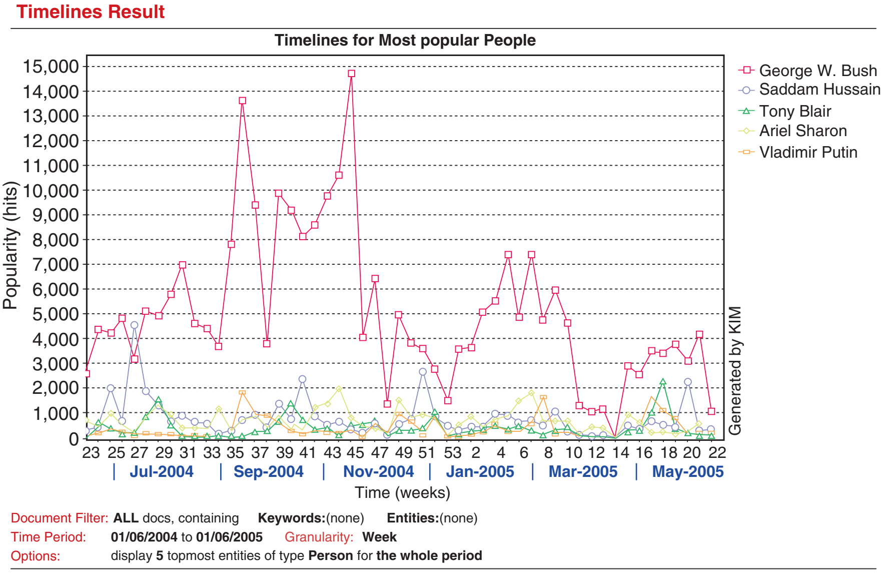
Time Period:
From : 1 / 6 / 2004 To : 1 / 6 / 2005 Granularity : Week

Type:
● Most popular entities ○ Selected entities

Options:
Scope: ● for each time unit ○ for the whole period

Display: 5 topmost entities of type ------Person

Fig. 6.6 Timeline request form – most popular people for a period

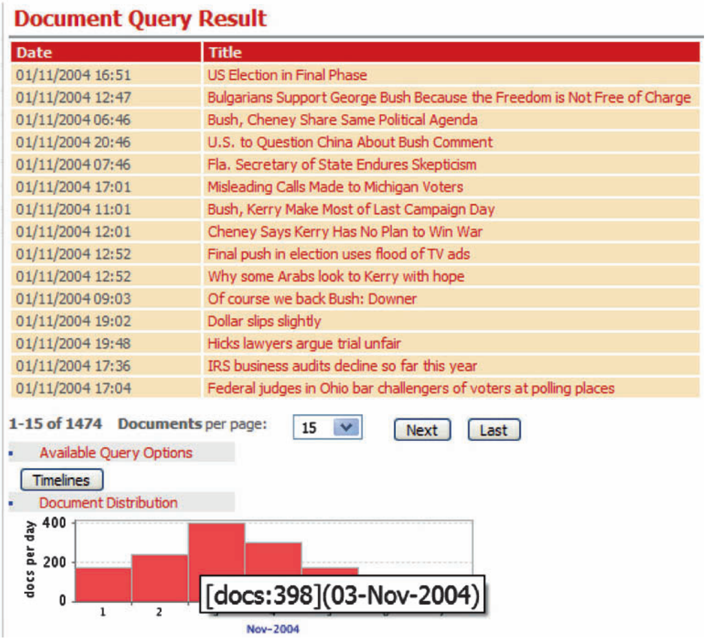


Document Filter: **ALL** docs, containing **Keywords:**(none) **Entities:**(none)

Time Period: **01/06/2004** to **01/06/2005** Granularity: **Week**

Options: display **5** topmost entities of type **Person** for **the whole period**

Fig. 6.7 Timelines result chart



Document Query Result

Date	Title
01/11/2004 16:51	US Election in Final Phase
01/11/2004 12:47	Bulgarians Support George Bush Because the Freedom is Not Free of Charge
01/11/2004 06:46	Bush, Cheney Share Same Political Agenda
01/11/2004 20:46	U.S. to Question China About Bush Comment
01/11/2004 07:46	Fla. Secretary of State Endures Skepticism
01/11/2004 17:01	Misleading Calls Made to Michigan Voters
01/11/2004 11:01	Bush, Kerry Make Most of Last Campaign Day
01/11/2004 12:01	Cheney Says Kerry Has No Plan to Win War
01/11/2004 12:52	Final push in election uses flood of TV ads
01/11/2004 12:52	Why some Arabs look to Kerry with hope
01/11/2004 09:03	Of course we back Bush: Downer
01/11/2004 19:02	Dollar slips slightly
01/11/2004 19:48	Hicks lawyers argue trial unfair
01/11/2004 17:36	IRS business audits decline so far this year
01/11/2004 17:04	Federal judges in Ohio bar challengers of voters at polling places

1-15 of 1474 Documents per page: 15

Fig. 6.8 Document result

The *document distribution* is displayed in an active chart which can be clicked to navigate deeper into smaller and smaller periods until the desired documents are found.

A screen similar to Fig. 6.8 would also appear if documents were required directly through the CORE Search interface. The documents content and associated document-level metadata and contained entities could also be seen if the user navigates further through clicking on a document title in the list.

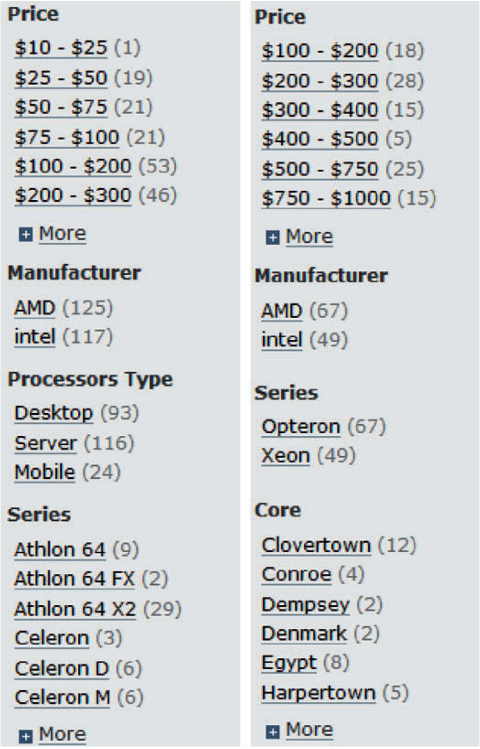
6.6.1 CORE Search Versus Standard Faceted Search

The standard notion for faceted search considers searching objects in multi-dimensional space, where the objects are characterized with respect to several independent aspects (facets of criteria). Each facet can be regarded as an attribute, which can be given a value for each object. For each facet (some of) the values are presented to the user, so, that she can further specify her needs (search query) by means of simple selection of one of the values.

Such example is the search for hardware items in the e-commerce portal http://www.newegg.com; the examples in Fig. 6.9 reflect its state on 3rd of Dec, 2007. When the user stars searching for a CPU, the she is initially presented with the following facets Price, Manufacturer, Processor Type, Series, and few others (on the left hand side of Fig. 6.8). The number of matching items is presented in brackets after each of the values. If the user selects “Server” from the Processor type list, she gets an updated faceted search pane, where the values for each of the facets are updated, “Processor Type” facet disappears and a new facet “Core” is presented (on the right hand side of Fig. 6.9).

The philosophy of faceted search is that the most relevant facets and values are presented, to help the user further constrain her search. This paradigm is similar to the Navigational searches, as the ones offered by taxonomy-based directories (e.g., http://www.dmoz.org). The difference is that the objects are classified against several independent criteria, rather than a single classification hierarchy. Further, the categories under each of these facets are automatically collected, rather than pre-determined.

CORE search is a special case of faceted search. In the standard faceted search a single object (in our case document) can have a single value for each of the facets. The model of CORE Search is much different than this. As presented on schema (1) of Fig. 6.10, in CORE we start with documents that are annotated with references to entities. Within single document there could be multiple annotations referring one and the same entity; each of them bears information irrelevant for CORE search (e.g., offsets of the references in the document). Schema (2) presents the structure of the CORE index – single Occurrence consolidates the information about the number of annotations, referring a concrete entity in a specific document. Finally, schema (3) represents the virtual facets, which are underlying CORE Search, as opposed to the standard facets, presented on schema (4) in Fig. 6.10. The main characteristics of CORE facets are as follows:



Price

$10 - $25 (1)
$25 - $50 (19)
$50 - $75 (21)
$75 - $100 (21)
$100 - $200 (53)
$200 - $300 (46)

⊞ More

Manufacturer

AMD (125)
intel (117)

Processors Type

Desktop (93)
Server (116)
Mobile (24)

Series

Athlon 64 (9)
Athlon 64 FX (2)
Athlon 64 X2 (29)
Celeron (3)
Celeron D (6)
Celeron M (6)

⊞ More

Price

$100 - $200 (18)
$200 - $300 (28)
$300 - $400 (15)
$400 - $500 (5)
$500 - $750 (25)
$750 - $1000 (15)

⊞ More

Manufacturer

AMD (67)
intel (49)

Series

Opteron (67)
Xeon (49)

Core

Clovertown (12)
Conroe (4)
Dempsey (2)
Denmark (2)
Egypt (8)
Harpertown (5)

⊞ More

Fig. 6.9 NewEgg.com: faceted search for CPUs (on the left), narrowed down for server CPUs (on the right)

- Single facet is created for each class of entities (e.g., CF_C1 and CF_C2);
- Each of these facets can have multiple values, namely entities of different classes, that were referred in the document (e.g., Entity2 and Entity3 are both values of CF_C2);
- Each facet value is actually a pair of a value and a number, which in our case represents the number of occurrences of the entity in the document.

In a sense, CORE facets represent "weighted" relationships between documents (or any other type of objects) and entities (or values).

We can represent each CORE facet and each standard facet as mathematical functions with the following description:

$$\text{CORE\_Facet} : O \to \{V, n\}^* \quad \text{Standard\_Facet} : O \to V$$

It becomes obvious that the data-model underlying CORE search is richer than the one behind the standard faceted search. Still, the philosophy for search and navigation remains the same:

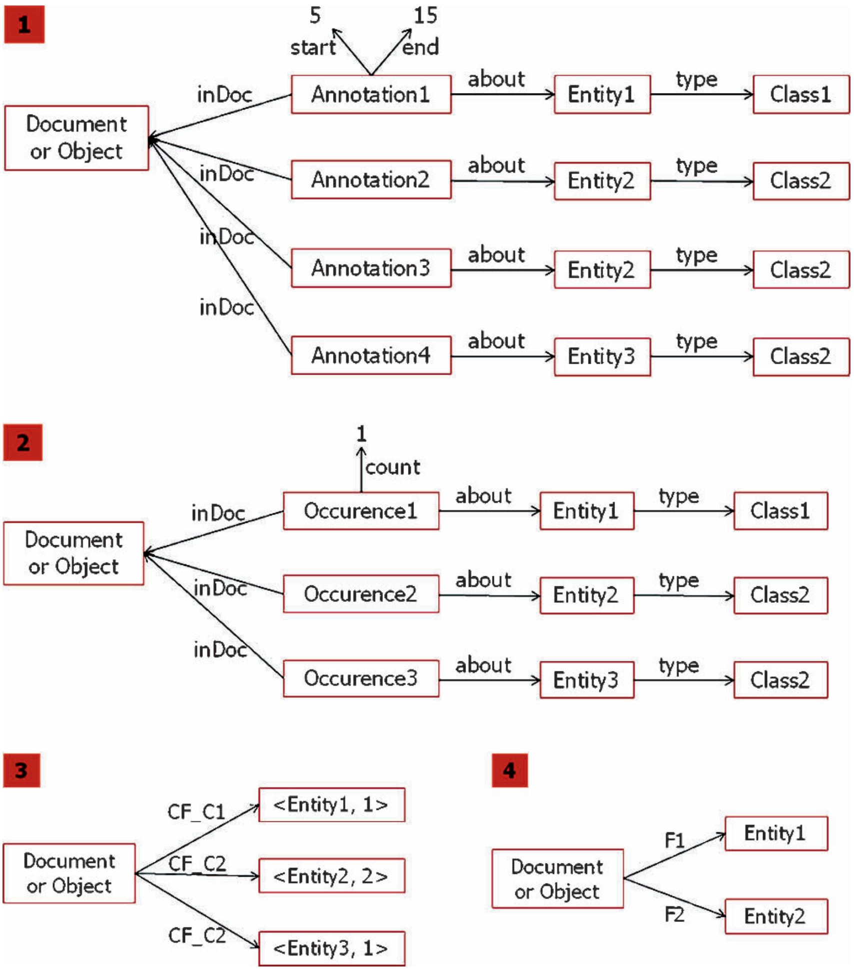


Fig. 6.10 Models: (**a**) Semantic annotations, (**b**) CORE index, (**c**) CORE Virtual Facets, and (**d**) Standard Facets

- Facet values are aggregated, across the dataset (the search space);
- The highest scoring values are presented to the user, as those have the highest chance to be subject of her interest;
- When the user puts some constraints, those affect the suggested values, so that the most informative ones are offered.

There are also some differences between CORE Search and the NewEgg's Guided search, taken as representative of the standard faceted search paradigm. At present, CORE search does not support automatic selection of the most relevant facets. This is partly because of the nature of facets in CORE – as those are virtually created on the basis of the classes of entities and can be customized by the user, so, it is not obvious that automatic modification of the sets of presented facets will be intuitive. The second difference is that in CORE Search we do not present the number of objects matching each facet value. The reason here is that we believe that ordering by popularity already provides to the user sufficient evidence regarding the selectivity of the different values. Thus, we prefer not to show this figures on the screen in order to display the optimal amount of information for the sake of better ergonomics.

6.7 Ongoing Work

Semantic Web applications, although gifted with obvious advantages, have a long way to go to match the scale introduced by years of development of relational database systems or the mass analysis of web content performed by contemporary search engines. The KIM Platform and its CORE module currently achieve real-time retrieval from about a million documents and a million of entity descriptions which should be enough in an intranet environment. However, the objectives are to reach levels of efficiency allowing us to cover a significant amount of the currently available web resources, and we will definitely continue in this direction.

In this regard, KIM already has a cluster architecture where its semantic analysis functionality, being the main bottleneck, is distributed over independent annotator components running (optionally) on different machines. Despite the obvious advantages of such architecture there are still some centralized synchronization points for our semantic repository, indexes and database. A work in distributing the semantic repository done by us or another group, allowing for effective intersections of query results across repositories, would bring the scale of such applications as KIM to a completely different level.

Regarding the CORE module it would be interesting to introduce means for defining the scope of the context of co-occurrence of entities to paragraphs, sets of documents or others.

The semantic repository of KIM and CORE DB are not tightly integrated at present. CORE DB “knows” the class hierarchy and very little about the semantics of the entities – their aliases and class memberships. This allows CORE Search to involve look up for organizations (i.e., entities of a sub-class of *Organization*) with specific strings in their aliases. However, formal typed relationships between entities cannot be involved, even when those are “known” in the semantic repository; such example would a statement that <*Org1, tradedOn, Exchange2*>. An idea we are currently considering is whether to introduce more of the semantics of the entities to the underlying database representation of the CORE module or to introduce a low-level integration of the RDBMS with the semantic repository.

At present CORE DB is using the semantic annotation, that is,the metadata generation and ontology population services provided by other modules of KIM. In the opposite direction, CORE DB can facilitate the metadata generation in two ways:

- Both class- and instance-level disambiguation of entity occurrences can be performed via comparison on the CORE profiles of the candidates and the context;
- Relation extraction is a tough IE task, so, when the confidence threshold is high, there are a relatively small number of extracted relations. This makes the descriptions of the new entities, extracted by KIM, to be relatively flat – there are not many attributes and relationships to other entities; the degree of connectivity in the semantic repository is getting lower as it grows through extracted knowledge. Associative relationships, extracted from CORE DB, can be added to the semantic repository. Although the types of the relations may not be easily determined, such extension would be useful, as the confidence in the existence of a relation could be high enough. A possible implementation is to add to the semantic repository the relationship between each entity and, say, the ten others it is most strongly associated with in CORE DB.

The current ranking schema in CORE is very simple – numbers of occurrences are counted without any normalization with respect to the size of the document or frequency of the entity (the TF part of TF/IDF). We are working on various statistical measurements and metrics for ranking, popularity and timelines calculation, which can be adjusted and changed dynamically.

Another interesting potential option that the future holds for us might be the cooperation with information extraction groups that would like to run their own domain specific extensions on our platform and specifically employing the CORE module, a thing we have already done in the past with KIM.

6.8 Conclusion

This paper presented a novel search paradigm based on popularity-ordered faceted search on top of semantic annotations. The technology used to demonstrate this was the CORE module of the KIM semantic annotation, indexing, and retrieval platform. The ideas behind it and their advantages to the traditional keyword search have been demonstrated on the conceptual level and through the CORE Search user interface. The essence of the approach is a specific indexing, performed on the basis of semantic annotation of text with respect to named entities and key-phrases allowing for semantic search through adaptation of standard probabilistic IR models.

The frequency of occurrence of an entity indicates its level of association with the given context and can also be considered as “popularity”. Various parts of documents or collections of such can be considered contexts or compound contexts. The co-occurring entities can be used to form a “statistical profile” of an entity, a document, or a group of such objects.

The paper also presented the beneficial symbiosis of a relational database and a semantic repository in the CORE module of KIM for the purpose of bi-directional indexing of the association of semantic annotations and contexts.

Acknowledgment This work was supported by the IST Programme of the European Community under SEKT (IST-IP-2003-506826).

References

Davies J, Fensel D, van Harmelen F (2003) Towards the Semantic Web, Wiley, UK.

Dill S, Eiron N, Gibson D, Gruhl D, Guha R, Jhingran A, Kanungo T, McCurley KS, Rajagopalan S, Tomkins A, Tomlin JA, Zienberer JY (2004) A case for automated large scale semantic annotation. Journal of Web Semantics, 1(1).

Guha R, McCool R (2003) Tap: a semantic web platform. Computer Networks, 42, 557–577.

Guha R, McCool R, Miller E (2003) Semantic Search. WWW 2003, Budapest, Hungary.

Kiryakov A, Ognyanov D, Manov D (2005a) OWLIM – a pragmatic semantic repository for owl. In Proceedings of International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2005), WISE 2005, 20 Nov, New York City, USA.

Kiryakov A, Popov B, Terziev I, Manov D, Ognyanoff D (2005b) Semantic annotation, indexing, and retrieval. Journal of Web Semantics, 2(1) http://www.websemanticsjournal.org/ps/pub/2005–10.

Landauer T, Dumais S (1997) A solution to Plato's problem: the latent semantic analysis theory of acquisition, induction and representation of knowledge. Psychological Review, 104(2).

Popov B, Kiryakov A, Ognyanoff D, Manov D, Kirilov A (2004) KIM – a semantic platform for information extraction and retrieval. Journal of Natural Language Engineering, 10(3–4), Sep 2004, 375–392, Cambridge University Press, Cambridge.

Rocha C, Schwabe D, de Aragao MP (2004) A Hybrid Approach for Searching in the Semantic Web. WWW 2004, New York, USA.

Terziev I, Kiryakov A, Manov D (2005) PROTON Ontology (PROTo ONtology). SEKT Project. http://proton.semanticweb.org/.

Chapter 7
Semantically Enhanced Search and Browse

Alistair Duke and Jörg Heizmann

Abstract Squirrel, a search and browse tool that provides access to semantically annotated data is described. The tool offers a hybrid approach to search allowing the user to enter simple search terms and then refine their search or browse to related material through the presentation of appropriate metadata. Squirrel builds upon and integrates a number of the semantic technology components described elsewhere in the book. These include machine learning and information extraction components which generate, extract and manage semantic metadata contained within and about textual documents at index time. A number of run-time components have also been integrated to deliver an enhanced user experience such as natural language generation which provides natural language summaries of knowledge held in formal (ontological) structures; device independence which allows the tool to be run on multiple devices; result consolidation which presents the most relevant textual content of result documents rather than a simple list of results; and a natural language interface which translates natural language queries into structured queries formulated with respect to a given ontology.

7.1 Introduction

This chapter describes Squirrel, a search and browse tool that provides access to semantically annotated data. Search is seen as a key application that can benefit from semantic technology with improvements to recall and precision versus conventional Information Retrieval techniques. Squirrel builds upon and integrates a number of the semantic technology components described elsewhere in this book. These include machine learning and information extraction components which generate, extract and manage semantic metadata contained within and about textual documents at index time. A number of run-time components have also been

A. Duke (✉)
Next Generation Web group, BT Research, Ipswich, IP5 3RE, UK
e-mail: alistair.duke@bt.com

J. Davies et al. (eds.), *Semantic Knowledge Management*,

integrated to deliver an enhanced user experience such as natural language generation which provides natural language summaries of knowledge held in formal (ontological) structures; device independence which allows the tool to be run on multiple devices; result consolidation which presents the most relevant textual content of result documents rather than a simple list of results; and a natural language interface which translates natural language queries into structured queries formulated with respect to a given ontology.

A hybrid approach has been adopted to provide a balance between the speed and ease of use of simple free text search and the power of semantic search. Users are able to enter simple search terms and then refine their search or browse to related material through the presentation of appropriate metadata. Alternatively, users can express a query using natural language which is analysed and converted into a semantic query.

The chapter is structured as follows. The next section introduces a scenario which illustrates both the requirements and benefits of enhanced search and browse. Section 7.3 describes the salient features of the supporting components that comprise Squirrel and presents an architecture for the system. In Sect. 7.4 further detail is provided about the user experience with a description of the important features of the interface. Section 7.5 discusses related work and outstanding issues whilst we conclude in Sect. 7.6 with a view of the way forward.

7.2 Scenario

Squirrel has been trialled in a case study which is developing an improved Digital Library for BT, as discussed elsewhere in this volume. The following scenario describes a Digital Library user carrying out a knowledge seeking task, making use of the features provided by Squirrel and its supporting components.

The user has an initial high level goal to seek information about the field of Home Health Care and has little idea about what is contained in the library which might be of use. He first enters “Home Health Care” into the search box and hits the “Go!” button. The first result screen includes a summary of the sorts of resources that have been found that might be able to meet his needs. These include the textual resources from the library that is that there are a number of journal articles, conference papers, periodicals and web pages (shared by library users and indexed as library resources) that match his query. In addition, there a number of library topics that also match his query including one which is itself called “Home Health Care”. Further matches include a number of organisations from a domain knowledge base whose description includes the search term.

In addition to this summary, the user is also presented with the top ranked textual resources (in typical search engine style). This simple list of documents is augmented with the ability to refine the search based on the properties of the documents in the result set, including a hierarchical display of the topics of the results documents, date of publish, author name, etc.

Our user decides that they are interested in current affairs related to their topic rather than scientific papers. He chooses to refine the search to the periodicals section of the library. The topic hierarchy is rebuilt to accommodate the documents in the refined results set. Furthermore, our user is interested in the financial aspects of home health care and notices that a topic called "Economic Conditions" has been presented. He then chooses this topic to further refine their results set.

A short taster of each result together with its title and other metadata is presented. This is enhanced by highlighting the named entities that have been recognised in the text such as people, organisation, locations, etc. Our user places their mouse on one such highlighted term, "United Health Products" and is shown a pop-up which tells him that this is a company. Intrigued about the company, he clicks on the term and is taken to a screen giving further information about the company including a natural language summary about the company as well as related entities such as the people who work for it. There is also a link to all of the documents where this company (including any of its aliases such as UHP, etc.) is mentioned. Our user clicks on this link but then decides he would rather view the results as a consolidated summary rather than a list of discrete results. In this view the most relevant portions of the result documents are shown to the user meaning he can quickly view the available material rather than having to navigate to several different pages. The user again refines the contents of the summary by selecting one of more of the topics that have been assigned to the presented material.

Our user is also able to use the natural language interface by entering a natural language query. The addition of a question mark identifies it as such to Squirrel. Thus our user might commence their search by entering the query "Which articles are about 'Home Health Care'?" followed by "Which periodicals are about Economic Conditions?"

7.3 Architecture

Squirrel integrates a number of components developed in the SEKT project. This section briefly describes the features of the components and then presents an architecture illustrating how they have been integrated.

7.3.1 Proton

PROTON is a lightweight general purpose ontology developed in SEKT which includes a module specific to the knowledge management domain. PROTON is used by SEKT components for metadata generation and as a basis for knowledge modelling and integration.

A world knowledge base (originally developed as part of the KIM platform (Popov et al. 2004)) has been expressed in PROTON. The knowledge base is comprised of more than 2,00,000 entities. These are gathered semi-automatically from a range of high quality public data sources. It includes around 36,000 locations, 1,40,000 companies and other organisations and the world's pre-eminent politicians, businesspeople and technologists.

7.3.2 Full-Text Index

The SEKT Integration Platform (SIP) provides a full-text indexing facility which is used to index the various forms of textual resource including the labels and literal properties of the ontological instances and classes. This allows metadata, such as authors' names, topic names, knowledge base entities, to be discovered and presented to the user as a way to refine their search or as an alternative way to find documents, since the metadata is always related to a set of documents in some way. The index of ontological instance will, given a search term, provide a set of matching URIs which can then be used to query the ontology repository to extract further details of the concept they refer to.

7.3.3 KAON2

Squirrel uses KAON2 (described in Chap. 2) as an ontology layer, providing access to ontological instances and reasoning support. KAON2's facility to extract ontology instances from relational databases via a mapping function allows a greatl increase in the number of instances accessed (versus a file based approach where all instances must be stored in memory). KAON2 also supports the Description Logic-safe subset of the Semantic Web Rule Language (SWRL) which allows, through formal reasoning, knowledge to be presented against concepts that goes beyond that provided by the structure of the ontology. For example, one of the attributes displayed in the document presentation is "Organisation". This is not an attribute of a document in the PROTON ontology; however, affiliation is an attribute of the Author concept and has the range "Organisation". As a result, a rule was introduced into the ontology to infer that the organisation responsible for a document is the affiliation of its lead author. Other rules have also been introduced including one allowing the "collaborates with" attribute of an author to be inferred by querying the ontology for all the papers where the author in question has co-authored with others and then inferring that these authors are his or her collaborators.

7.3.4 Natural Language Generation

Natural Language Generation (NLG), described in the next chapter, is used by Squirrel to provide natural language summaries on ontological entities. The PROTON world knowledge base contains a rich source of knowledge about entities, which may be mentioned in the documents indexed. When a Squirrel search is carried out and certain people, organisations or other entities occur in the result set, the user is presented with a summary of information regarding those entities. In addition, document results can be enhanced with brief descriptions of key entities that occur in the text. For example, the knowledge base contains company related metadata, which can be presented to the user as natural language. An example is shown further below in Fig. 7.4.

7.3.5 DIWAF

The SEKT Device Independence Web Application Framework is a server side application, which provides a framework for presenting structured data to the user (Glover and Davies 2005). The framework does not use a mark-up language to annotate the data. Instead, it makes use of templates, which are "filled" with data, rather like a mail merge in a word processor. These templates allow the selection, repetition and rearrangement of data, interspersed with static text. The framework can select different templates according to the target device, which present the same data in different ways.

Squirrel has been built as a DIWAF application. This enables the applications to be easily adapted to alternative device or context requirements (such as different languages). Currently, templates are available to deliver the application to users via a WAP-enabled mobile device, via a PALM PDA and via a standard web browser.

7.3.6 Ontology Generation

Ontology Generation (OG) (see Chap. 2) can automatically identify an ontology from the content of the documents and then classify the documents according to the ontology (Fortuna et al. 2005). In SEKT this process is generally carried out at index-time which allows the knowledge engineer to modify the generated ontology if required. The process results in hierarchical topic ontology and a set of document classifications into that ontology (in accordance with PROTON). These results are used by Squirrel to present the topic hierarchy thus allowing the user to refine their search by topic. The OG component can also be used by Squirrel to generate clusters of documents at query-time. This could be used in the general case where some or all of the documents being searched have not been classified at index-time.

7.3.7 User Profile Construction and Usage

The PROTON ontology includes concepts allowing the interests of users to be stored and modelled as a profile. Such a profile is an important step in providing relevant knowledge to people and is used within Squirrel to personalise the search experience. The profile allows the search tool to predict what the user is interested in based upon their observed interests. Profiles can be manually or automatically created. An automatic approach places less burden on the user than a manual one but relies on the assumption that the techniques used to collect the profile are accurate (which is why a combination of automatic and manual approaches is often adopted). The approach adopted in SEKT has been to observe the web browsing habits of the individual users and to extract the topics associated with the pages that have been accessed. A level of interest for each of the topics is also determined.

In PROTON, the profile consists of an expression of both long-term and short-term interest in topics from a PROTON topic ontology. The Squirrel search tool makes use of these profiles to modify the way the results are presented to the user, providing context to the user's search query. When ranking documents, the topics of the documents in the result set can be considered against the level of interest in topics in the user's profile. Thus documents with a topic recorded in the profile with a strong short-term interest would be presented first. Topics are related to other topics (either by a sub or super topic or some other weaker relation) so these relationships can also be considered to support the application of profiles.

7.3.8 Massive Semantic Annotation

Squirrel makes use of the results of a Massive Semantic Annotation (see Chap. 6) process, carried out at index time. The function uses KIM (Popov et al. 2004), which employs GATE's ontology-based information extraction module to identify named entities in texts. For each of these it carries out instance disambiguation; that is, it either identifies an existing PROTON instance of which the entity is an occurrence or creates a new instance. An annotation (which consists of the instance URI, the document it occurs in and its location within the text) is stored using KIM's highly scalable (due to the very large number of annotations) knowledge store.

In order to support user responsive entity presentation and browsing, Squirrel makes use of the OWLIM semantic repository (Kiryakov et al. 2005). OWLIM is considered to be the fastest OWL repository available. The dual repository approach adopted by Squirrel is justified by the benefits that each repository provides. KAON2 offers relational database mapping and rule support whilst the scalability and speed of OWLIM are suited to handling the volume of data resultant from the semantic annotation process.

7.3.9 Text Segmentation

During the indexing process, documents are segmented at topical boundaries in order to support the presentation of consolidated results to the user. This uses a C99 segmenter (Choi 2000) enhanced to consider the distribution of named entities and key phrases identified in the text in addition to the distribution of words. Following segmentation the subdocuments are classified using the TextGarden classifier described above. These classifications are used in two ways by Squirrel. Firstly, to allow the user to refine their summaries based upon topics of interest and secondly to reorder the presentation of the summary based upon the topics in the user's profile.

7.3.10 Natural Language Querying

The ORAKEL (Cimiano et al. 2007) natural language interface translates natural language queries to structured queries formulated with respect to a given ontology. This translation essentially relies on a compositional-semantic interpretation of the question guided by two lexica: a domain-specific and a domain-independent one. As the name suggests, the domain-independent lexicon is encoded into the system a priori and captures the domain-independent meaning of prepositions, determiners, question pronouns etc., which typically remain constant across domains. The domain-specific lexicon needs to be created by an lexicon engineer who is assumed to create the lexicon in various cycles. In fact, ORAKEL builds on an iterative lexicon development cycle in which the lexicon is constantly updated by the lexicon engineer on the basis of the questions the system failed to answer so far. The end users are then able to directly interact with the Digital Library data using natural language questions, which are translated into SPARQL queries. The underlying mechanism, however, is hidden from the users – the only thing users need to do is to input the query just as their normal questions and then they get the result from the portal. The obvious benefit of natural language querying is thus that users can pose fairly complex queries to the system without having to learn a formal language such as SQL or SPARQL. Of course, as for keyword-based retrieval, they will need to get used to the vocabulary understood by the system.

Figure 7.1 gives an overview of the ORAKEL system, which has been designed in a flexible manner allowing to easily replace the system's target query language. As the architecture described here relies on the KAON2 system as inference engine and knowledge repository, ORAKEL was adapted to generate SPARQL queries which are supported by KAON2. Further, a lexicon was generated for the Proton ontology, which specifies the possible lexical representations of the ontology elements in the users' queries. This lexicon was constructed in three iterations: one initial iteration of 6 h and 2 follow-up iterations of 30 min each in which the questions not answered by the system were examined.

7.3.11 Architecture

The components described above have been integrated as shown in the architecture in Fig. 7.2. The figure shows a complete end-to-end architecture for indexing and querying.

The sources of textual data, shown at the bottom of the Fig. 7.2, are stored together with their associated metadata in a database. These sources may be (for example) records of bibliographic data (as in the SEKT Digital Library case

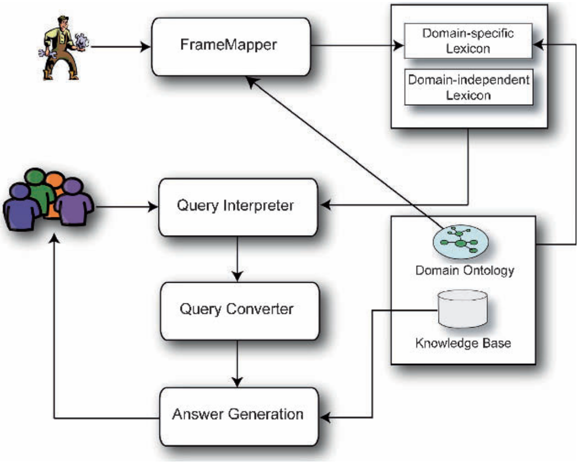


Fig. 7.1 Overview of ORAKEL natural language system

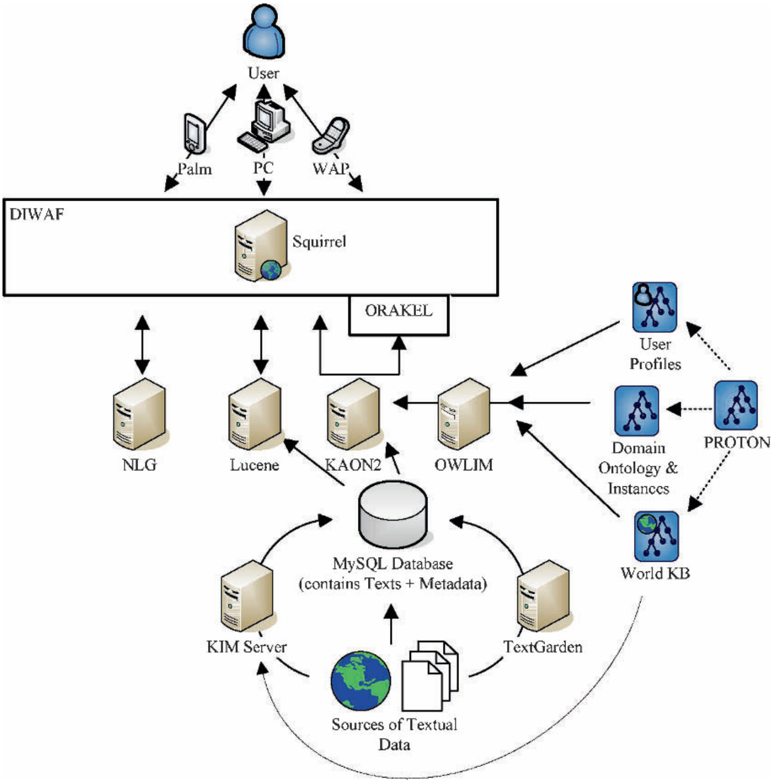


Fig. 7.2 Squirrel architecture

study), webpages identified by a crawler, or legal judgments (as in the SEKT Legal case study). At this stage, the metadata can be augmented by one or more entity extraction or classification components.

Once these steps have been carried out, the contents of the database can then be indexed by SIP as described in Sect. 7.3.2.

A parallel activity to free-text indexing is to load into KAON2 the contents of the database, the PROTON ontology, its world knowledge base and user profiles.

User queries are first passed to SIP in order to retrieve a set of matching documents and entities. Squirrel examines the results of this query and sorts them into entity results and document results. An entity result consists of the URI of the entity. For each of these URIs, Squirrel queries KAON2 via the programming API to determine the type (PROTON class) of the entity and the properties associated with the type for which the entity may have values. Squirrel can then further query KAON2 to determine the values of these properties (if indeed there are any).

Following this, Squirrel is able to present the "meta-result" to the user. This is a summary of the types and number of entities it has found and the types and number of documents (the type of document will typically indicate its source (e.g., webpage, RSS feed, journal article, conference paper). The user is presented with a set of options, which allow them to determine which path they wish to take for their search. The meta-result is accompanied by the default initial result display, which is a list of documents (of any type) that satisfy the query. The first 10 documents from this list are shown, ranked against the query. If a user profile is available, the ranking is adjusted to reflect the topics of interest in the profile. The user interface and the various options provided by the meta-results are described further, with examples, in Sect. 7.4.

When displaying document results, Squirrel queries the database to determine if any annotations are associated with each document. These can then be integrated in the display as highlighted portions of text, which allow the user to "mouse over" to gain further details such as PROTON class and main alias. The user is also given the ability to refine the document set by topic.

When displaying entity results, Squirrel can call the NLG web service (described in Sect. 7.3.4) to generate a summary for particular entities.

The natural language interface supported by ORAKEL is utilised when users append a query with a question mark. As described in Sect. 7.3.10, ORAKEL interacts with the data via KAON2 and its SPARQL interface.

7.4 Interface Description

7.4.1 Initial Search

Users are permitted to enter terms into a text box to commence their search. This initially simplistic approach was chosen based on the fact that users are likely to be comfortable with it due to experience with traditional search engines. If they wish, the user can specify which type of resource (from a configurable list) they are looking for (e.g., publications, web articles, people, organisations) although the default is to search in all of them.

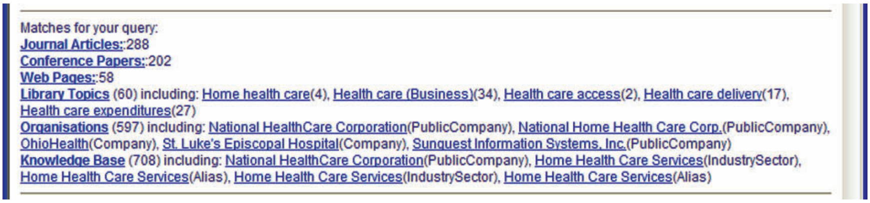
The first task Squirrel carries out after the user submits a search is to call the SIP index and then use KAON2 to look up further details about the results, which may be textual resources or ontological entities. In addition to instance data, the labels of ontological classes are also indexed. This allows users to discover classes and then discover the corresponding instances and the documents associated with them without knowing the names of any of the instances; for example, a search for "Airline Industry" would match the "Airline" class in PROTON. Selecting this would then allow the user to browse to instances of the class where they can then navigate to the documents where those instances are mentioned. This is an important feature since with no prior knowledge of the domain it would be impossible to find these documents using a traditional search engine.

Textual items can by separated by their type, for example, Web Article, Conference Paper, Book, etc. Squirrel is then able to build the meta-result page based upon the textual content items and ontological instances that have been returned.

7.4.2 Meta-Result

The meta-result page is intended to allow the user to quickly focus their search as required and to disambiguate their query if appropriate. The page presents the different types of result that have been found and how many of each type. In order not to introduce unnecessary overhead on the user, the meta-result page also lists a default set of results allowing the user to immediately see those results deemed most relevant to their query by purely statistical means.

The meta-result for the "home health care" query is shown in Fig. 7.3 under the sub-heading "Matches for your query". The first items in the list are the document classes. Following this is a set of matching topics from the topic ontology. In each case, the number in brackets is the number of documents attributed to each class or topic. Following the topics, a list of other matching entities is shown. The first five



Matches for your query:
Journal Articles::288
Conference Papers::202
Web Pages::58
Library Topics (60) including: Home health care(4), Health care (Business)(34), Health care access(2), Health care delivery(17), Health care expenditures(27)
Organisations (597) including: National HealthCare Corporation(PublicCompany), National Home Health Care Corp.(PublicCompany), OhioHealth(Company), St. Luke's Episcopal Hospital(Company), Sunquest Information Systems, Inc.(PublicCompany)
Knowledge Base (708) including: National HealthCare Corporation(PublicCompany), Home Health Care Services(IndustrySector), Home Health Care Services(Alias), Home Health Care Services(IndustrySector), Home Health Care Services(Alias)

Fig. 7.3 Meta-result

matching entities are also shown allowing the user to click the link to go straight to the entity display page for these. Alternatively they can choose to view the complete list of items. Squirrel can be configured to separate out particular entity types (as is the case with topics and organisations as shown in Fig. 7.3) and display them on their own line in the meta-result.

The returned documents can be ranked according to the user's profile if one has been specified. The algorithm to perform this ranking checks to see which documents are attributed to topics that exist in the profile and then adjusts the ranking using the degree of interest for those topics. For each document, the degree of relevance provided by SIP (which is between 0 and 1) is added to the cumulative degree of interest from the topics in the profile that are also attributed to the document. The documents are then re-ranked based upon these new relevance figures and displayed to the user. The algorithm can be formulated as shown in the following equation.

$$R = r + \Sigma_{\mathrm{i=1..k.}} \begin{cases} w(t_i) & t \in U \\ 0 & t \notin U \end{cases} \tag{7.1}$$

Where R is the degree of relevance for a document and r is the (statistical) degree of relevance provided by SIP. A topic attributed to the document is denoted by t_i where i is the index of a topic in the set of attributed topics. $w(t_i)$ denotes the level of interest in the topic t_i if t is a member of the set of topics in the user's profile, U. If the topic is not in the profile then no adjustment is made.

The algorithm attempts to maintain a balance between results with a high relevance ranking mainly due to the SIP result and those that are principally deemed to be of interest according to the profile. Thus, if the user has just switched their focus, the results should not skew too far in favour of the profile which might be slightly biased towards their previous focus.

7.4.3 Refining by Topic

Alongside the results, the user is presented with the topics associated with the documents in the result set. Not all topics are shown here since in the worst case where each document has many distinct topics the list of topics presented to the user would be unwieldy. Instead an algorithm takes the list of topics that are associated with the collection of documents in the result set, and generates a new list of topics representing the narrowest "common ancestors" of the documents' topics.

The algorithm works by repeatedly applying two phases – an "extend" phase, which generates parents of the current topic list, and a "sweep" phase, which merges together identical parents. The algorithm terminates when there is only one topic in the list (a single common ancestor), or when there are no more parents. The set of narrowest "common ancestors" for the whole result set generally provides a more manageable topic list from which the user can refine the results.

Having selected a topic and viewed the subset of documents from the result set, the user can switch to an entity view for the topic. The user can also reach this view by selecting a topic from the meta-result section. Instead of showing the documents of the topic, the meta-data for the topic is shown, which includes the broader, narrower and related topics. This allows the user to browse around the topic ontology. Each topic is shown with the number of documents it contains in brackets after it. Two links to document results are also shown. The first takes the user to a list of documents that have been attributed to the topic itself. The second takes the user to a list of documents that include all subtopics of the current topic. The layout of entity views in Squirrel are defined by templates. This allows the administrator to determine what meta-data is shown and what is not. The use of these templates is discussed further in Sect. 7.4.6 where a "Company" entity view is described.

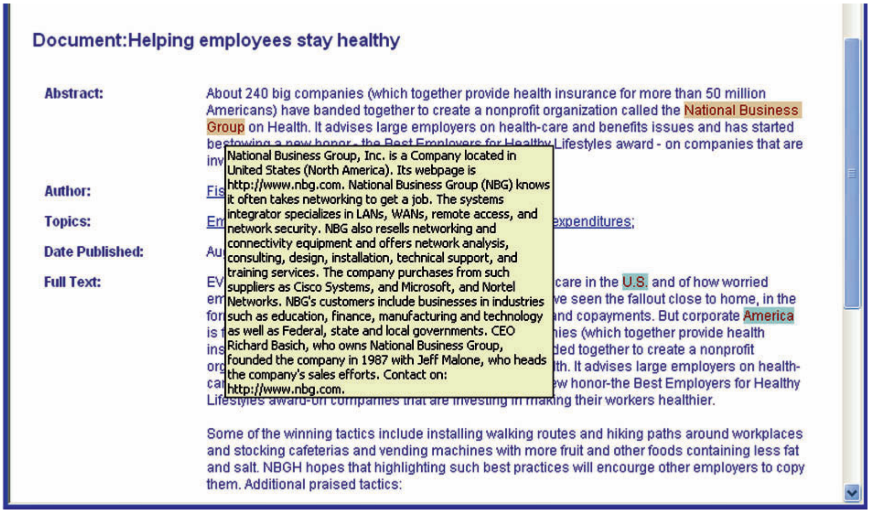
7.4.4 Attribute-Based Refinement

Each document result list has a link, which opens a refiner window. This allows the user to refine the results based upon the associated metadata. The metadata shown and the manner in which it is displayed are configurable through the use of entity type specific templates that are configured by an administrator or knowledge engineer. Documents can be refined by the user based upon their authors, date of publication, etc. The approach adopted has been to allow the user to enter free text into the attribute boxes and to re-run the query with the additional constraints. An alternative would be to list possible values in the result set. However, the potential size of this list is large and is difficult to present to the user in a manageable way. The disadvantage of the free-text approach is that the user can reduce the result set to zero by introducing an unsatisfiable constraint – which is obviously undesirable. Squirrel attempts to address this by quickly showing the user the number of results as soon as the constraints have been set. The user can then modify them before asking Squirrel to build the result page.

7.4.5 Document View

The user selects a document from the reduced result set, which takes them to a view of the document itself. This shows the meta-data and text associated with the document and also a link to the source page if appropriate – as is the case with web-pages. Since web-pages are published externally with specific formatting data, the text of the page is extracted at index-time. Any semantic mark-up that is applied at this stage can then be shown on the extracted text at query-time. However, the user should always be given the option to navigate to the page in its original format. Figure 7.4 shows a screenshot of the document view.

The document view also shows the whole document abstract, marked-up with entity annotations. "Mousing over" these entities provides the user with further



Document:Helping employees stay healthy

Abstract: About 240 big companies (which together provide health insurance for more than 50 million Americans) have banded together to create a nonprofit organization called the National Business Group on Health. It advises large employers on health-care and benefits issues and has started bestowing a new honor - the Best Employers for Healthy Lifestyles award - on companies that are
inv

Author: Fis

Topics: Em expenditures;

Date Published: Au

Full Text:
EV care in the U.S. and of how worried
em ve seen the fallout close to home, in the
for nd copayments. But corporate America
is f nies (which together provide health
ins ded together to create a nonprofit
org lth. It advises large employers on health-
ca w honor-the Best Employers for Healthy
Lifestyles award-on companies that are investing in making their workers healthier.

Some of the winning tactics include installing walking routes and hiking paths around workplaces and stocking cafeterias and vending machines with more fruit and other foods containing less fat and salt. NBGH hopes that highlighting such best practices will encourge other employers to copy them. Additional praised tactics:

National Business Group, Inc. is a Company located in United States (North America). Its webpage is http://www.nbg.com. National Business Group (NBG) knows it often takes networking to get a job. The systems integrator specializes in LANs, WANs, remote access, and network security. NBG also resells networking and connectivity equipment and offers network analysis, consulting, design, installation, technical support, and training services. The company purchases from such suppliers as Cisco Systems, and Microsoft, and Nortel Networks. NBG's customers include businesses in industries such as education, finance, manufacturing and technology as well as Federal, state and local governments. CEO Richard Basich, who owns National Business Group, founded the company in 1987 with Jeff Malone, who heads the company's sales efforts. Contact on: http://www.nbg.com.

Fig. 7.4 Document view

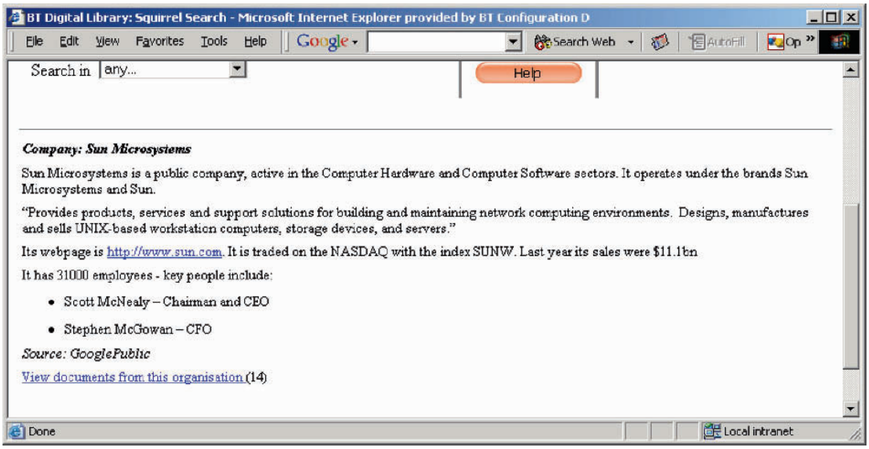
information about the entity extracted from the ontology using the NLG component. Clicking on the entity itself takes the user to the entity view.

7.4.6 Entity View

The entity view for “Sun Microsystems” is shown in Fig. 7.5. It includes a summary generated by the NLG Web Service described in Sect. 7.3.4. The summary displays information related not only to the entity itself but also information about related entities such as people who hold job roles with the company. This avoids users having to browse around the various entities in the ontology that hold relevant information about the entity in question. The relationship between people and companies is made through a third concept called JobPosition. Users would have to browse through this concept in order to find the name of the person in question.

7.4.7 Consolidated Results

Users can choose to view results as a consolidated summary of the most relevant parts of documents rather than a discrete list of results. This view is appropriate when a user is seeking to gain a wider view of the available material rather than looking for a specific document. The view allows them to read or scan the material without having to navigate to multiple results. Figure 7.6 shows a screenshot of a summary for a query for “Hurricane Katrina”. For each subdocument in the summary the user is able to view the title and source of the parent document, the


BT Digital Library: Squirrel Search - Microsoft Internet Explorer provided by BT Configuration D
File Edit View Favorites Tools Help
Google
Search Web
AutoFill
Op
Search in any...
Help
Done
Local intranet

Company: Sun Microsystems

Sun Microsystems is a public company, active in the Computer Hardware and Computer Software sectors. It operates under the brands Sun Microsystems and Sun.

"Provides products, services and support solutions for building and maintaining network computing environments. Designs, manufactures and sells UNIX-based workstation computers, storage devices, and servers."

Its webpage is http://www.sun.com. It is traded on the NASDAQ with the index SUNW. Last year its sales were $11.1bn

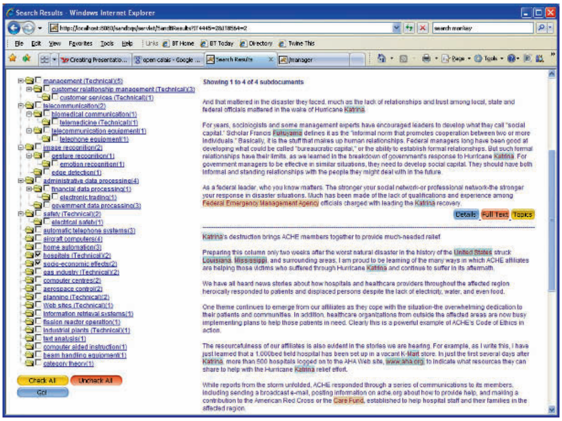
It has 31000 employees - key people include:

- Scott McNealy – Chairman and CEO
- Stephen McGowan – CFO

Source: GooglePublic

View documents from this organisation (14)

Fig. 7.5 Company entity view


Search Results - Windows Internet Explorer
http://localhost:8080/sandbqs/servlet/SandBResults?IT4445=28J18564=2
search monkey
File Edit View Favorites Tools Help
Links BT Home BT Today Directory Twine This
Creating Presentatio...
open calais - Google ...
Search Results
/manager
Page Tools

- □ management (Technical)(5)
  - □ customer relationship management (Technical)(3)
    - □ customer services (Technical)(1)
- □ telecommunication(2)
  - □ biomedical communication(1)
    - □ telemedicine (Technical)(1)
  - □ telecommunication equipment(1)
    - □ telephone equipment(1)
- □ image recognition(2)
  - □ gesture recognition(1)
    - □ emotion recognition(1)
  - □ edge detection(1)
- □ administrative data processing(4)
  - □ financial data processing(1)
    - □ electronic trading(1)
  - □ government data processing(3)
- □ safety (Technical)(2)
  - □ electrical safety(1)
- □ automatic telephone systems(3)
- □ aircraft computers(4)
- □ home automation(3)
- ☑ hospitals (Technical)(2)
- ☑ socio-economic effects(2)
- □ gas industry (Technical)(2)
- □ computer centres(2)
- □ aerospace control(2)
- □ planning (Technical)(2)
- □ Web sites (Technical)(1)
- □ information retrieval systems(1)
- □ fission reactor operation(1)
- □ industrial plants (Technical)(1)
- □ text analysis(1)
- □ computer aided instruction(1)
- □ beam handling equipment(1)
- □ category theory(1)

Check All Uncheck All
Go!

Showing 1 to 4 of 4 subdocuments

And that mattered in the disaster they faced, much as the lack of relationships and trust among local, state and federal officials mattered in the wake of Hurricane Katrina.

For years, sociologists and some management experts have encouraged leaders to develop what they call "social capital." Scholar Francis Fukuyama defines it as the "informal norm that promotes cooperation between two or more individuals." Basically, it is the stuff that makes up human relationships. Federal managers long have been good at developing what could be called "bureaucratic capital," or the ability to establish formal relationships. But such formal relationships have their limits, as we learned in the breakdown of government's response to Hurricane Katrina. For government managers to be effective in similar situations, they need to develop social capital. They should have both informal and standing relationships with the people they might deal with in the future.

As a federal leader, who you know matters. The stronger your social network-or professional network-the stronger your response in disaster situations. Much has been made of the lack of qualifications and experience among Federal Emergency Management Agency officials charged with leading the Katrina recovery.

Details Full Text Topics

Katrina's destruction brings ACHE members together to provide much-needed relief.

Preparing this column only two weeks after the worst natural disaster in the history of the United States struck Louisiana, Mississippi, and surrounding areas, I am proud to be learning of the many ways in which ACHE affiliates are helping those victims who suffered through Hurricane Katrina and continue to suffer in its aftermath.

We have all heard news stories about how hospitals and healthcare providers throughout the affected region heroically responded to patients and displaced persons despite the lack of electricity, water, and even food.

One theme continues to emerge from our affiliates as they cope with the situation-the overwhelming dedication to their patients and communities. In addition, healthcare organizations from outside the affected areas are now busy implementing plans to help those patients in need. Clearly this is a powerful example of ACHE's Code of Ethics in action.

The resourcefulness of our affiliates is also evident in the stories we are hearing. For example, as I write this, I have just learned that a 1,000bed field hospital has been set up in a vacant K-Mart store. In just the first several days after Katrina, more than 500 hospitals logged on to the AHA Web site, www.aha.org, to indicate what resources they can share to help with the Hurricane Katrina relief effort.

While reports from the storm unfolded, ACHE responded through a series of communications to its members, including sending a broadcast e-mail, posting information on ache.org about how to provide help, and making a contribution to the American Red Cross or the Care Fund, established to help hospital staff and their families in the affected region.

Fig. 7.6 Consolidated results

topics into which the subdocument text has been classified or navigate to the full text of the document. A topic tree is built which includes a checkbox for each topic. These allow the user to refine the summary content by checking or unchecking the appropriate boxes.

7.4.8 Natural Language Interface

The prototype system deployed on the BT Digital Library contained (1) question answering functionality as realised by the ORAKEL system, (2) ontology learning functionality to discover new topics as implemented in the Text2Onto system as well as (3) automatic text classification functionality directly integrated into the KAON2 inference engine as a built-in predicate. The resulting question answering prototype thus not only allows users to ask questions about existing metadata in the BT Digital Library, but also about topics found by Text2Onto as well as to invoke the automatic text classification system at runtime. The net results are that a variety of queries about authors, topics etc. of documents could be answered successfully against the domain ontology and database. Examples of such questions are:

- What conference papers did Davies write?
- Which conference papers do you know?
- Who wrote which document?
- Who wrote "The future of web services"?
- Who wrote about "Knowledge Management"?
- What article deals with Photography?
- Which journal articles were written by whom?
- What are the topics of "The future of web services"?
- Which conference papers were classified as religion?
- Which articles are about "Intellectual Capital"?
- What articles were written by Lent and Swami?
- Who is the author of a document that talks about which concept?
- Who wrote which articles about what?
- Which documents are about F-Logic and Insurance?

Furthermore, ORAKEL was also modified to process quoted text by matching it against a text index of the metadata in the database using a special purpose predicate match. The question "*What articles are about 'Intellectual Capital'*?"? is translated into the following SPARQL query:

```
SELECT ?w16 WHERE {
?w16 rdf:type
<http://proton.semanticweb.org/2005/04/protonu#Article>
.
?w16 <http://proton.semanticweb.org/2005/04/protont#hasSubject>
?x17.
?x17 rdfs:label x18.
match(x18,"Intellectual Capital")
}
```

The application of our prototype enhances the access to BT's Digital Library showing the integration of data and sources using an inference engine, precise question answering exploitation of newly discovered topics (dynamically) as well as on-the-fly text classification.

7.5 Discussion and Related Work

A number of emerging products and prototypes exist in the Search and Browse space. This section will now briefly consider how the best of these relate to the work described in this chapter.

ISX Corporation's Concept Object Web (Starz et al. 2005) gathers information from documents using automated and human extraction techniques and then present this to users as answers to queries rather than leaving the user to read the set of documents in the query response in order to gain the answer. The response to an initial free-text query consists of summaries of documents together with a list of instances that appear in them split into customisable categories. These instances can be selected and added to the initial query in order to refine the results set. The user can also find out more about the instances. In this case they are shown a knowledge object, which is an aggregation of all the semantic information about the entity including links to the source documents and how each fact was obtained. Users can navigate through the knowledge base by selecting the relation values of the current object. The approach is similar in many respects to that adopted by Squirrel. It provides better support for refinements based upon entities but does not allow the user to refine their search based on the topic of documents or browse around a topic ontology in order to find related documents. Additional facilities offered by Squirrel include NLG and tailoring results according to a user profile.

The /facet (Hildebrand et al. 2006) system adopts a navigational model or facet browsing approach where a user can navigate to heterogeneous resources by making selections on the properties of other semantically related types. This benefits users who do not have a clear idea of what they are searching for or know how it is described. Only those attribute values that, if selected, will lead to at least one result being shown are made available. This ensures the user does not take any "blind alleys". They can also back up by removing previously chosen attributes from their search. One issue with facet browsing is that where there are many possible selections the interface becomes unwieldy. /facet overcomes this by by suggesting possible terms to the user as he types in a text box. Again, only keywords that produce actual results are suggested. An impressive feature of /facet is its ability to build facet selection directly from the metadata with no configuration and at query time. Although /facet does not contain many of the features of Squirrel such as named entity recognition, result consolidation or NLG, the navigation approach adopted and its ability to cope with a large number of facets with no configuration indicate how Squirrel could be extended in this fashion.

The Excalibur product from Convera[1] provides a search interface that allows the user to enter queries and then refine or disambiguate them based on topics that the system knows about; for example, a search for "mobile industry" receives the response:

Did you mean: cell phone as a "telecom", Industries
mobile as a "ceiling decor", Industries

Selecting one of these then tailors the results appropriately indicating that documents have been attributed to topics at index time. In addition, once a user has selected a topic they are then offered related topics:

[1] http://www.convera.com/

Narrow search:	consumer electronics, telecom
Broaden search:	LG cell phone, Motorola cell phone, Nokia cell phone,Samsung cell phone,Sony cell phone, camera phone,clamshell phone, multimedia phone
Related:	cell accessory, sectoral development, by-product

This indicates that there is some sort of topic hierarchy behind the system, but it is not clear that this extends to a full ontology akin to PROTON. Although the system highlights named entities (such as people) in the text, the user is not able to select these and find out more about them, browse to related ontological entities or even refine a search based on the values of properties that particular entities have. Furthermore, the lack of an ontology behind the entities makes it harder to apply rules of the nature described in Sect. 7.3.

7.6 Conclusion and Future Work

We have described Squirrel, a tool for searching and browsing semantically annotated textual resources. The tool provides a hybrid approach to search allowing the user to enter simple search terms and then refine their search or browse to related material through the presentation of appropriate metadata. The tool integrates a number of state-of-the-art components to provide ontology management, named entity recognition on ontology generation and classification. It also includes a set of novel features such as result consolidation, natural language generation, ontology based user profiling, device independence and a natural language query interface.

The tool has been trialled in the BT Digital Library case study. An account of the evaluation is given in Chap. 17.

A number of potential areas for further development include the ability to identify the relationships between entities that are discovered in the query response. For example, a query such as “Java Microsoft” would match a number of different entities including both a topic and a company. The appropriate entities could be chosen based upon a combination of the popularity in the knowledge base (how many documents contain them as annotations), the user’s profile and finally an order of precedence is specified for the domain (e.g., in the digital library, if a topic is identified then this might be chosen over an entity with the same name). In this case the topic Java and the company entity Microsoft might be chosen and as such it might be appropriate to initially show documents from the topic where annotations of the company have been identified. Similarly a query for “Bill Gates Microsoft” would identify the person and company from the knowledge base. Here, as well as showing documents where annotations of both occur, it might be appropriate to show how the two entities are related, which in this case is via the JobPosition instance relating the person to the company.

The display of entity results could also be improved with the use of user profiles. Where a user searches using a term that closely matches two or more entities there is a need for disambiguation (predicting which of the entities the user is searching for by matching the results against the profile). For example, if a user is searching for documents written by an author called Davies (of which there are many different separate instances) the correct author can be chosen by matching the topics of the documents the individual authors have written against topics in the user's profile.

A second area for development is concerned with adopting a reduced configuration, faceted browsing approach such as that offered by /facet and described in the previous section.

Finally, there is a need for better and optimised integration between ontology management and full-text search. Currently the two are separate components and must be queried separately, either sequentially (as in Squirrel) or in parallel followed by an intersection step. This has inherent performance issues which could be addressed by a combined approach able to take advantage of optimisation techniques. In a similar vein the use of both OWLIM and KAON2 in Squirrel would be improved via the use of a single repository that is able to offer the best features of both these components. The development of the ORAKEL system will aim at making it more robust as well as well as at providing enhanced answer generation capabilities beyond returning a mere list of tuples. In some cases, answers could be summarised or at least grouped after some criteria.

Acknowledgment The work described in this chapter was developed as part of the SEKT project, under EU funding (IST IP 2003506826). Further project details may be found at http://www.sekt-project.com/.

References

Choi FYY (2000) Advances in domain independent linear text segmentation'. In Proceedings of NAACL, Seattle, USA, April: 26–33.

Cimiano P, Haase P, Heizmann J (2007) Porting natural language interfaces between domains – A case study with the ORAKEL system. In Proceedings of the International Conference on Intelligent User Interfaces (IUI): 180–189.

Fortuna B, Grobelnik M, Mladenić D (2005) Semi-automatic Construction of Topic Ontology', Semantics, Web and Mining. Joint International Workshop, EWMF 2005 and KDO 2005, Porto, Portugal, October 3–7.

Glover T, Davies J (2005) Integrating device independence and user profiles on the web. BT Technology Journal Vol. 23.

Hildebrand M, van Ossenbruggen J, Hardman L (2006) /facet: A browser for heterogeneous semantic web repositories'. In Proceedings of the 5th International Semantic Web Conference, Athens, USA, November, 2006.

Kiryakov A, Ognyanov D, Manov D (2005) OWLIM – A pragmatic semantic repository for OWL. In Proceedings of International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2005), WISE 2005, 20 November, New York City, USA.

Popov B, Kiryakov A, Ognyanoff D, Manov D, Kirilov A (2004) KIM – A semantic platform for information extaction and retrieval. Journal of Natural Language Engineering, Vol. 10, Issue 3–4, 2004: 375–392.

Starz J, Kettler B, Haglich P, Losco J, Edwards G, Hoffman M (2005) The concept object web for knowledge management. Proceedings of the 4th International Semantic Web Conference, Galway, Ireland, November, 2005.

Chapter 8
Semantic Web for Knowledge Sharing

Nicholas J. Kings and John Davies

Abstract In general, knowledge sharing tools combine the functions of searching for and distributing information. Underpinning knowledge sharing tools is the premise that someone in the user's wider community has already created or accessed relevant information (explicit knowledge transfer) or someone is able to provide help or advice (tacit knowledge transfer). Whatever the modes of communication used, the primary goal of the software is to create group memory and team awareness. This chapter describes a knowledge sharing tool, Squidz, to facilitate users in a community of practice or community of interest to share Web pages, as well as gain an awareness of others' interests and expertise. The approach taken by Squidz is to associate a formal topic ontology together with an informal folksonomy, through the ability to annotate Web pages. Furthermore, knowledge sharing occurs as a result of normal user activity (browsing).

8.1 Introduction

In general, knowledge sharing tools combine the functions of searching for and distributing information. Underpinning knowledge sharing tools is the premise that someone in the user's wider community has already created or accessed relevant information (explicit knowledge transfer) or someone is able to provide help or advice (tacit knowledge transfer). Knowledge sharing software supports the activities of collating, categorising and distributing information (Shneiderman 2003). Whatever the modes of communication used, the primary goal of the software is to create group memory and team awareness (Udel 2004, Rafaeli and Raban 2005).

This chapter describes a knowledge sharing tool, Squidz, to facilitate users in a community of practice or community of interest to share Web pages, as well as gain an awareness of other's interests and expertise.

N.J. Kings (✉)
Next Generation Web group, BT Research, Ipswich, IP5 3RE, UK
e-mail: nick.kings@bt.com

J. Davies et al. (eds.), *Semantic Knowledge Management*,

8.1.1 Models of Information Sharing

The Semantic Web (Berners-Lee et al. 2001) can provide enhanced information access based on the exploitation of machine-processable meta data. Central to the vision of the Semantic Web are ontologies, which are seen as facilitating knowledge sharing and re-use between agents, be they human or artificial (Fensel 2001). As such, the use of ontologies and supporting tools offer an opportunity to significantly improve knowledge management capabilities on the intranets of organisations and on the wider web. Furthermore, Mika (2005) suggests that although the Semantic Web has been defined to facilitate machine understanding of the World Wide Web, the process of creating and maintaining that shared ontology is a purely social activity. Mika further proposes that the understanding of social presence is crucial in understanding how an ontology evolves and gains acceptance.

Recently, we have seen the emergence of a number of very popular community-based systems on the Web, often referred as social computing systems. Typically in systems such as flickr[1] and del.icio.us,[2] instead of using a centralised form of classification, users are encouraged to assign freely chosen keywords, called tags, to pieces of information or data, a process known as tagging. As a community of users generate a series of tags for overlapping and common items, a "folksonomy" can been seen to emerge (Golder and Huberman 2006, Marlow et al. 2006). Since folksonomies are incremental and end user-generated and therefore inexpensive to implement, advocates of folksonomy believe that it provides a useful low-cost alternative to more traditional controlled vocabularies or classification schemes.

The combination of social networking systems such as FOAF[3] and XFN[4] with the development of tag-based folksonomies has been seen as providing significant amounts of end user generated metadata; as such, blogs and wikis were initially were seen as a basis for the semantic environment. This has been seen as part of a change from accessing static Web pages to the use of the web as an application platform (O'Reilly 2005):

- The change from centralised information sources to an approach of creating and distributing Web content itself, characterised by open communication, the willingness to share and re-use information, and "the market as a conversation" (Levine et al. 2000).
- The change from using web sites as point sources of information, stored within static Web pages, to sources of remotely accessible information, through the use of network accessible APIs or services.

In contrast to defined, formal taxonomies, categories in a folksonomy may appear to be arbitrary and idiosyncratic. However, a particular tag is chosen for

[1] http://www.flickr.com/

[2] http://del.icio.us/

[3] http://xmlns.com/foaf/0.1/

[4] http://gmpg.org/xfn/

a particular Web page based upon an individual's own understanding of the content being tagged, which combines the personal, social, and technical understanding of that content (Mika 2005). By publishing a tag and tagged content to a wider audience, other users are subtly encouraged to explore other tagged content and other users' interests.

Pind (2005) suggests that there is anecdotal evidence to show that the sets of tags used within a community tend to converge upon a common set of agreed meanings and usage. Pind further suggests, however, that tagging software could be improved by the addition of the following five features: the software should suggest appropriate tags; the software should display related tags and topics; suggest tags that others have used to describe the same items; infer topic hierarchies from the way tags are used; and, allow a user to quickly edit and change tags that have already been applied. The Squidz tool addresses all of these issues.

8.2 Squidz Design

The main function of Squidz is to share knowledge in the form of textual annotations about Web pages, within a community of business users. The secondary purpose of Squidz is to allow a user to discover new social contacts or sources of information, through discovery of shared interests. The important features implemented by the software are:

- To improve software adoption, Squidz must be useful to an individual user without relying on any other's contributions. Squidz should be able to be used as an advanced book marking tool.
- All users can view and comment upon any other's annotations.
- An annotation is made in a technical and a social context. The technical context is represented by the topics associated with the annotation; the social context is represented by the community where the annotation is posted.
- The software filters and presents appropriate annotations, based upon each user's technical and social context. In effect, a relevant annotation is one made by a closely related person within a topic area related to the current page.
- Squidz models a user's social network in a simple manner (Shneiderman 2003): sharing pages to oneself; sharing to close work or team colleagues; sharing to members of a community of practise; or, sharing to all users of the system. By sharing to a particular set of people, the user is stating that, in their opinion, a page is more relevant to those people than others; the model of sharing, as such, does not have a concept of private bookmarks.

Squidz is implemented as a browser plug-in, making use of a number of web services to classify browsed Web pages and to retrieve annotations made by other members of the user community; Squidz is just one potential interface to the meta-data generated within the community. Squidz is being developed for communities of interest, or practice, within a corporate environment. In common

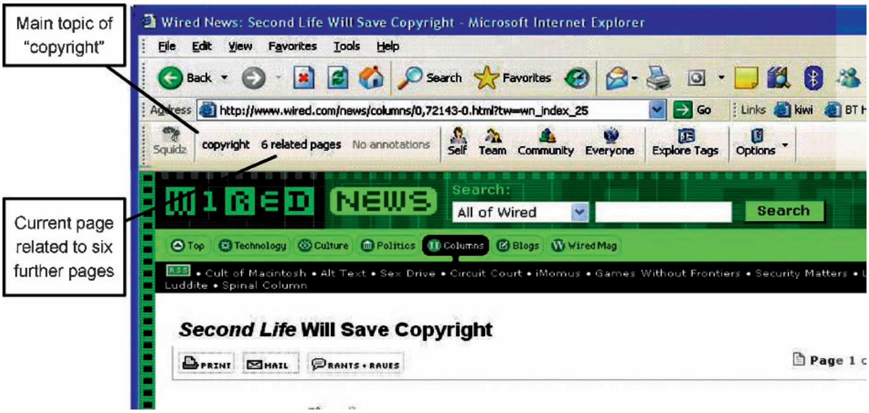

Main topic of "copyright"
Current page related to six further pages
Wired News: Second Life Will Save Copyright - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://www.wired.com/news/columns/0,72143-0.html?tw=wn_index_25
Go Links kiwi BT H
Squidz copyright 6 related pages No annotations
Self Team Community Everyone Explore Tags Options
WIRED NEWS
Search: All of Wired Search
Top Technology Culture Politics Columns Blogs Wired Mag
RSS • Cult of Macintosh • Alt Text • Sex Drive • Circuit Court • iMomus • Games Without Frontiers • Security Matters • Luddite • Spinal Column
Second Life Will Save Copyright
PRINT MAIL RANTS + RAVES
Page 1 c

Fig. 8.1 Squidz tool bar within Internet Explorer

with other tools being developed within the SEKT project, Squidz is utilising the PROTON ontology (Bontcheva et al. 2006, Kiryakov 2006). PROTON is an ontology for knowledge management, modelling entities such as Documents with associated Topics as well as Users with Interests and Communities of Users.

As a user browses the internet or their intranet, the Web page is classified against the formal ontology, and that classification is used to retrieve related pages. By clicking on the icons and words in the plug-in, separate information windows are created, rather than attempting to merge information within the browsed page.

Figure 8.1 shows the main user interface to Squidz, after a user has been browsing: the Web page's main topic is "copyright" and that six related annotated pages have been previously shared by other users; the main topic for a Web page is either the highest ranked topic returned by the classifier or, if this user has already annotated this page, the main topic is taken from that annotation.

As each page is visited, the URL for the Web page is sent to the classifier web service; the URL is also sent to the knowledge server to retrieve related topics and annotations. The classifier web service returns the four highest ranking subject topics, for each Web page, as well as up to ten of the most significant words and phrases identified in the text. For example, a Web page containing about Web2.0 may be classified as "Users", "Design" and "Internet", whereas a news page may be classified against topics such as "Elections", "Planning" and "Politics".

8.2.1 Semantic Annotations

When sharing a Web page in Squidz, the user has the opportunity to provide some folksonomy-style tags which describe the content of the page, along with a comment about the page. Squidz models an annotation as the following: the user's comment;

a set of formal topics, generated by the page classifier; a set of informal user-supplied tags; the target audience (individual, team, community or world as chosen by the user); and, a set of keywords and phrases, also generated by the page classifier. For each user tag, Squidz derives an associated set of keywords; each tag is semantically characterised by that keyword vector rather than the character string the user chose to represent the concept denoted by that particular tag.

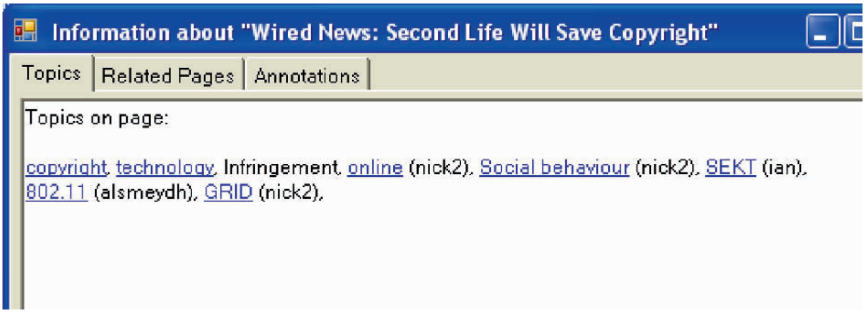
As Web pages are annotated with tags, the set of words for each tag is re-calculated automatically based upon the keywords stored within the page annotation. The keywords are associated with each tag in order to find related pages even if pages have been tagged with different terms; the derived keywords allow pages to be tagged with similar words, even though each user has their own understanding of that word. For example, one user's tag of "project" may represent the same concept as another user's tag of "SEKT". Adding an annotation forces the system to recalculate the keywords associated with the user's tags and subsequently all pages that are now related to the current page. The set of keywords, associated with a tag can change over time, as the use of the tag changes.

Annotations are also made to a particular context, such as "Self", "Team", "Community" or "Everyone". This is used as a recommendation as to whom the annotation may be most relevant; by taking into account the suitability of an annotation, this may reduce the "cost" of a user understanding the importance or relevance of that particular Web page (Shneiderman 2003).

8.3 Using Squidz

Squidz provides the user with "peripheral vision" of previously shared Web pages and annotations related to the current page. As a Web page is viewed by a user, a request for related pages and their associated annotations is made to the remote web service.

Figure 8.2 shows that eight topics have been identified that are related to the current page: three topics from the defined topic ontology, and, five user created tags:


Information about "Wired News: Second Life Will Save Copyright"
Topics
Related Pages
Annotations

Topics on page:

copyright, technology, Infringement, online (nick2), Social behaviour (nick2), SEKT (ian), 802.11 (alsmeydh), GRID (nick2),

Fig. 8.2 Topics related to current page

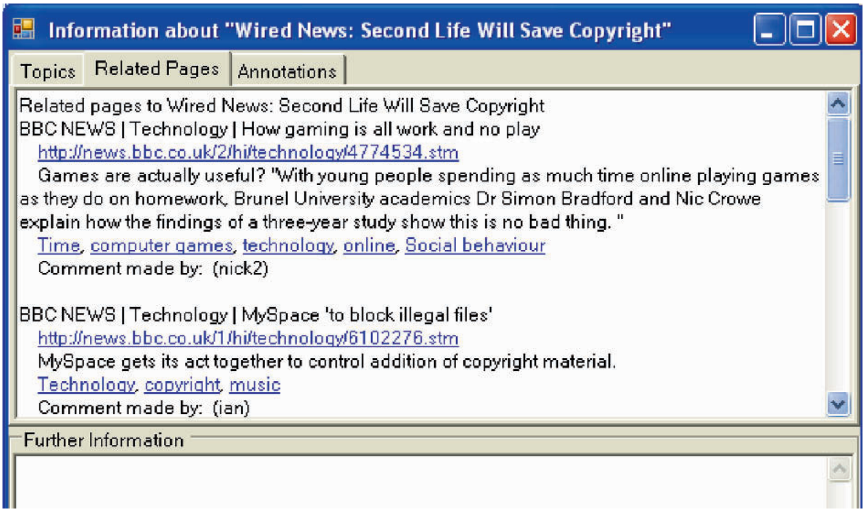
user- generated tags are identified by placing the name of the user who generated the tag in parentheses. As described above, the defined topics have been identified by the classifier. The list of related user tags is formed from the tags found within the returned list of related pages. Each underlined topic or tag can be clicked to cause all of the annotations made with that particular topic to be retrieved and displayed; if a topic is not underlined, then not annotations have been made containing that topic.

Figure 8.3 shows the scrollable list of annotated pages found to be related to the currently viewed Web page. As in Fig. 8.2, clicking an underlined topic will retrieve annotations; clicking an underlined URL will cause the browser to load that particular page. Implementing the user interface in this manner allows a user to explore across Web pages, rather than having to visit a particular website and then start exploring tags and relations from that point onwards.

8.3.1 Sharing Annotations

By clicking on “Self”, “Team”, “Community” or “Everyone” (as shown Fig. 8.3), a user can choose to share an annotation about the currently viewed Web page. This is implemented via a less intrusive pop-up window, rather than by changing the contents of the main browser widow. Figure 8.4 shows that comments about a particular Web page will be shared to members of this user’s community.

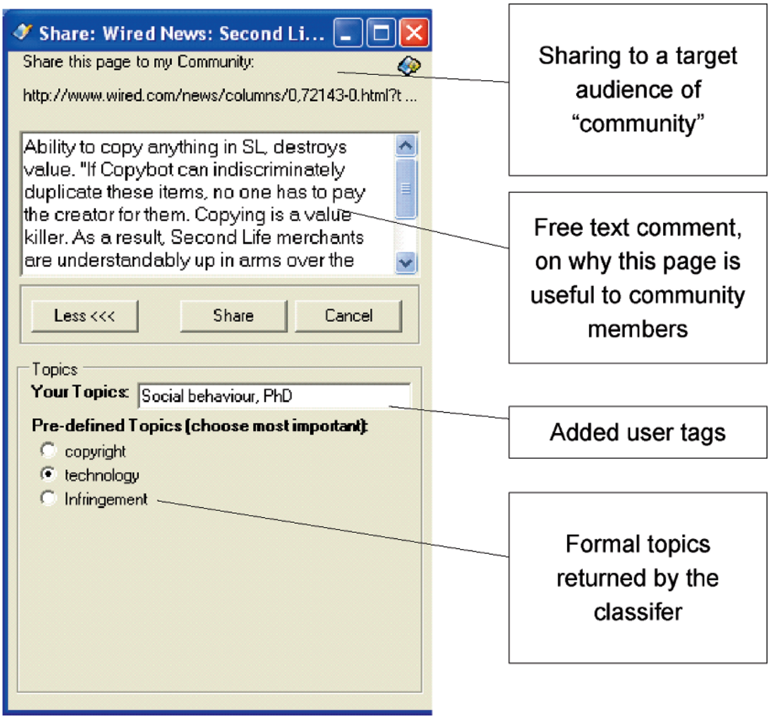
Sharing to a page to a particular group of people does not preclude others from viewing that annotation, as having the ability to view every person’s contribution is a crucial way of building an active set of users. However, the user-chosen target


Information about "Wired News: Second Life Will Save Copyright"
Topics
Related Pages
Annotations
Further Information

Related pages to Wired News: Second Life Will Save Copyright
BBC NEWS | Technology | How gaming is all work and no play
http://news.bbc.co.uk/2/hi/technology/4774534.stm
Games are actually useful? "With young people spending as much time online playing games as they do on homework, Brunel University academics Dr Simon Bradford and Nic Crowe explain how the findings of a three-year study show this is no bad thing. "
Time, computer games, technology, online, Social behaviour
Comment made by: (nick2)

BBC NEWS | Technology | MySpace 'to block illegal files'
http://news.bbc.co.uk/1/hi/technology/6102276.stm
MySpace gets its act together to control addition of copyright material.
Technology, copyright, music
Comment made by: (ian)

Fig. 8.3 Pages related to current page, ranked by social network



Share: Wired News: Second Li...

Share this page to my Community:

http://www.wired.com/news/columns/0,72143-0.html?t ...

Ability to copy anything in SL destroys value. "If Copybot can indiscriminately duplicate these items, no one has to pay the creator for them. Copying is a value killer. As a result, Second Life merchants are understandably up in arms over the

Your Topics: Social behaviour, PhD

Pre-defined Topics (choose most important):

○ copyright
● technology
○ Infringement

Sharing to a target audience of "community"

Free text comment, on why this page is useful to community members

Added user tags

Formal topics returned by the classifer

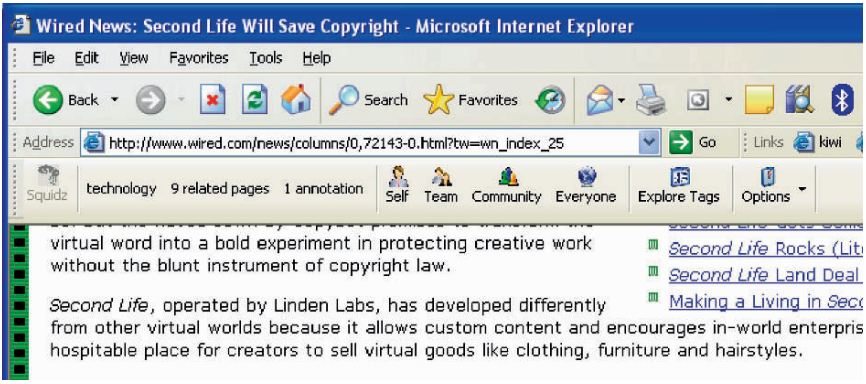
Fig. 8.4 Sharing an annotation

group of the annotation is used in the ranking algorithm when calculating related pages: a page shared to "Team" by one of my team members is ranked higher than a page shared by the same person but shared to "Community".

Figure 8.5 shows that after the annotation has been made, the Web page is now related to nine pages, and the main formal topic for the page is shown as "technology". By making an annotation the topics related to this page have changed, as shown by comparing Figs. 8.2 and 8.6. By making this annotation, "Infringement" is now able to be clicked: there is at least one annotation for this topic. The annotation explicitly mentions the user's tag of "PhD", which would be expected to be shown for this page. However, by annotating this page, the algorithm has re-calculated the tags' keyword vectors, such that there is now a similarity between this page and the tags of "Tagging" and "Classification techniques".

8.4 Evaluating Squidz

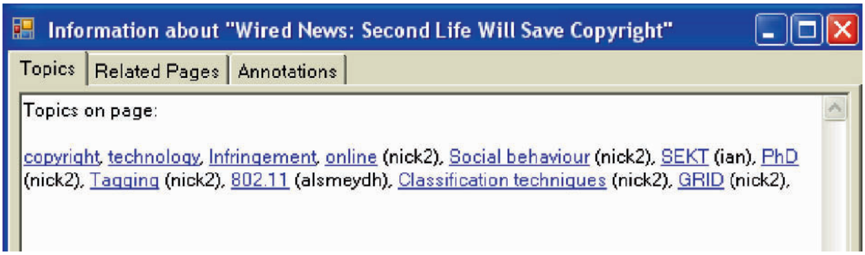
For the trial, twenty five users registered and downloaded the software: ten of those were found to be regular users and a further eight users have made at least one annotation. A number of Squidz users were interviewed for 30 min with a series of


Wired News: Second Life Will Save Copyright - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://www.wired.com/news/columns/0,72143-0.html?tw=wn_index_25 Go Links kiwi
Squidz technology 9 related pages 1 annotation Self Team Community Everyone Explore Tags Options
Second Life Rocks (Lit
Second Life Land Deal
Making a Living in Seco

virtual word into a bold experiment in protecting creative work without the blunt instrument of copyright law.

Second Life, operated by Linden Labs, has developed differently from other virtual worlds because it allows custom content and encourages in-world enterpris hospitable place for creators to sell virtual goods like clothing, furniture and hairstyles.

Fig. 8.5 Updated display, after a page has been annotated


Information about "Wired News: Second Life Will Save Copyright"
Topics
Related Pages
Annotations

Topics on page:

copyright, technology, Infringement, online (nick2), Social behaviour (nick2), SEKT (ian), PhD (nick2), Tagging (nick2), 802.11 (alsmeydh), Classification techniques (nick2), GRID (nick2),

Fig. 8.6 Recalculated related topics, after the annotation has been added

semi-structured questions. The interview was structured to explore a number of themes around knowledge sharing, in general, and the use of Squidz, in particular.

Longer term user acceptance will be an important aspect to measure, since Squidz constantly monitors web page access and that could be considered as quite intrusive. However, the continual classification of a web page against the formal ontology was seen to be of great use. Squidz was seen to be of particular benefit when a user was searching for new information, through the ability to display related and previously annotated web pages.

8.5 Conclusions

The approach taken by Squidz is to associate a formal topic ontology together with an informal folksonomy, through the ability to annotate Web pages. Presenting information on related pages and topics, through the toolbar, allows a user to browse and explore when convenient to the user, rather than forcing a particular mode of usage; knowledge sharing occurs as a result of normal user activity (browsing). Squidz is intended to test the hypothesis that information sharing is more effective when the software is aware of both the social and technical context of that information. Though further improvements to the tool are planned, Squidz has already gained positive user feedback and acceptance.

Acknowledgement This work was supported by the IST Programme of the European Community under SEKT (IST-IP-2003-506826).

References

Berners-Lee T, Hendler J and Lassila O (2001) The semantic web. Scientific American.

Bontcheva K, Davies J, Duke A, Glover T, Kings N and Thurlow I (2006) Semantic Information Access. In Davies, J., Studer, R. & Warren, P. (Eds.) Semantic Web Technologies: Trends and Research in Ontology-based Systems. Wiley, Chichester.

Fensel D (2001) Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce, Springer, Berlin Heidelberg.

Golder S and Huberman B A (2006) Usage patterns of collaborative tagging systems. Journal of Information Science, Vol. 32, No. 2, 198–208. Web page: http://www.hpl.hp.com/research/idl/papers/tags/index.html, last viewed on 21st July, 2006.

Kiryakov A (2006) Ontologies for Knowledge Management. IN Davies, J., Studer, R & Warren, P. (Eds.) Semantic Web Technologies: Trends and Research in Ontology-based Systems. Wiley, Chichester.

Levine R, Locke C, Searls D and Weinberger D (2000) The Cluetrain Manifesto: The End of Business as Usual, Pearson Education, London.

Marlow C, Naaman M, Boyd, D and Davis M (2006) HT06, Tagging Paper, Taxonomy, Flickr, Academic Article, ToRead. Hypertext 2006. ACM Press, New York. Web page: http://www.danah.org/papers/Hypertext2006.pdf, last viewed on 8th March, 2007.

Mika P (2005) Ontologies are us: A unified model of social networks and semantics. IN Gil, Y., Motta, E., Benjamins, V. R. & Musen, M. A. (Eds.) The Semantic Web - ISWC 2005, Galway, Ireland. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence) vol. 3729, Springer, Berlin Heidelberg.

O'Reilly T (2005) What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. O'Reilly Media, Inc. Web page: http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html, last viewed on 27th February, 2007.

Pind L (2005) Folksonomies: How we can improve the tags. Pinds.com. Web page: http://pinds.com/articles/2005/01/23/folksonomies-how-we-can-improve-the-tags, last viewed on 27th February, 2007.

Rafaeli, S and Raban, D R (2005) Information sharing online: A research challenge. International Journal of Knowledge and Learning, Vol. 1, No. 2, 62–79. Web page: http://www.inderscience.com/search/index.php?mainAction = search&action = record&rec_id = 6251, last viewed on 27th February, 2007.

Shneiderman B (2003) Leonardo's Laptop: Human Needs and the New Computing Technologies, MIT, London.

Udel J (2004) The new social enterprise. InfoWorld, Vol. 26, No. 13, 47, 50–52.

Chapter 9
Natural Language Generation from Ontologies

Kalina Bontcheva and Brian Davis

Abstract In order to realise the semantic web vision, the creation of semantic annotation, the linking of web pages to ontologies, and the creation, evolution and interrelation of ontologies must become automatic or semi-automatic processes. Natural Language Generation (NLG) takes structured data in a knowledge base as input and produces natural language text, tailored to the presentational context and the target reader. NLG techniques use and build models of the context and the user and use them to select appropriate presentation strategies.

In the context of Semantic Web or knowledge management, NLG can be applied to provide automated documentation of ontologies and knowledge bases. Unlike human-written texts, an automatic approach will constantly keep the text up-to-date which is vitally important in the Semantic Web context where knowledge is dynamic and is updated frequently. This chapter presents several Natural Language Generation (NLG) techniques that produce textual summaries from Semantic Web ontologies. The main contribution is in showing how existing NLG tools can be adapted to take Semantic Web ontologies as their input, in a way which minimises the customisation effort.

A major factor in the quality of the generated text is the content of the ontology itself. For instance, the use of string datatype properties with implicit semantics leads to the generation of text with missing semantic information. Three approaches to overcome this problem are presented and users can choose the one that suits their application best.

9.1 Introduction

The Semantic Web aims to add a machine tractable, application-independent layer to complement the existing web of natural language hypertext. In order to realise this vision, the creation of semantic annotation, the linking of web pages to

K. Bontcheva (✉)
Department of Computer Science, University of Sheffield, Sheffield, UK
e-mail: K.Bontcheva@dcs.shef.ac.uk

J. Davies et al. (eds.), *Semantic Knowledge Management*,

ontologies, and the creation, evolution and interrelation of ontologies must become automatic or semi-automatic processes. Natural Language Generation[1] (NLG) takes structured data in a knowledge base as input and produces natural language text, tailored to the presentational context and the target reader (Reiter and Dale 2000). NLG techniques use and build models of the context and the user and use them to select appropriate presentation strategies. For example, deliver short summaries to the user's WAP phone or a longer multimodal text if the user is using their desktop.

In the context of Semantic Web or knowledge management, NLG can be applied to provide automated documentation of ontologies and knowledge bases. Unlike human-written texts, an automatic approach will constantly keep the text up-to-date which is vitally important in the Semantic Web context where knowledge is dynamic and is updated frequently. The NLG approach also allows generation in multiple languages without the need for human or automatic translation (Aguado et al. 1998). This is an important problem firstly because textual documentation is more readable than the corresponding formal notations and thus helps users who are not knowledge engineers to understand and use ontologies. Secondly, a number of applications have now started using ontologies to encode and reason with internally, but this formal knowledge needs to be also expressed in natural language in order to produce reports, letters, etc. In other words, NLG can be used to present structured information in a user-friendly way.

There are several advantages to using NLG rather than using fixed templates where the query results are filled in:

- NLG can use different sentence structures depending on the number of query results, for example, conjunction versus itemised list.
- depending on the user's profile of their interests, NLG can include different types of information – affiliations, email addresses, publication lists, indications on collaborations (derived from project information).
- given this variety of what information from the ontology can be included and how it can be presented, depending on its type and amount, writing templates will be unfeasible because there will be too many combinations to be covered.

This variation comes from the fact that it is expected that each user of the system will have a profile comprising of user supplied (or system derived) personal information (name, contact details, experience, projects worked on), plus information derived semi-automatically from the user's interaction with other applications. Therefore, there will be a need to tailor the generated presentations according to user's profile.

NLG systems that are specifically targeted towards Semantic Web ontologies have started to emerge only recently. Initial ones were based on templates, verbalising closely the ontology structure. More recent ones generate more fluent reports, oriented towards end-users, not ontology builders.

[1] For an in-depth introduction to NLG see Reiter and Dale (2000).

In contrast to these applied NLG approaches, at the other end of the spectrum are sophisticated ones, which offer tailored output based on user models. The trade-off is between applied approaches, exploring generalities in the domain ontology and with lower customisation overheads, and, on the other hand, sophisticated, more flexible and expressive systems, which, however tend to be difficult to adapt by non-NLG experts. Our experience shows that knowledge management and Semantic Web Ontologies tend to evolve over time, so it is essential to have an easy-to-maintain NLG approach.

9.2 Generation from Semantic Web Ontologies

9.2.1 Generation from Taxonomies

PEBA is an intelligent online encyclopedia which generates descriptions and comparisons of animals (Dale et al. 1998). The system uses text patterns for text planning which are appropriate for the fairly invariant structure of the generated descriptions. PEBA has a taxonomic knowledge base which is directly reflected in the generated hypertext because the links always point at other elements in the hierarchy. Based on the discourse history, that is, what was seen already, the system modifies the page-opening to take this into account. For example, if the user has followed a link to *marsupial* from a node about the *kangaroo*, then the new text will be adapted to be more coherent in the context of the previous page:

"Apart from the Kangaroo, the class of Marsupials also contains the following subtypes"… (Dale et al. 1998).

The main focus in PEBA is on the generation of comparisons which improve the user's understanding of the domain by comparing the currently explained animal to animals already familiar to the user (from common knowledge or previous interaction).

The system also does a limited amount of tailoring of the comparisons, based on a set of hard-coded user models based on stereotypes – novice or expert – used for variations in language and content. For example, when choosing a target for a comparison, the system might pick cats for novice users, as they are commonly known animals.

9.2.1.1 Ontogeneration

The ONTOGENERATION project (Aguado et al. 1998) explored the use of a linguistically oriented ontology (the Generalised Upper Model (GUM)) (Bateman et al. 1995) as an abstraction between generators and their domain knowledge base (chemistry in this case). The Generalised Upper Model (GUM) is a linguistic ontology with hundreds of concepts and relations, for example, part-whole, spatiotemporal, cause-effect. The types of text that were generated are: concept definitions, classifications, examples, and comparisons of chemical elements.

However, the size and complexity of GUM make customisation more difficult for non-experts. On the other hand, the benefit from using GUM is that it encodes all linguistically-motivated structures away from the domain ontology and can act as a mapping structure in multilingual generation systems. In general, there is a trade-off between expressivity and the number of linguistic constructs in the ontology.

9.2.2 Shallow/Template-Based Generation

Wilcock (2005) provides an overview of shallow/template or XML-based NLG. His work provides descriptions of XML based pipelined architectures for NLG, involving: Text Planning, Micro planning and Surface Realisation. The research is based on practical experience which applies NLG to a spoken dialog system. The author claims that XML as a generation tool fits into the pipeline model for NLG. He argues that powerful methods for processing XML already exist. In Wilcock's approach XML transformations are performed on text plan trees in order to produce text specification trees using Extensible Stylesheet Language Transformations – XSLT.

Wilcock (2005) emphasizes that the role played by XML transformations across text plans as a consequence of using XML templates implies template based text planning and not just template based generation, whereby the text plan is passed through various stages of the NLG pipeline for processing using XSLT.

An example implementation is provided in Wilcock (2005), whereby a bilingual NLG component, which generates responses in both Finnish and English concerning, queries against a Helsinki Bus timetable enquiry system. The linguistic output varies from short elliptical phrases to full sentences.

An advantage is that XML is that the internal representation can be defined as a DTD – Document Type Definition or XML schema and furthermore XML can be checked by standard XML validation techniques. However, XSLT is not suitable for processing all languages, especially where substantial morphological processing is required as is the case in Finnish. One can however use XSLT with extension functions for JAVA. Wilcock (2005) argues that in comparison to general programming languages, XSLT is more accessible and usable by non-skilled users, that is, linguists. However, as XSLT can be classed as a scripting languages (such as Perl, Python) we argue that the cost up of upskilling a non-specialist user would be perhaps no different than learning a general purpose programming language. However XSLT does have the advantages of its declarative nature which is very in line with the underlying grammar/linguistics formalisms and educational/development languages (i.e., declarative languages such as PROLOG and LISP).

One avenue proposed in Wilcock (2005) is a hybrid combination of shallow XML-based methods in combination with traditional deeper NLG methods such as OpenCCG (White 2004, 2006, White and Baldridge 2003) (which in itself combines both unification based and statistical approaches). Such combined approach is likely to fulfil the needs for scalable, efficient and high quality method of generating texts from ontologies.

9.2.3 Inference-Based Content Selection

The work on NLDI (Mellish and Sun 2005) focuses on the problem of selecting the relevant material for inclusion into the final natural language output of the system, referred to as *content determination* in NLG. The authors Mellish and Sun (2005) argue that little success has been achieved at deriving general models for the structure of content determination since the specifics of content determination vary according to the target domain. Furthermore the authors highlight that literature within NLG reference architecture “has little to say” about this topic. Top down and bottom up classes of content determination are distinguished, whereby top-down approaches have specific goals in mind such as “convincing or persuading” the reader about some piece of content. On the other hand bottom-up approaches produce more descriptive texts and their goals are more widely dispersed (Mellish and Sun 2005). Text coherence plays an important role in these types of problems. In general, content determination is a difficult problem since two different “worlds” are involved – the domain model itself and the linguistic model, the system is attempting to map to (Mellish and Sun 2005).

The work presented in Mellish and Sun (2005) is different from other approaches (such as Bontcheva and Wilks 2004) in that it verbalises the ontology class axioms instead of generating text concerning individuals in the ontology. A given axiom may have multiple rules and hence multiple possible realisations. Where a logical formula has multiple realisations, a measure of linguistic complexity as in the number of words in the English string generated can be used the make the selection. Better measures according to Mellish and Sun (2005) could also take into account the shape of the parse tree. In general, linguistic complexity does not necessarily mirror the complexity of the underlying formula. Though complex formulae can yield complex linguistic output, complexity also depends on the mapping between logical formula and surface realisation and the underlying linguistic resources (Mellish and Sun 2005).

With respect to the selection of material for content determination, the same procedure adopted in ILEX (O’Donnell et al. 2001) is used. Logical axioms can be seen as forming a graph where each axiom corresponds to different possible transitions in a coherent text, that is, a text proceeds from one sentence to another by exploiting shared entities or by virtue of rhetorical relations between sentences. Detecting the axioms to be expressed in the answer – this involves a best-first search for axioms, starting at an entity X where each axiom is evaluated based on the following measurements: (1) how close it is (in terms of edges of the graph) to the concept X, (2) how intrinsically important and understandable it is, and (3) how may times it has already been presented to the user.

Following the ILEX model, these three measures are multiplied together for a text of length *n* – the *n* facts with the highest measure are selected for inclusion as content. Three components perform the measure. The first component measure ensures that the retrieved axioms are relevant to the question or query to be answered – best suited axioms are ones containing the class X. On the other hand axioms that are only indirectly involved with X can be selected if they score well according to component 2 (provided there are not enough closer axioms). The second component can sensitize the system to the user preferring an axiom that involves concepts known to the user or axioms that have not previously been told to them. However, the authors note that this has not yet been exploited. Finally component 3 penalizes axioms already presented.

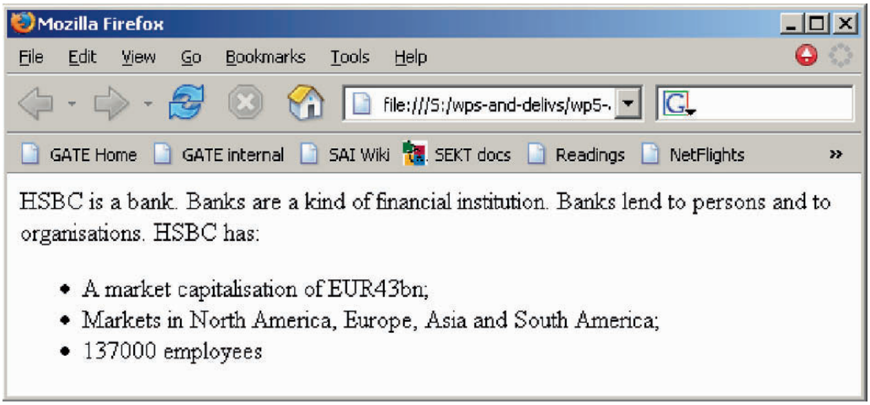
9.2.4 NLG in CLEF – Clinical E-Science Framework

CLEF (Rector et al. 2003) focuses in integrating clinical care with biomedical research and the creation of an Electronic Patient Repository. The project involves the use of IE (Information Extraction) from medical texts and then NLG from the resulting metadata. Techniques from language generation such as Power et al. (1998) are applied to produce reports from integrated information and metadata in the repository and to provide an electronic health care record for the medical specialist. In addition, the NLG contribution to CLEF aims to provide a natural language interface to the Knowledge base in order to aid knowledge engineers to build and maintain the knowledge base and furthermore to permit clinical experts to provide quality assurance across the knowledge base while still being shielded from the underlying complex formalisms. In this respect, the work bears similarity to the controlled language approach from SEKT, where again users can edit ontologies in natural language (see Chap. 4).

9.3 Summary Generation from Ontologies

This section focuses in depth on the summary generation problem, addressed in the ONTOSUM system (Bontcheva 2005), which was developed in the context of the SEKT research project. Summary generation starts off by being given a set of statements (i.e., triples), in the form of RDF/OWL. Since there is some repetition, these triples are first pre-processed to remove already said facts. In addition to triples that have the same property and arguments, the system also removes triples involving inverse properties with the same arguments, as those of an already verbalised one. The information about inverse properties is provided by the ontology (if supported by the representation formalism). An example summary is shown in Fig. 9.1.

The ONTOSUM system is implemented as a set of components in the GATE infrastructure (Bontcheva et al. 2004), which provides an easy-to-use graphical development environment for NLP systems. In particular, we make use of its ontology support, which provides language-independent access to ontologies. The advantage



HSBC is a bank. Banks are a kind of financial institution. Banks lend to persons and to organisations. HSBC has:

- A market capitalisation of EUR43bn;
- Markets in North America, Europe, Asia and South America;
- 137000 employees

Fig. 9.1 Example generated summary

of this approach is that our generator can handle RDF, DAML + OIL, and OWL, without any modifications, as it uses the format-independent GATE API, rather than format-specific ones.

The lexicalisations of concepts and properties in the ontology can be specified by the ontology engineer, be taken to be the same as concept names themselves, or added manually as part of the customisation process. For instance, the AKT ontology[2] provides label statements for some of its concepts and instances, which are found and imported in the lexicon automatically. ONTOSUM is parameterised at run time by specifying which properties are to be used for building the lexicon.

A similar approach was first implemented in a domain- and ontology-specific way in the MIAKT system (Bontcheva and Wilks 2004). In ONTOSUM it is extended towards portability and personalisation, that is, lowering the cost of porting the generator from one ontology to another and generating summaries of a given length and format, dependent on the user target device.

Similar to the PEBA system, summary structuring is done using *discourse/text schemas* (Reiter and Dale 2000), which are script-like structures which represent discourse patterns. They can be applied recursively to generate coherent multisentential text. In more concrete terms, when given a set of statements about a given concept/instance, discourse schemas are used to impose an order on them, such that the resulting summary is coherent. For the purposes of our system, a coherent summary is a summary where similar statements are grouped together.

The schemas are independent of the concrete domain and rely only on a core set of four basic properties – active-action, passive-action, attribute, and part-whole. When a new ontology is connected to ONTOSUM, properties can be defined as a sub-property of one of these four generic ones and then ONTOSUM will be able to verbalise them without any modifications to the discourse schemas. However, if more

[2] http://www.aktors.org/ontology/

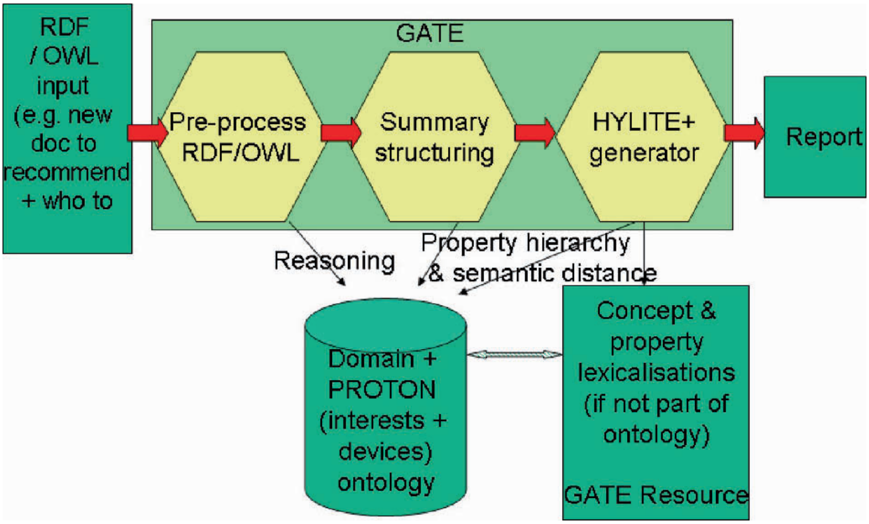

RDF / OWL input (e.g. new doc to recommend + who to
GATE
Pre-process RDF/OWL
Summary structuring
HYLITE+ generator
Report
Reasoning
Property hierarchy & semantic distance
Domain + PROTON (interests + devices) ontology
Concept & property lexicalisations (if not part of ontology)
GATE Resource

Fig. 9.2 Knowledge generation architecture

specialised treatment of some properties is required, it is possible to enhance the schema library with new patterns, that apply only to a specific property.

Next ONTOSUM performs *semantic aggregation*, that is, it joins RDF statements with the same property name and domain as one conceptual graph. Without this aggregation step, there will be three separate sentences instead of 1 bullet list (see Fig. 9.1), resulting in a less coherent text.

Finally, ONTOSUM verbalises the statements using the HYLITE + surface realiser. The output is a textual summary. The overall system architecture is shown in Fig. 9.2 and further details can be found in Bontcheva (2005).

9.4 Customising OntoSum to New Ontologies

Here we provide a step-by-step example first of how to customise ONTOSUM for an ontology, and next – of the generation process itself. The two processes can be repeated iteratively, that is, the user can carry out some customisation, run ONTOSUM, analyse the problems, then go back to editing the lexicon or the ontology, etc.

The first stage in connecting an ontology to ONTOSUM is to execute the lexicon generation component, to obtain automatically the domain lexicon from the ontology. This lexicon can be extended and refined further manually by the user.

Once the lexicon has been completed, the next (optional) step is to introduce in the ontology property hierarchy the four linguistically-motivated properties discussed above. To facilitate this process, ONTOSUM has a heuristic for properties whose name starts with *has*, which are classified automatically as attributive on the basis of their names.

In addition, ONTOSUM enables the user to specify properties which should be filtered out from the textual summaries (see below for more detail).

At this stage, ONTOSUM is ready to be run on a given RDF/OWL description of an instance to produce its natural language summary. It is the responsibility of the application which calls ONTOSUM to choose which instance is to be described, that is, to provide the RDF/OWL input. In the simplest case, this could be the user browsing the ontology, clicking on an instance, and asking for its textual summary.

For the sake of simplicity, throughout this example we will assume that there are no length restrictions for the summary and it will be generated as plain text.

In this example we will use the AKT ontology and part of the RDF description of one of the authors appears below:

```
    <rdf:Description
 rdf:about="http://…#K.Bontcheva.dcs.shef.ac.uk">
       <ns0:family-name>Bontcheva</ns0:family-name>
       <ns0:full-name>Kalina Bontcheva</ns0:full-name>
       <ns0:given-name>Kalina</ns0:given-name>
       <ns0:has-appellation>Dr</ns0:has-appellation>
       <ns0:has-email-address>K.Bontcheva@dcs…</ns0:has-…>
       …
       <rdf:type rdf:resource="http://…#Researcher-In-Academia"/>
    </rdf:Description>
```

As shown in Fig. 9.2, the first phase is *pre-processing*. Let us assume that there were no previous explanations, so no properties of this instance need to be removed to avoid repetition. During pre-processing ONTOSUM also removes the properties specified for filtering out by the user. For example, properties encoding the provenance of this instance or providing lexical information (e.g., full-name) may be excluded in this way. Let us assume, for example, `full-name`, `family-name`, and `given-name` are specified as properties to be filtered out. Consequently, at the end of the pre-processing phase the input is transformed into:

```
    <rdf:Description
 rdf:about="http://…#K.Bontcheva.dcs.shef.ac.uk">
       <ns0:has-appellation>Dr</ns0:has-appellation>
       <ns0:has-email-address>K.Bontcheva@dcs…</ns0:has-…>
       …
       <rdf:type rdf:resource="http://…#Researcher-In-Academia"/>
    </rdf:Description>
```

The next phase is *summary structuring*. Here we will consider two alternatives: one where the has heuristic has been enabled and one where it was disabled.

When the heuristic is enabled, the system would know which properties of the given instance are attribute properties, because their names start with has. Therefore the user would not need to specify their lexicalisations and the existing ONTOSUM schemas can be applied to order the triples. As we took a simple example containing only attribute properties, their order will not change, that is, will remain as it was in the original input. However, as they are the same type of property, they will be aggregated into one semantic relation (ATTR) with several values – one for each property value. If there were other property types, then more semantic relations will be created and ordered according to the discourse schemas.

```
ATTR (Researcher-In-Academia: K.Bontcheva.dcs.shef.ac.uk,
      [    Appellation: Dr,
           string: K.Bontcheva@dcs.shef.ac.uk,
           string: + 4401142221930,
           string: http://www.dcs.shef.ac.uk/~kalina/
      ]
```

The information that Dr is a value of type Appellation and email, telephone, and URL are strings comes from the range restrictions in the property definitions in the ontology. Also, the lexical entry for `K.Bontcheva.dcs.shef.ac.uk` is used instead of the unique identifier from the ontology.

Given this input, the HYLITE + generator will verbalise it as:

```
Kalina Bontcheva has a Dr appellation, K.Bontcheva@dcs.shef..,
http://www.dcs.shef.ac.uk/~kalina/, and + 4401142221930.
```

The problem with this summary comes from the fact that the ontology engineer Appellation), while the rest is encoded as datatype properties with range `string` Since this is not an ontology class, the generator only provides its value (e.g., + 4401142221930), but it lacks the information that this is a telephone number, as this is only encoded implicitly in the property name – `has-telephone-number.`

One solution is to modify the ontology by introducing the required classes (Email, Telephone number, etc.) and changing the property ranges from string to these new classes. This would have the benefit of making explicit the semantics of these properties and their values. However, it may not always be desirable to modify the ontology.

The second solution is to provide the generator with manually written mapping rules which map the ranges of given properties to their lexical classes, for example, lex-mapping(has-email, string, email). Then, using these mappings, ONTOSUM will produce instead:

```
ATTR(Researcher-In-Academia: K.Bontcheva.dcs.shef.ac.uk,
      [      Appellation: Dr,
             email: K.Bontcheva@dcs.shef.ac.uk,
             telephone number: + 4401142221930,
             web page: http://www.dcs.shef.ac.uk/~kalina/

      ]
```

The disadvantage of the second approach is that it requires the user to create manually these mappings for each problematic datatype property. However, the number of such properties is often quite small and, in our experience, it is feasible to do that in cases when the ontology itself cannot be modified.

The third approach is to not define these properties as attribute properties, that is, to disable the has heuristic. In that case, ONTOSUM would use instead the lexicalisations of the properties themselves, derived automatically from their definitions. The disadvantage of this approach is that the structuring module will not be able to aggregate the four statements, as they will involve four different properties:

```
has-appellation(Researcher-In-Academia: K.Bntcheva.dcs.shef,
                     Appellation: Dr)
has-email-address(Researcher-In-Academia: K.Bontcheva.dcs.shef,
                     string: K.Bontcheva@dcs.shef…)
has-telephone-number(Researcher-In-Academia: K.Bontcheva.dcs…,
                     string: + 4401142221930)
has-web-address(Researcher-In-Academia: K.Bontcheva.dcs.shef,
                     string: http://www.dcs.shef.ac.uk/~kalina/)
```

Consequently, they will be verbalised as four separate sentences:

```
Kalina Bontcheva has a Dr appellation.
Kalina Bontcheva has email K.Bontcheva@dcs.shef.ac.uk.
Kalina Bontcheva has web page http://www.dcs.shef.ac.uk/~kalina.
Kalina Bontcheva has telephone number + 4401142221930.
```

In this case the information that K.Bontcheva@dcs.shef.ac.uk is an email address comes from the lexicalisation of the property `has-email-address`. The same is true for the other datatype properties.

However, while the problem with the implicit semantics of the datatype properties has been solved, the resulting summary is no longer so concise. One solution, to be implemented in future work, would be to implement a syntactic aggregation component which merges two sentences when they have the same subject and verb.

9.5 Using Ontology Mapping to Run ONTOSUM on Different Ontologies

Previously we discussed how ONTOSUM is adapted to a new ontology. However, frequently there is more than one ontology describing the same or similar domains (De Bruijn et al. 2004). For example, both the AKT and SWRC ontologies have concepts describing researchers, their publications, contact details, etc. Therefore, having customised ONTOSUM to the AKT ontology, instead of adapting it to SWRC from scratch, one could use *ontology mapping* rules (De Bruijn et al. 2004) to "translate" the SWRC instance descriptions into AKT ones and then run ONTOSUM without modifications.

In order to experiment with this approach, we designed manually a set of mapping rules for concepts and properties in the two ontologies. Some concept mappings are:

swrc:AssistantProfessor is mapped to akt:Lecturer-In-Academia, swrc:AssociateProfessor – to akt:Senior-Lecturer-In-Academia, etc.

Respectively, some property mappings are: swrc:name – akt: full-name, swrc:phone – akt:has-telephone – number, swrc:fax – akt:has-fax-number, swrc:homepage – akt:has-web-address. Some SWRC properties, for example, photo do not have a corresponding property in the AKT ontology. Therefore, no mapping was provided for them and, consequently, they are not included in the generated summaries.

Once defined, the mapping rules are applied to transform automatically instance descriptions from the SWRC to the AKT ontology, prior to sending them to ONTOSUM for generation. For example, the SWRC instance describing York Sure[3] looks as follows:

```
<rdf:Description rdf:about="http://www.aifb.uni-
            karlsruhe.de/Personen/viewPersonOWL#instance?id_db = 20">
    <rdf:type>
    <owl:Class rdf:about="&swrc;AssistantProfessor"/>
  </rdf:type>
  <swrc:name rdf:datatype="&xsd;string"> York Sure </swrc:name>
  <swrc:phone rdf:datatype="&xsd;string"> + 49 (0) 721 608 6592
  </swrc:phone>
  <swrc:fax rdf:datatype="&xsd;string"> + 49 (0) 721 608 6580
  </swrc:fax>
  <swrc:homepage rdf:datatype="&xsd;string">
    http://www.aifb.uni-karlsruhe.de/WBS/ysu
  </swrc:homepage>
  </rdf:Description>
```

After applying the mapping rules and removing properties for which no mappings exist, the system obtains:

```
<rdf:Description rdf:about="http://www.aifb.uni-
  kalsruhe.de/Personen/viewPersonOWL#instance?id_db = 20">
    <rdf:type>
     <owl:Class rdf:about="http…#Lecturer-In-Academia"/>
    </rdf:type>
    <akt:full-name rdf:datatype="&xsd;string"> York Sure </
    akt:full-name>
    <akt:has-telephone-number rdf:datatype = "…"> + 49 (0)
    721 608 6592
    </akt:has-telephone-number>
    <akt:has-fax-number rdf:datatype = "…"> + 49 (0) 721 608
    6580
```

[3] The author is grateful to York Sure for supplying the SWRC instance data.

```
</akt:has-fax-number>
<akt:has-web-address rdf:datatype="&xsd;string">
  http://www.aifb.uni-karlsruhe.de/WBS/ysu
</akt:has-web-address>
</rdf:Description>
```

When this input is passed to ONTOSUM, it generates the following textual summary, without requiring any customisation:

```
  York Sure has a telephone number + 49 (0) 721 608
6592, a fax number + 49 (0) 721 608 6580, and a web page
http://www.aifb.uni-karlsruhe.de/WBS/ysu.
```

The advantages of using ontology mapping to enable ONTOSUM to run on different ontologies are: *(1)* no ONTOSUM customisation is required by the user; *(2)* ontology mapping can be performed by ontology engineers and there are even some tools that automate parts of this process (De Bruijn et al. 2004).

A future extension of this approach would be to allow for more sophisticated mapping or even *ontology merging*, in order to enable ONTOSUM to verbalise also properties and concepts which do not exist in the original ontology. In this case, some limited customisation will be required, mainly concerned with providing new lexical information.

9.6 Discussion

This chapter presented several Natural Language Generation (NLG) techniques that produce textual summaries from Semantic Web ontologies. The main contribution of this work is in showing how existing NLG tools can be adapted to take Semantic Web ontologies as their input, in a way which minimises the customisation effort.

A major factor in the quality of the generated text is the content of the ontology itself. For instance, the use of string datatype properties with implicit semantics leads to the generation of text with missing semantic information. Three approaches to overcome this problem were presented here and users can choose the one that suits their application best.

In particular, one innovative aspect of the ONTOSUM approach, in comparison to previous NLG systems for the Semantic Web, is that it implements tailoring/personalisation based on information from the user's device profile. Most specifically, we developed methods for generating summaries within a given length restriction (e.g., 160 characters for mobile phones) and in different formats – HTML for browsers and plain texts for emails and mobile phones (Bontcheva 2005).

Another novel feature is its use of *ontology mapping* rules (De Bruijn et al. 2004) to enable users to run the system on new ontologies, with minimal customisation effort.

Future work will focus on the creation of a user-friendly tool for specifying new summary structuring schemas, because at present this is done directly in the generator's internal structures, which are hard to understand for non-specialists. Another strand of this work will aim at further investigation of the use of ontology mapping and merging in NLG systems.

Another major area for future work is system evaluation. NLG systems are normally evaluated with respect to their usefulness for a particular (set of) task(s), which is established by measuring user performance on these tasks, that is, *extrinsic* evaluation. This is often also referred to as *black-box* evaluation, because it does not focus on any specific module, but evaluates the system's performance as a whole. Therefore future work needs to carry out a qualitative evaluation of the textual summaries when they appear within a complete semantically-enabled knowledge management system.

Acknowledgement This work was supported by the IST Programme of the European Community under SEKT (IST-IP-2003-506826).

References

Aguado G, Bãnón A, Bateman JA, Bernardos S, Fernández M, Gómez-Pérez A, Nieto E, Olalla A, Plaza R, Sánchez A (1998) ONTOGENERATION: Reusing domain and linguistic ontologies for Spanish text generation. In Workshop on Applications of Ontologies and Problem Solving Methods, ECAI'98.

Bateman JA, Magnini B, Fabris G (1995) The Generalized upper model knowledge base: Organization and use. Towards Very Large Knowledge Bases, 60–72.

Bontcheva K (2005) Generating Tailored Textual Summaries from Ontologies. Second European Semantic Web Conference (ESWC'2005). Crete.

Bontcheva K, Wilks Y (2004) Automatic Report Generation from Ontologies: the MIAKT approach. In Nineth International Conference on Applications of Natural Language to Information Systems (NLDB'2004).

Bontcheva K, Tablan V, Maynard D, Cunningham H (2004) Evolving GATE to meet new challenges in language engineering. Natural Language Engineering, 10, 3–4: 349–373.

Dale R, Oberlander J, Milosavljevic M, Knott A (1998) "Integrating Natural Language Generation and Hypertext to produce Dynamic Documents", Interacting with Computers 11, 2: 109–135.

De Bruijn J, Martin-Recuerda F, Manov D, Ehrig M (2004) State-of-the-art survey on Ontology Merging and Aligning v1. (Technical report, SEKT project deliverable D4.2.1. http://sw.deri.org/ jos/sekt-d4.2.1-mediation-survey-final.pdf).

Mellish C, Sun X (2005) Natural language directed inference in the presentation of ontologies. In: Proceedings of the 10th European Workshop on Natural Language Generation, Aberdeen.

O'Donnell M, Knott A, Mellish C, Oberlander J (2001) ILEX: The architecture of a dynamic hypertext generation system. Natural Language Engineering, 7:225–250.

Power R, Scot D, Evans R (1998) What you see is what you meant: direct editing with natural language feedback. In: ECAI-98, Springer, Berlin Heidelberg, 677–681.

Rector A, Rogers J, Taweel A, Ingram D, Kalra D, Milan J, Gaizauskas R, Hepple M, Scott D, Power R (2003) Clef – joining up healthcare with clinical and post-genomic research. In Second UK E-Science"All Hands Meeting", Nottingham, UK.

Reiter E, Dale R (2000) Building Natural Language Generation Systems. Cambridge University Press

White M (2004) Reining in CCG chart realization. In Proceedings of the 3rd International Conference on Natural Language Generation (INLG-04).

White M (2006) Efficient realization of coordinate structures in Combinatory Categorial Grammar. Research on Language and Computation. 4, 1, (June 2006): 39–75(37).

White M, Baldridge J (2003) Adapting chart realization to CCG. In: Proceedings of the 9th European Workshop on Natural Language Generation.

Wilcock G (2005) An overview of shallow XML-based natural language generation. In: The Second Baltic Conference on Human Language Technolgies, Proceedings, Tallinn, Estonia, 67–78.

Chapter 10
Ontology Generation from Social Networks

Marko Grobelnik, Dunja Mladenić, and Blaž Fortuna

Abstract Analysis of social network data is gaining popularity with the increased availability of real-world data including data publicly available over the Internet such as publications and data resulting from interactions via social networking platforms and e-communication tools. In this chapter we present an approach to constructing light-weight ontologies from social network data. In relation to the more traditional

(semi-)automatic ontology learning techniques we reuse the approach typically used in learning ontologies from text (see Grobelnik and Mladenić(2006) for details) where the candidate instances and classes for the ontology are lexical items described by a set of attributes. We replace lexical items with nodes in the social network and attributes by descriptions of the node context in the graph. Similar techniques can then be applied for ontology learning either from text or from social networks.

To prove our claims we perform experiment on a real life dataset taken from a mid-size organization (700-800 people). The dataset represents log files from organizational spam filter software giving us the set of e-mail transactions for a period of 19 months resulting in 2.7 million successful e-mail transactions used here for analysis and ontology learning.

The main contribution of this work is an architecture consisting of five major steps that enable transformation of the data from a set of e-mail transactions inside an organization to an ontology representing the structure of the organization.

10.1 Introduction

Analysis of social network data is gaining popularity with the increased availability of real-world data including data publicly available over the Internet such as publications and data resulting from interactions via social networking platforms and e-communication tools. In this work we address the challenge of organizing information available in social networks and present an approach to constructing light-weight ontologies from social

M. Grobelnik (✉)
Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
e-mail: Marko.Grobelnik@ijs.si

J. Davies et al. (eds.), *Semantic Knowledge Management*,

network data. In relation to the more traditional (semi-)automatic ontology learning techniques we reuse the approach typically used in learning ontologies from text (see Grobelnik and Mladenić, 2006, for details) where the candidate instances and classes for the ontology are lexical items described by a set of attributes. In the work described here, we replace lexical items with nodes in the social network and attributes by descriptions of the node context in the graph. Similar techniques can then be applied for ontology learning either from text or from social networks.

To prove our claims we perform experiment on a real life dataset taken from a mid-size organization (700–800 people). The dataset represents log files from organizational spam filter software giving us the set of e-mail transactions for a period of 19 months resulting in 2.7 million successful e-mail transactions used here for analysis and ontology learning.

The main contribution of this work is an architecture consisting of five major steps that enable transformation of the data from a set of e-mail transactions inside an organization to an ontology representing the structure of the organization. Starting with log files from the institutional e-mail server where the data include information about e-mail transactions we perform data cleaning and remove irrelevant e-mail transactions. From the obtained set of e-mail transactions, we construct a graph where vertices are e-mail addresses that are connected if there is a transaction between them. The e-mail graph is then transformed into a sparse matrix and analyzed with ontology learning tools producing an ontological structure.

The rest of this chapter is organized as follows. Section 10.2 gives background of the proposed approach. Section 10.3 describes related work. The approach and its evaluation are described in Sect. 10.4. Some ideas for future work are given in Sect. 10.5.

10.2 Constructing Ontologies from Social Networks

Ontology learning techniques hitherto have dealt almost exclusively with the problem of extracting and modeling knowledge from text documents. The reason for this is that text is the most natural way of encoding information with attached semantics. But text is not the only data modality which could be modeled using ontological structures. In this work we show how to build the ontological models from social network data. In particular, we show the example of modeling organizational structure of an organization from its e-mail server log files.

10.2.1 The Main Idea

Approaches to ontology learning from text could be summarized in the following main steps: (a) first, extract candidates for future ontological instances and ontological classes, (b) next, describe them with appropriate attributes, and (c) finally with a set of heuristic rules or analytical approaches (e.g. similarity measures etc.)

determine the ontological elements and their connectivity. An important fact is that each element is described with a set of attributes which carry information about the element and which allow further processing. In the cases where we process text, this information usually identifies a set of words or phrases which characterize the element, their context within the text etc.

For learning ontologies from social networks we follow the same intuition as used for text. Since a social network is a general graph structure (vertices connected with directed or undirected edges), we redefine the above approach in the following way:

- Instead of textual lexical elements (words and phrases) we use graph vertices (points in the graph);
- instead of attributes describing properties of lexical items we describe vertices with a context where they appear in the graph;
- instead of heuristic and analytical rules applicable on text we propose and develop new ones appropriate for dealing with graph data.

Apart from the above three replacements all other operations can be shared between approaches for learning ontologies from text and from social networks.

10.2.2 Hypotheses and Contribution

There are two main hypotheses that we will aim to confirm. The first is that a graph (social network) data can be modeled by ontological structures (taxonomies). The second is that in a research institution e-mail data roughly correspond to organizational structure on an institution.

The main contributions of this work are the following. Transformation of social network data into the form suitable for ontology learning (sparse vector representation). Example of a light-weight ontology construction from e-mail server log data. Evaluation of the constructed model.

10.3 Related Work

Social network analysis is a relatively mature field – a good overview of the traditional methods and approaches in the field is described in (Wasserman and Faust 1994). With the advent of the Web and consequently increased interest in network related topics, the whole area attracted increased interest – areas such as Machine Learning, Data Mining, Semantic Web, Information Retrieval etc. recognized benefits coming out of the use of graph data being in the most cases social network data (graphs constructed in social processes). Maybe one of the most prominent everyday usages of information coming from the network is ranking web search results which partly led to success of the Google search engine.

Current research on social networks is centered on problems such as, how to handle dynamics of networks; visualization of very large networks; creation of generative models to explain underlying laws of network generation; interchanging network data with other data modalities (such as text, images etc); and different applications on top of network data, such as modeling the spread of an influence, modeling trust, improving search results, collaborative methods etc. Addressing network dynamics is gaining popularity especially in connection to the Internet graph, modeling its evolution (Albert and Barabasi 1999) and investigating different properties such as power-law degree distribution (Faloutsos et al. 1999) and shrinkage diameter phenomena (Leskovec et al. 2005). The dynamics of social network are commonly connected to study of community identification (Kumar et al. 1999) and evolution (Backstrom et al. 2006; Aggarwal and Yu 2005).

Our work fits into the category of problems where the goal is to find structure in large networks. Such identified structure would reflect the underlying processes which generated the network – this would allow us to detect and possibly explain phenomena within the network.

10.4 Approach Description

Our approach to ontological modeling of e-mail social networks consists from the following five steps shown in Fig. 10.1:

1. Initially we start with log files from the institutional e-mail server. The data include information about e-mail transactions with three main fields: time of the transaction, sender e-mail address and the list of receiver e-mail addresses.
2. After cleaning of the log files we get the data in the form of e-mail transactions which include e-mail addresses of sender and receiver. This cleaning phase removes non-valid and non-relevant e-mail transactions.
3. From the set of e-mail transactions we construct a graph where vertices are e-mail addresses and two such vertices are connected if there exists an e-mail transaction between them.

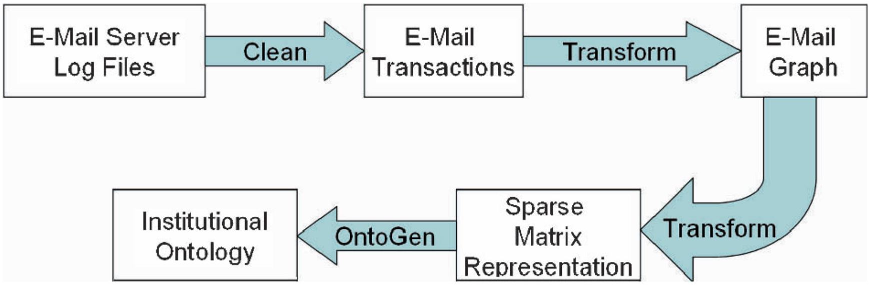


Fig. 10.1 Diagram showing transformation of the e-mail server log data into an ontological representation of institution

4. The e-mail graph is transformed into a sparse matrix, where *(i, j)* element of the matrix is non-zero if *i*-th and *j*-th vertices are connected directly or indirectly over the number of hops given as a parameter.
5. The sparse matrix representation of the graph is analyzed with ontology learning tools producing an ontological structure roughly corresponding to the organizational structure of the institution where the e-mails came from.

10.4.1 Data Description

The data used in this experiment is a collection of log files with e-mail transactions from a middle size institution spam filter software Amavis (http://www.amavis.org/). Each line of the log files denotes one event at the spam filter software. We were interested in the events on successful e-mail transactions which include information on time when the event happened, who sent the e-mail (sender), and who received it (a list of receivers). An example of a successful e-mail transaction is the following line:

```
2005 Mar 28 13:59:05 patsy amavis[33972]: (33972-01-3) Passed
CLEAN, [217.32.164.151] [193.113.30.29] < e-mail addr1 > - >
<e-mail addr2>, Message-ID:<
21DA6754A9238B48B92F39637EF307FD0D4781C8@i2km41-
ukdy.domain1.systemhost.net>, Hits: -1.668, 6389ms
```

The above line tells us that there has been an e-mail sent from `e-mail addr1` to `e-mail addr2 (real addresses suppressed for anonymity)` at 2005 March 28 13:59:05. The advantage of using spam filter log files is additional information provided by spam filter software which tags each e-mail transaction as being "CLEAN" (the example above) or "SPAM."

The log files include e-mails data from September 5, 2003, to March 28, 2005, which sums up to 12.8 Gb of data. After filtering out successful e-mail transactions there is 564 Mb of data, which contains ~2.7 million of successful e-mail transitions to be used for further processing. The whole dataset contains references to ~45,000 e-mail addresses. After the data cleaning phase, the number is reduced to ~17,000 e-mail addresses out of which 770 e-mail addresses are internal inside the home institution.

10.4.2 Data Cleaning

Spam filtering software log files records of all the successful e-mail transaction events – it doesn't control the existence of the e-mail addresses, successful delivery etc. It also includes all kinds of automatically generated email messages from mailing-lists, notifications etc. In general, for the analysis of institutional e-mails we are not interested in all the transactions which do not contribute to the overall goal of modeling the organizational structure in an institution. Therefore we perform three kind of data cleaning operations:

- Deleting all e-mail transactions which include e-mail addresses with escape and unusual characters. This operation deletes most of the automatically generated e-mail messages from e.g. notification or other similar services. An example of such e-mail address is a notification from Yahoo Groups.
- Deleting all e-mail transactions where the pair < sender, receiver > appears less then a certain number of times (in our experiments we use the value 10). This filter deletes most the typos in e-mail addresses.
- Deleting all e-mail transactions with e-mail addresses which communicate with less then a certain number of different e-mail addresses (set to 10 in our experiments). This filter removes e-mail addresses which are rarely in use.

10.4.3 Data Transformation

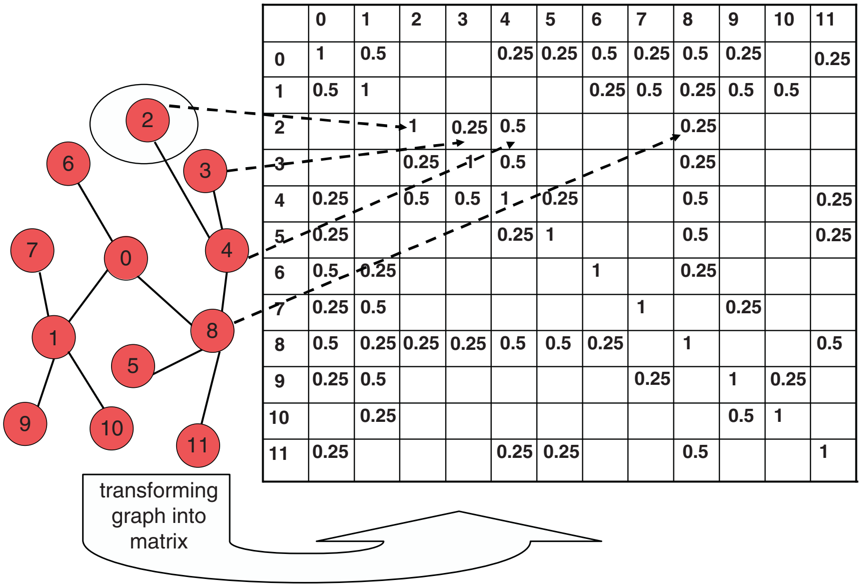
In the process of processing the original log data toward ontological representation there are two significant data transformation we have applied (see Fig. 10.1). Here we describe them in detail.

The first transformation transforms a collection of e-mail transactions into an e-mail graph. From the set of transactions we construct a graph where vertices are e-mail addresses and edges between vertices represent communication between the e-mail addresses. In other words, there is an edge between two vertices representing two e-mail addresses if there are e-mail transactions between them. Edges are additional labeled with the intensity of communication (number of transactions between e-mail addresses representing both vertices).

The second transformation transforms an e-mail graph into a sparse matrix. Here we use a representational trick proposed in (Mladenić and Grobelnik 2004) which helps to compare and calculate similarity between vertices in the graph represented as rows in the sparse matrix. In order to get sparse vectors from a graph, we represent a graph with N vertices as an NxN sparse matrix. The matrix is constructed so that the *Xth* row gives information about the *Xth* vertex and has nonzero components for the columns representing vertices from the neighborhood of vertex *X*. We have defined neighborhood of a vertex to contain all the vertices at the distance of up to *d* steps from the vertex. Consequently, the *Xth* row has non-zero component in the *Xth* column and all the other columns that represent the neighbors at step *1, 2, 3, … d*. Intuitively, the *Xth* row numerically represents the neighborhood of the *Xth* vertex within the graph, with the values calculated using the following formula $1/2^d$, where *d* is the distance in the number of steps from the *Xth* vertex. Figure 10.2 illustrates the graph transformation on an example graph assuming all the edges in the graph have weight 1.

Similarity between vertices (i.e. rows in the sparse matrix) is calculated by using traditional cosine similarity, a measure commonly used in information retrieval and text-mining. Cosine similarity between the two vectors D_1 and D_2, having elements x_{1i} and x_{2i} respectively, is defined in Eq. 10.1.

$$Sim\left(D_1, D_2\right) = \frac{\Sigma_i x_{1i} {}^{i} x_{2i}}{\sqrt{\Sigma_j x_j^{2}}\sqrt{\Sigma_k x_k^{2}}} \tag{10.1}$$



	0	1	2	3	4	5	6	7	8	9	10	11
0	1	0.5			0.25	0.25	0.5	0.25	0.5	0.25		0.25
1	0.5	1					0.25	0.5	0.25	0.5	0.5	
2			1	0.25	0.5				0.25			
3			0.25	1	0.5				0.25			
4	0.25		0.5	0.5	1	0.25			0.5			0.25
5	0.25				0.25	1			0.5			0.25
6	0.5	0.25					1		0.25			
7	0.25	0.5						1		0.25		
8	0.5	0.25	0.25	0.25	0.5	0.5	0.25		1			0.5
9	0.25	0.5						0.25		1	0.25	
10		0.25								0.5	1	
11	0.25				0.25	0.25			0.5			1

Fig. 10.2 Illustration of the graph transformation into a sparse matrix where the rows represent instances (vertices) and columns represent neighborhood with weights relative to the distance from the vertex in that row. Here we have set the maximal distance to $d = 2$ (i.e., vertices further than two hops away are ignored). Notice that the diagonal elements have weight 1 (showing that each vertex is in its own neighborhood). The dashed lines point out neighboring vertices and the corresponding weights for the vertex labeled as 2, which has four non-zero elements in its sparse vector representation (1, 0.25, 0.5, 0.25) corresponding to four vertices (labeled in the graph as 2, 3, 4, 8). The ilustration is from Mladenić and Grobelnik (2004)

Having an e-mail graph represented in the form where we can efficiently compare vertices, allows us to use most of the methods for unsupervised learning (clustering) which is the natural basis for ontology learning approaches.

10.4.4 Ontology Modeling

For ontological modeling of the data we used the ontology learning tool OntoGen (Fortuna et al. 2005a). The tool can handle data represented as a set of feature vectors describing properties of ontological instances. Using several machine learning techniques (the most prominent being k-means clustering, latent semantic indexing, support vector machines, uncertainty sampling and active learning) OntoGen helps the user to construct an ontological structure directly from the data by giving proposals and analyzing user's decisions.

As the input for ontology learning is the set of ontological instances (represented as sparse vectors) – in our case each instance corresponds to one e-mail address within the organization. On the output we want to get an ontology in the form of taxonomy modeling relation "subcommunity-of" which would correspond to the organizational structure of the institution where the e-mail data is coming from.

The actual experiment consisted from 770 e-mail addresses from Jožef Stefan Institute. Each of these e-mail addresses was described with the subset of 17,000 e-mail addresses being in the direct or indirect contact with the target e-mail address. Each e-mail address was represented as sparse vector from the e-mail graph.

The result of ~10-min session with OntoGen tool using k-means clustering is shown in Fig. 10.3. The only additional information included was the labels of communities based on the organization unit names associated to e-mail addresses.

10.4.5 Visualization

One possible way of presenting complex data is to use techniques for visualization of high dimensional data. These typically include the major step of reducing high dimensionality of the observed data down to two or three dimensions while preserving the high level relationships between the data points. In the same way we can visualize the e-mail transactions data. We use the tool Document Atlas (Fortuna et al. 2005) for document visualizations. Since the sparse matrix representation (described in the previous sections) corresponds to typical bag-of-words representation (used in Document Atlas), we can use Document Atlas also for visualization of e-mail data.

Figure 10.4 shows the anonymized dataset presented in the previous section in the geographical relief visualization. Each data point represents one e-mail address, similar e-mail addresses are closer on the image and the density of points results in higher elevation of the terrain. From the image it can be seen the top level structure of the e-mail data, corresponding to top level concepts in the generated ontology. We can see four major "mountains" representing four major topics of the institute: computer science, chemistry, physics and administration in the middle serving as a connector between the areas.

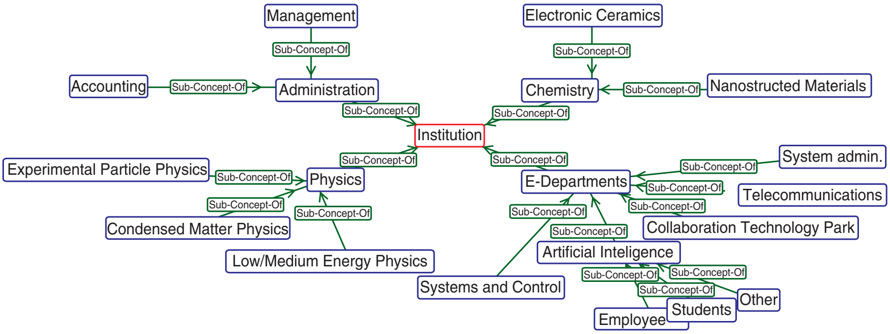


Fig. 10.3 Organizational structure of research Institute modeled from the e-mail data in a 10-min session with the ontology learning system OntoGen

10.5 Evaluation

We perform evaluation of the approach described in the previous sections indirectly by showing the compactness of the clusters when clustering the e-mail addresses by similarity. The main hypothesis we want to prove is that communication intensity follows organizational structure – in other words, people inside the same department are communicating more intensively between each other then to the people outside its organizational unit. We also hope to identify some exceptions spotting communities that are not captured in the formal organizational structure. We perform a “golden standard” style of comparison, where we compare the clustering as underlying method in OntoGen to the formal organizational structure.

Table 10.1 shows the result of 10-means clustering, where columns correspond to the clusters and rows correspond to the known organizational units. For each cluster we give the percent of e-mail addresses falling into the individual group – the sum of each column is 1. The highest number in each column is followed by exclamation mark (!).Table 10.1 shows that individual clusters actually contain e-mails belonging mainly to one of the departments. For instance, cluster C-0 contains mainly e-mail addresses associated to formal unit IT8, C-1 mainly from IT2, etc. In fact, 36.8% of all the e-mail addresses in cluster C-0 are from IT8, 21% of e-mail addresses are from unit B and 15% are from unit IT9. Knowing background this is not surprising, as IT8 and IT9 work on related topics in information technologies. Having such a large participation of people from unit B (Biology) in the mainly

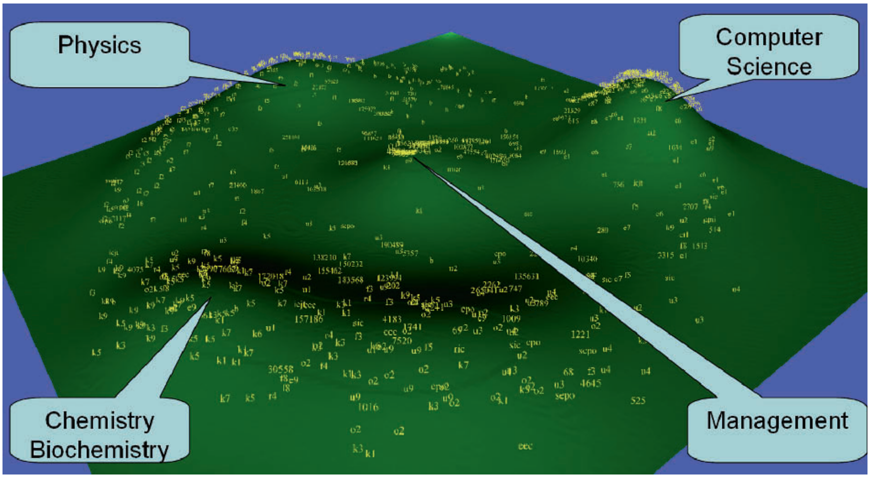


Fig. 10.4 Visualization of the Institute e-mail data as geographical terrain. The high level structure of the data shows four major areas on the map corresponding to the four major groups of the scientific departments at the institute (computer science, chemistry, physics and administration)

Table 10.1 Part of the clustering results into ten clusters (C-0, C-1, …, C-9) showing distribution of the clustered e-mails over the formal organizational structure (IT8, K3, B, IT2, ….)

	C-0	C-1	C-2	C-3	C-4	C-5	C-6	C-7	C-8	C-9
IT8	0.368!			0.03						
K3			0.04	0.02			0.1	0.04	0.06	0.01
B	0.21	0.05	0.01		0.03		0.361!	0.07	0.04	0.07
IT2		0.381!				0.01				
K5	0.01		0.01	0.448!	0.05	0.01				
F5		0.03	0.03		0.444!	0.15	0.01	0.347!	0.17	
IT9	0.15		0.01	0.02			0.01	0.01		

information technology community triggered further investigation on that community. It turned out that it is due to bioinformatics area being one of the research topics of that community.

10.6 Future Work

In the future work we plan to address temporal component of the e-mail data. This would give us insight into dynamic nature of a life of an organization, it would allow us to better understand social phenomena (such as getting new project, employing new person, estimating social value of a person etc.) as well as trying to predict future phenomena based on the past experience.

If possible we will try to include additional knowledge and information about the organization, types of people etc. This would give us possibility to model relations between the people and detect types of people. E.g. it would be possible to make a model of a secretary and associate other people in the organization with this model to see how the communication patterns follow certain profile.

Acknowledgments This work was supported by the Slovenian Research Agency and the IST Programme of the European Community under SEKT (IST-IP-2003-506826), NeOn Lifecycle Support for Networked Ontologies (IST-4-027595-IP), TAO (IST-2004-026460) and PASCAL Network of Excellence (IST-2002-506778).

References

Aggarwal C, Yu P (2005) Online analysis of community evolution in data streams. In Proceedings of ACM SIAM on Data Mining.

Albert R, Barabasi A L (1999) Emergence of scaling in random networks. Science, vol. 286, no. 5439, 509–12.

Backstrom L, Huttenlocher D P, Kleinberg J M, Lan X (2005) Group formation in large social networks: membership, growth, and evolution. In Proceedings of 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Faloutsos M, Faloutsos P, Faloutsos C (1999) On power-law relationships of the internet topology. In SIGCOMM.

Fortuna B, Mladenić D, Grobelnik M (2005) Visualization of text document corpus. Informatica (Ljublj.), vol. 29, no. 4, 497–5022. http://docatlas.ijs.si.

Fortuna B, Mladenić D, Grobelnik M (2005a) Semi-automatic construction of topic ontology. Proceedings of the 8th International Multiconference Information Society IS 2005. Ljubljana: Institut "Jožef Stefan", 170–173. http://ontogen.ijs.si.

Grobelnik M, Mladenić D (2006) Knowledge discovery for ontology construction. In Semantic web technologies: trends and research in ontology-based systems. Chichester: Wiley, 9–27.

Kumar R, Raghavan P, Rajagopalan S, Tomkins A (1999) Trawling the Web for emerging cyber-communities. In Proceedings of the Eighth World Wide Web Conference.

Leskovec J, Kleinberg J, Faloutsos C (2005) Graphs over time: densification laws, shrinking diameters and possible explanations. In Proceedings of 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Mladenić D, Grobelnik M (2004) Visualizing very large graphs using clustering neighborhoods. In: Morik et al. (eds.), Local Pattern Detection: International Seminar. Dagstuhl Castle, Germany, April 12–16, revised selected papers, Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence), vol. 3539, State-of-the-art survey. Springer Berlin Heidelberg: 89–97.

Wasserman S, Faust K (1994) Social network analysis: methods and applications. In: Structural Analysis in the Social Sciences. New York: Cambridge University Press.

Chapter 11
Capturing Document Semantics for Ontology Generation and Document Summarization

David Baxter, Bryan Klimt, Marko Grobelnik, David Schneider, Michael Witbrock, and Dunja Mladenić

Abstract When dealing with a document collection, it is important to identify repeated information. In multi-document summarization, for example, it is important to retain widely repeated content, even if the wording is not exactly the same. Simplistic approaches simply look for the same strings, or the same syntactic structures (including words), across documents. Here we investigate semantic matching, applying background knowledge from a large, general knowledge base (KB) to identify such repeated information in texts.

Automatic document summarization is the problem of creating a surrogate for a document that adequately represents its full content. Automatic ontology generation requires information about candidate types, roles and relationships gathered from across a document or document collection. We aim at a summarization system that can replicate the quality of summaries created by humans and ontology creation systems that significantly reduce the human effort required for construction. Both applications depend for their success on extracting the essence of a collection of text. The work reported here demonstrates the utility of using deep knowledge from Cyc for effectively identifying redundant information in texts by using both semantic and syntactic information.

11.1 Introduction

When dealing with a document collection, it is important to identify repeated information. In multidocument summarization, for example, it is important to retain widely repeated content, even if the wording is not exactly the same. Simplistic approaches just look for the same strings, or the same syntactic structures (including words), across documents. Frequently, however, although more or less the same

M. Witbrock (✉)
Cycorp, Inc., 7718 Wood Hollow Drive, Suite 250, Austin, Texas 78731, USA
e-mail: witbrock@cycorp.com

J. Davies et al. (eds.), *Semantic Knowledge Management*,

information is present in multiple documents, it cannot be identified by these simple methods. For example, “the president arrived at the Kremlin” and “Bush lands in Moscow” convey substantially overlapping information but have no lexical overlap whatsoever. Here we investigate semantic matching, applying background knowledge from a large, general knowledge base (KB) to identify and benefit from such repeated information in texts.

Automatic document summarization is the problem of creating an adequate surrogate representation for the full content of a document. Automatic ontology generation requires information about candidate types, roles and relationships gathered from across a document or document collection. We aim at a summarization system that can replicate the quality of summaries created by humans and ontology creation systems that significantly reduce the human effort required for construction. Both applications depend for their success on extracting the essence of a collection of text. In previous work (Leskovec et al. 2005; Fortuna et al. 2006) candidate sentences for summarization and candidate relationships for ontology generation in OntoGen were identified by analyzing a semantic graph of syntactic and shallow semantic relationships between terms extracted from sentences. Unfortunately, these extracted relationships produce spurious graph structure based on differences of phrasing, rather than on meaning, reducing the accuracy with which relationship importance can be estimated, and potentially interfering with the quality of summarization or ontology generation based on these relationships. In the work reported here, we seek to improve the production of semantic graphs from text by identifying repeated semantic structure using a large background knowledge base. In order to improve coverage of that background knowledge base, we also report on a method, independent of OntoGen, of fully automated ontology generation based on mechanical reading of sentences from Wikipedia articles.

OntoGen (Fortuna et al. 2005, 2006) is a semiautomatic, data-driven system that aids interactive users constructing topic ontologies. OntoGen suggests type concepts and relations, and their names, automatically assigns instances to types and supports users’ understanding of the growing ontology via browsing and graph visualization. The user retains full control of the ontology, and can manually add or delete types and relations and reassign instances to types. The system is data-driven; most of the aid it provides is based on input – usually a domain-relevant document collection – provided by the user. Instances and instance co-occurrences are extracted from the data together with the context in which they occur. OntoGen currently uses a standard bag-of-words document representation as the input for its knowledge discovery processing (Grobelnik and Mladenić 2005). The work reported here goes beyond the bag-of-words, applying deep background knowledge during ontology generation.

The rest of this chapter is structured as follows. We first provide a brief approach description in Sect. 11.2, followed by the description of the process for building equivalence classes in Sect. 11.3. Since the approach depends heavily on both the lexical and semantic reach of the background knowledge base, and since both are incomplete, even when using a very large KB like Cyc, Sect. 11.4 describes automated means for extending Lexical coverage, and Sect. 11.5 Ontological coverage of the kinds required for the application. The chapter concludes with ideas for future research directions, in Sect. 11.6.

11.2 Approach

We propose an approach where as input the system takes "factoid" triples identified in textual documents as proposed in (Leskovec et al. 2005). These triples represent, roughly, the Subject, Verb, and Direct Object from each sentence. Morphological variants are regularized using lemmatization, and anaphora resolution and name consolidation are applied as described in (Leskovec et al. 2004). Experiments used a corpus of news documents discussing events from around 1991, including two events prominent in the American media: the Rodney King incident and the appointment of Clarence Thomas to the US Supreme Court.

To allow for semantic consolidation of triples, in the current work, their elements were mapped both to concepts from the Cyc vocabulary (Matuszek et al. 2006; Lenat, 1995; http://opencyc.org) using the Cyc NL subsystem (Shah et al. 2006) and to WordNet (http://wordnet.princeton.edu) synonym sets (synsets) (Fellbaum 1998). Having mapped triple elements to semantic classes, the system assembles similar triples into equivalence classes, and assigns scores to these equivalence classes reflecting the degree of similarity among their triples. Broad coverage is achieved by using several kinds of lexical and ontological background knowledge to identify likely semantic redundancy in lexical triples, and by partially automating the process of obtaining both kinds of required background knowledge.

11.3 Building Equivalence Classes

Triples that might be grouped together are identified by associating each string with a set of patterns it matches in the semantic lexicon portion of the Cyc KB, ranging in specificity from exact string match at the most specific, to the "anything" pattern at the most general. In between are "denotes" patterns, which match all strings that denote the same concept (i.e., synonyms), "implies" patterns, which match strings whose denotations are in some sense subsumed by the same concept (e.g., "cat" and "dog", under the Cyc term `#$Animal`), and "taxonomy" patterns, which group together strings whose denotations belong to the same subject-specific taxonomy inside the Cyc ontology. To summarize, the following lists techniques for matching using Cyc's lexicon sorted by strength:

- **exact match** – exact string match
- **denotes** – all strings that denote the same concept
- **implies** – strings denotation subsumed by the same concept
- **taxonomy** – string denotation belonging to the same subject-specific taxonomy
- **anything** – the most general concept

To find the possible denotations of strings, we use a combination of Cyc's English lexicon (Panton et al. 2002), including inexact #$denotationRelatedTo mappings, and WordNet. Since some concepts are represented in both systems, some spurious ambiguity results, mitigated, as described below, by the partial mapping between Cyc terms and WordNet synsets contained in the Cyc KB. With both lexica, interpretations are filtered by the limited morphological and syntactic clues derivable from the position of a string in its triple. The first string in a triple, which is generally the subject of a verb, is only given interpretations consistent with its use as a proper name or a common noun. The second string is interpreted as an infinitive form of a transitive verb or copula, and the last string as a proper name, common noun, or adjective.

From patterns for individual strings, patterns for triples are constructed and scored based on their specificity (taken as the average of the specificities of the component string patterns) and the number of triples in a document, document group, or corpus that match them. Triples that match high-scoring patterns are clustered together as candidates for treatment as a unit in later processing for document summarization on ontology generation.

11.3.1 Patterns

Every string matches at least two patterns: an exact match to the string itself, and the "anything" pattern, which matches every string. In addition, for strings found in either the Cyc lexicon or WordNet, a set of "denotes" patterns is generated, one pattern for each synset or Cyc term the string might denote in the context in which it appears. Some WordNet synsets have been linked to Cyc terms, and where there is a string that denotes both a synset and its linked term, they are collapsed to just use the Cyc term. A named-entity recognizer (Krishnan and Manning 2006) is used to identify probable person names and membership in other broad classes.

From each "denotes" pattern, progressively more general patterns are generated, which might match a larger number of strings from the document, group, or corpus. For Cyc terms, collections to which the term belongs via either the `#$isa`(instance) relation or the `#$genls` (subset) relation, are identified; for geographical terms, the larger regions (e.g., country, continent, subcontinent, etc.) where it is found are also used. In both cases, terms that are overly general are omitted, according to Cyc's built in generality-estimate function. Finally, topical taxonomies (e.g., `#$FoodAndDrinkTaxonomy, #$MilitaryWeaponTaxonomy`) that include the terms are identified.

Cyc collections such as `#$Opaque` (the collection of all opaque things) and `#$ThreeDimensionalThing` are also generalizations of `#$PorkChop`, but are omitted from further processing because their generality is above the threshold – there are too many things that they could cluster with. For WordNet synsets, the system ascends two links in the "hypernyms" graph to get more general synsets; proceeding further retrieved overly general concepts.

The patterns for “pork chop” in order of increasing specificity are:

ANYTHING
(IMPLIES FoodOrDrink)
(TAXONOMY FoodAndDrinkTaxonomy)
(IMPLIES Food)
(IMPLIES MeatAndLegumeFood)
(IMPLIES CommonlyAvoidedFood)
(IMPLIES Meat)
(IMPLIES Meat-HumanFood)
(IMPLIES (MeatFn Mammal))
(IMPLIES Pork)
(IMPLIES PorkChop)
(IMPLIES Chop-MeatCut)
(DENOT PorkChop)
“pork chop”

Each pattern is assigned a specificity score ranging from0.01 for the “anything” pattern to 1.0 for exact string matching patterns. “Implies” pattern scores for Cyc terms are in the range 0.2–0.8 and are based on the terms’ generality estimate’, and for WordNet synsets either on the generality estimate of Cyc terms in their hypernym graph or else on the estimated number of hyponyms they have. All “denotes” patterns have a value of 0.9 because any strings matching them are synonyms, and thus, *prima facie*, equally valid candidates for merging for the purposes of document summarization. The patterns for “wealth”, including the use of WordNet synsets, shown in italics, are:

0.010: ANYTHING
0.362: (IMPLIES *N00024568: the way something is with respect to its main attributes*)
0.427: (IMPLIES *N04827181: an adequate or large amount*)
0.427: (IMPLIES *N04826612: how much of something is available*)
0.445: (IMPLIES N12489711: property | belongings | holding | material possession)
0.488: (IMPLIES Quantity)
0.500: (IMPLIES *N12596037: assets in the form of material possessions*)
0.523: (IMPLIES SystemCondition)
0.539: (IMPLIES Possession)
0.563: (IMPLIES ownsWealth)
0.578: (IMPLIES Assets-Possession)
0.592: (IMPLIES *N04832088: abundance | copiousness | teemingness*)
0.612: (IMPLIES MonetaryValue)
0.725: (IMPLIES GreatWealth)
0.725: (IMPLIES FinancialCondition)
0.900: (DENOT (SubcollectionOfWithRelationFromTypeFn MonetaryValue ownsWealth SocialBeing))
0.900: (DENOT GreatWealth)
0.900: (DENOT ownsWealth)
0.900: (DENOT *N13667176: wealth | wealthiness*)
0.900: (DENOT *N04832842: the quality of profuse abundance*)
0.900: (DENOT *N12596161: wealth | riches*)
0.900: (DENOT *N12495796: property that has economic utility: a monetary value or an exchange value…*)
1.0: “wealth”

Patterns for string triples are constructed using the string patterns for the first, second, and third positions. Their pattern specificity (ps) is computed as the average of the specificities of the component string patterns as follows:

```
ps(S V O) = Σ(ps(S) + ps(V) + ps(O) )/3
if (match_type = strings) ps(x) = 1.0
if (match_type = other) ps(x) = 0.01
if (match_type = denotation) ps(x) = 0.9
if (match_type = implication) ps(x) = relative_term_specificity[1](implication(x) ) * 0.6 + 0.2
```

Using this scoring mechanism, a pair of triples with identical strings would get the highest possible equivalence score, those varying in one element would get lower scores, and those with two or three varying elements would generally get even lower scores.

11.3.2 Clustering Triples in Documents

For triples with more than one relatively ambiguous word, using all possible denotations for each string with all the more general patterns for each denotation – even with a generality cutoff – can produce a huge number of mostly improbable or unhelpful patterns. To avoid this, an extremely simple-minded disambiguation heuristic was implemented that prefers denotations that take part in larger numbers of patterns, within a corpus.

Clustering of triples operates over input preprocessed by into triples and produces HTML output for human inspection as well as XML files suitable for downstream processing. An example XML element corresponding to a single cluster:

```
<equiv_class weight=‶“0.774”″”″ doc_id=‶“d58k.LA073089-0118”
<pattern>
   <token type=‶“string-match”″ value=‶“hundred”″/>
   <token type=‶“string-match”″ value=‶“feast”″/>
   <token type=‶“implies”″ value=‶“Shellfish”″/>
  </pattern>
  <matching_triples>
   <triple first=‶“hundred”″ second=‶“feast”″ third=‶“oyster”″/>
   <triple first=‶“hundred”″ second=‶“feast”″ third=‶“shrimp”″/>
   </matching_triples>
  </equiv_class>
```

This equivalence class has a weight, or confidence, of 0.774 out of 1.0, and groups together the triples “`hundred | feast | oyster`” and “`hundred | feast | shrimp`” based on exact string matches for “`hundred`” and “`feast`”, and the fact that “`oyster`” and “`shrimp`” both denote kinds of shellfish.

[1] Relative_term_specificity is an ad hoc measure specific to the Cyc KB of, approximately, how deep in the KB the term lies. Its values lie in the range [0.0 1.0].

11.3.3 Explaining Pattern Matches

Occasionally it is not obvious why a particular string is judged to match a particular pattern. Patterns of the "implies" type are most commonly in need of such justification. The system can expose the chain of reasoning behind such matches, for debugging or improvement purposes. For instance, it is, at first confusing to notice that the system believes that "raise" implies the Cyc term `#$IBTGeneration-Original`. Its justification reveals that this is because "raise" belongs to the synset V00910165, whose gloss is "cause to be heard or known; express or utter", which has the hypernym *V00909455*, including the strings *"express* | *verbalize* | *verbalise* | *utter* | *give tongue to"*, which is linked to the Cyc term `#$IBTGeneration-Original` (the production of an "Information Bearing Thing" generally). This is probably not the intended sense of "raise", but it is at least plausible and understandable, and points to the need for sense disambiguation in a fully developed system.

Further development of the triple clustering functionality would ideally be informed by data reflecting its impact on the behavior of a larger document summarization system. We expect that it would be improved by an expansion and correction of the existing WordNet-to-Cyc mappings, and by using a more sophisticated disambiguation approach. For example, when disambiguating a noun in the final position of a triple, it may be useful to take into consideration the likely interpretations of the verb in the second position to identify the most plausible complements of that verb. When used in the context of a functioning system with a training corpus, it is also likely that many of the combinations of weights and generality estimates and limits assigned manually or heuristically in the current system could be tuned by machine learning techniques.

11.3.4 Results

In one run of the system, a total of 299 documents containing 20,365 triples (avg 68.11 triples/doc) were analyzed. Of these, 1,835 (9%) were placed into equivalence classes, within documents, on the basis of combined Cyc and WordNet semantics, as summarized in Table 11.1. Precision for the three highest-ranked sets of clusters was estimated by manually evaluating 30 of the equivalence classes in each set to determine whether or not they contained roughly the same information. Recall was not calculated due to the expense of producing a gold-standard equivalence set.

The equivalence classes that scored higher than 800 typically contained one or two strings that matched exactly, and one or two of the "*denote*" patterns, where the words are synonyms of each other (see examples in Table 11.2).

The equivalence classes that scored between 750 and 799 typically contained on "*implies*" pattern and two string-match patterns (see Table 11.3 for examples).

Table 11.1 Cluster scores with the number of clusters their average number of tuples per cluster and precision

Cluster score	# Clusters	Avg. tuples/cluster	Precision (est) (%)
800+	46	2.043	93
750–799	125	2.176	79
700–749	327	2.223	44
650–699	225	2.107	Unknown
600–646	14	2.000	Unknown
550–599	94	2.106	Unknown
500–549	18	2.278	Unknown

Table 11.2 Two examples of triple classes with their corresponding members

Triple Argument 1	Triple Predicate	Triple Argument 2
"Scripps Clinic"	:denotes Developing AProduct	"drug"
Scripps Clinic	Develop	Drug
Scripps Clinic	Produce	Drug
:denotes (WN: delicacy …)	"be"	"essential"
diplomacy	Be	Essential
Finesse	Be	Essential

Shaded rows correspond to triple classes whose members have been consolidated using the described semantic matching techniques. Rows without shading contain the actual lexical triples that the classes comprise. The triples in this table are grouped on the basis of shared semantic denotation. Table 11.2 shows the use of slightly more remote semantion relationships in the Cyc knowledge base

Table 11.3 Examples of equivalence classes with lower equivalence scores than those in Table 11.2, whose format is repeated here. The relationship unifying the terms in the lexical triples is more remote in this case, relying on semantic relationships (implications) rather than simple shared denotations

Triple Argument 1	Triple Predicate	Triple Argument 2
:implies #$StoneStuff	"be"	"dense"
Magma	Be	Dense
Rock	be	Dense
"casino"	"offer"	:implies SportsPlayingArea
Casino	Offer	Bowling alley
Casino	Offer	Golf course

11.4 Lexical Coverage

Of the 11,804 distinct string-plus-position pairs in this corpus, we were unable to identify denotations from the Cyc lexicon, WordNet, or the named-entity recognizer for 1,584, or 13.4%. Without using WordNet (which recognized 2,121 otherwise unrecognized strings), or the named-entity recognizer (NER), which recognized 1,125, the system, relying on Cyc's lexicon alone, would have failed to find denotations for 40.9% of the terms (see Table 11.4 for details).

As the system has no way to cluster the completely unknown strings other than by literal matching, so a separate component described in Sect. 11.5 was developed to place them automatically as concepts within the Cyc ontology, with lexical information, making them available for clustering in future iterations.

11.5 Automatic Ontology for Unknown Phrases

To increase the overall system's ability to identify equivalent triples, the Cyc KB was expanded by automatically adding simple taxonomic and lexical information about unknown words and phrases from the input set to the ontology using sentences from Wikipedia. Figure 11.1 provides schematic description of the process.

In an experiment, the system was provided with a list of 3,498 phrases from the input triples that an early version of the equivalence-class system had been unable to understand (notably, that version did not include WordNet support). For each such uninterpreted string, the system starts by searching for a Wikipedia page with the string as its title. Using a downloaded a copy of Wikipedia, this is as simple as looking for a file with the string as its name. For example, for the unknown string "polygraph", the system finds the file "wikipedia/P/o/l/Polygraph", which corresponds to http://en.wikipedia.org/wiki/Polygraph. In the case of a Redirect page, the system follows the redirect one level. The system was able to find pages for 45% of the unknown terms in our dataset, yielding 1,228 pages.

Once a page has been downloaded, the Sentence Extractor identifies a sentence that defines the term. In the prototype, this simply consisted of taking the first sentence in the article. In the "polygraph" example above, the sentence extracted was:

> A polygraph (commonly yet incorrectly referred to as a lie detector) is a device that measures and records several physiological variables such as blood pressure, heart rate, respiration and skin conductivity while the subject is asked a series of questions.[2]

While the performance of the sentence extractor might be improved using machine learning techniques, our evaluation found that this simple approach – taking the first sentence in the article body – works surprisingly well, finding a relevant, complete sentence in 90.3% of 175 hand-labeled pages.

Before the system tries to actually understand the sentence, it first creates a stub term for the concept and adds lexical information to that stub so the term can be recognized by semantic interpreters. In the case of "polygraph", a new Cyc constant called `#$Polygraph` is created, which the string is then lexically mapped to using a `#$nameString` assertion (e.g., `#$nameString #$Polygraph "polygraph"`). The unknown string and extracted sentence are then given to Cyc's EBMT-L™ (Example-based Machine Translation to Logic) semantic interpreter, which attempts to extract logical sentences from the natural language sentence. This is done by comparing the syntax of the sentence with "templates" built from English/

[2] Permanent link - http://en.wikipedia.org/w/index.php?title = Polygraph&oldid = 104718615.

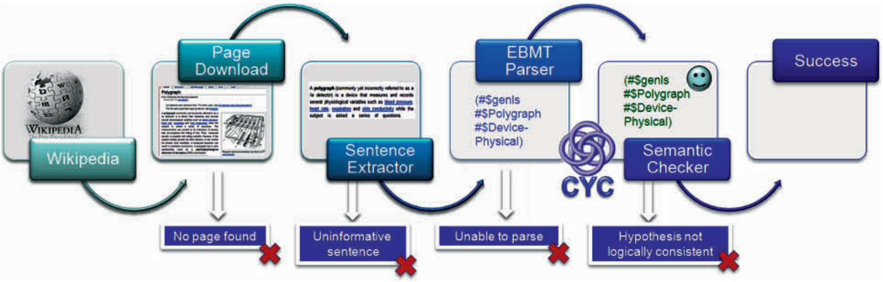


Fig. 11.1 Schematic description of the process for automatic ontology extension for unknown phrases. First, an attempt is made to find an appropriate page (with a title matching the phrase), in Wikipedia. If such a pages is found, and the first sentence contains the target term, it is processed by the Example-Based MT to Logic (EBMTL) component of Cyc, which uses semantically parameterized abstracted linkage patterns (from a slightly modified version of the CMU link parser (Sleator and Temperley 1991)) to produce CycL translations of NL sentences. If this succeeds in finding a match, and the CycL is not inconsistent with the KB (for example, by creating a supernumerary instance of a collection with known cardinality), it is added to the KB, providing at least minimal taxonomic and lexical information for the term. Table 11.5, below, details the performance of the stages of this automated ontological process

Table 11.4 Recognition numbers for the system broken down by position

Total distinct strings in corpus	Subject		Verb		Complement		Total	
	4,410	100.0%	2,013	100.0%	5,381	100.0%	1,1804	100.0%
Unknown to system	788	17.9%	109	5.4%	687	12.8%	1,584	13.4%
Known only to WordNet or NER	1,286	29.2%	563	27.0%	1,398	25.0%	3247	27.5%
WordNet only	546	12.4%	563	27.0%	1,012	18.8%	2,121	17.0%
NER only	739	16.8%	0	0%	386	7.2%	1,125	9.5%

CycL pairs given to the system as training examples. Since these templates include links identified by the CMU link parser (Grinberg et al. 1995) as features, they can generalize widely, allowing, for example, adverbial constructions and embedded clauses to be ignored.

Very few training examples are needed to enable EBMT-L to understand a large number of the Wikipedia sentences. Among the training examples used for this work were:

“A cat is a mammal.” → (`#$genls #$Cat #$Mammal`)

“Russia is a country.” → (`#$isa #$Russia #$IndependentCountry`)

With only twenty training examples, the system was able to generate CycL for 53% of the extracted sentences. With more templates, this number could almost certainly have been improved. Cases where no template matched the syntactic pattern of the sentence accounted for roughly half of the sentences that could not be interpreted. The other most common failure mode involved sentences that contained a variant of the focal string, rather than the original string. For instance, pages about people often use the person's full name (including middle name), while the initial page name strings, used to create the stub, does not. This problem could be overcome in several different ways, ranging from adding more lexical information to the KB (using e.g. a Viterbi alignment to find the portion of the sentence that most closely matches the search phrase, and adding lexical assertions for the "close" phrase), to using a named entity recognizer and then determining after the fact whether the named-entity has a name that is close to that of the search phrase, to modifications to the EBMT-L system itself to handle these sorts of variations more gracefully.

The result of this process is a set of CycL sentences about the terms represented by the input unknown strings. The sentence above describing polygraphs, for example, contains the same linkages and focal string and therefore matches the template for "A cat is a mammal." Allowing the system to generate the logical formula (`#$genls #$Polygraph #$PhysicalDevice`) as a partial interpretation of the sentence. In cases of natural language ambiguity, several alternative hypotheses are generated. For example, "Magma is molten rock…" could be interpreted as (`#$genls #$Magma #$StoneStuff`) or, less satisfactorily, (`#$genls #$Magma #$RockMusic`).

The CycL assertions generated are then checked for logical consistency using shallow inference in Cyc (somewhat confusingly called "semantic well-formed formula [WFF] checking"). As a result, a small number (about 16%) of the sentences are rejected as semantically incoherent. Of these, no factually correct interpretations were unjustly rejected. Semantically ill-formed hypotheses are generally the result of natural language ambiguity. For example, the sentence "A total is a sum" is interpreted by the EBMT-L system as possibly meaning (`#$genls #$Total #$Sum-TheProgram`). In other words, the word "sum" could be referring to the computer program with that name. However, the syntactic analysis of the system is consistent with an EBMT-L example using the #$genls (sub-type) relationship. Since `#$Sum-TheProgram` is not a collection, it is not semantically compatible with `#$genls`, so this hypothesis is rejected.

In the end, the system was able to suggest correct ontological placements for 20% of the extracted sentences, or roughly 9% of the unknown strings; Table 11.5 details the performance of the stages of this automated ontological process.

While the logical representations extracted from Wikipedia contain only a small subset of the information in the Wikipedia articles, these results are still quite valuable. To help find equivalence classes, it is sufficient to know that a polygraph is a `#$PhysicalDevice`. While having the purpose of the device in the KB might be useful, it is not necessary. Similarly, while it would be more precise to say that a polygraph is an electrically powered device, and that it is a measurement device, it is still the case that just knowing that it is a physical device is enough to determine many of the sorts of things that it could, or could not, enter into equivalence classes with.

Table 11.5 During automated concept acquisition, a multistage process selects and analyzes potentially relevant content from the English Wikipedia. For the initial implementation, this table shows, as percentages, how many candidate concepts survive each processing stage, both at each step and cumulatively. The reported figures are based on a hand-labeled sample

From prev. stage	All strings	Stage
45.0	45.0	Retrieved a page from Wikipedia
90.3	40.6	Extracted a relevant, complete, definitional sentence
77.8	31.6	Extracted sentence contained string being investigated
97.6	30.9	CMU link parser parsed the sentence
70.0	21.6	Link parse matched against an EBMT template
91.7	19.8	All other strings needed to fill the template are understood
84.4	16.7	At least one CycL sentence is well-formed (WFF).
53.8	9.0	At least one sentence is correct

11.6 Conclusion and Future Research

The work reported here demonstrates the utility of using deep knowledge from Cyc for effectively identifying redundant information in texts by using both semantic and syntactic information. In this case, the redundancy identified was within the set of extracted S-V-O triples for a particular document. The chosen equivalence class weighting scheme was effective, with tighter classes (those with higher weights) generally judged to be of higher quality. We suspect that the number of high-precision classes could be substantially improved by learning the parameters used in the weight calculation, both at the level of weighting semantic information, and at the level of weighting its combination.

It is important to reiterate that the equivalence class results reported here are only for intra-document equivalences. Although time constraints prevented us from extending the implementation, we believe that performing the same analysis for interdocument equivalence classes would result in many more equivalence classes, with more members, due to the inherently higher likelihood of interdocument, versus intradocument redundancy.

The work carried out to expand the ontological coverage using Wikipedia, to support higher productivity and accuracy of equivalence class identification, is also very promising. While the number of unknown concepts that made it all the way through the pipeline is relatively small, the information available at the end of the process was of sufficiently high quality that it could be very quickly reviewed by human ontologists, or perhaps volunteers[3], and used in the process of finding equivalence classes. There are several obvious places in the pipeline where improvements could be made fairly quickly. For example, adding just a few more training example for the EBMT-L system is likely to substantially increase the number of sentences that are understood. Taking advantage of the strong context

[3] Cycorp maintains an internet game, http://game.cyc.com, to support exactly this kind of volunteer review and contribution.

(first sentence in a Wikipedia article) to loosen the syntactic match constraints (e.g. by no longer requiring a link verb) is also likely to increase productivity.

Similarly, changes to the way the system finds pages, and text, about the unknown string are likely to bring substantial dividends. For example, the system could be more intelligent in its use of redirects, and make use of more Wikipedia meta-information (e.g. "This Wikipedia article about a person is a stub." Similarly, recursive investigation of other unknown terms in the definitional sentence should also help. Of course, we do not expect to ever achieve 100% recall for unknown phrases, due both to incomplete coverage in Wikipedia and errors in the triple-extraction process, but it seems possible that the system will come close, achieving very high coverage for NLP tasks. Future work should attempt to estimate the upper bounds of recall for unknown strings from Wikipedia, and extend the scope to attempt to recover relevant, nontaxonomic relationships.

Acknowledgments This work was supported by the Slovenian Research Agency and the IST Programme of the European Community under SEKT (IST-IP-2003-506826), NeOn Lifecycle Support for Networked Ontologies (IST-4-027595-IP), TAO (IST-2004-026460) and PASCAL Network of Excellence (IST-2002-506778).

References

Fellbaum C (Ed.) (1998) WordNet: An electronic lexical database. Cambridge, MA: MIT. ISBN-13 978-0-262-06197-1.

Fortuna B, Mladenić D, Grobelnik M (2006) Semi-automatic construction of topic ontologies. In: Ackermann, Berendt, Grobelnik, Mladenić (Eds.). Semantics, Web and Mining: Joint International Workshops, EWMF 2005 and KDO 2005, Porto, Portugal, October 3 and 7, 2005: revised selected papers, Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence), vol. 4289. Springer, Berlin Heidelberg, 121–131.

Fortuna B, Mladenić D, Grobelnik M (2005) Visualization of Text Document Corpus. Informatica Journal, vol. 29, no. 4: 497–502.

Grinberg D, Lafferty J, Sleator D (1995) A robust parsing algorithm for link grammars. Carnegie Mellon University Computer Science Technical Report CMU-CS-95-125, and Proceedings of the Fourth International Workshop on Parsing Technologies, Prague, September.

Grobelnik M, Mladenić D (2005) Automated Knowledge Discovery in Advanced Knowledge Management. Journal of Knowledge Management, Vol. 9, 132–146.

Krishnan V, Manning C D (2006) An Effective Two-Stage Model for Exploiting Non-Local Dependencies in Named Entity Recognition. Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, 1121–1128.

Lenat D B (1995, November) Cyc: A Large-Scale Investment in Knowledge Infrastructure. Communications of the ACM Vol. 38, no. 11.

Leskovec J, Grobelnik M, Milic-Frayling N (2004) Learning Sub-structures of Document Semantic Graphs for Document Summarization. In Workshop on Link Analysis and Group Detection (LinkKDD2004). The Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Leskovec J, Milic-Frayling N, Grobelnik M (2005) Impact of Linguistic Analysis on the Semantic Graph Coverage and Learning of Document Extracts. National Conference on Artificial Intelligence AAAI-2005, Pittsburgh, PA.

Matuszek C, Cabral J, Witbrock M, DeOliveira J (2006) An Introduction to the Syntax and Content of Cyc. In Proceedings of the 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering, Stanford, CA, March 2006.

Panton K, Miraglia P, Salay N et al. (2002) Knowledge Formation and Dialogue Using the KRAKEN Toolset. In Proceedings of the Fourteenth National Conference on Innovative Applications of Artificial Intelligence, 900–905. Edmonton, Canada.

Shah P, Schneider D, Matuszek C, Kahlert R C, Aldag B, Baxter D, Cabral J, Witbrock M, Curtis J (2006) Automated Population of Cyc: Extracting Information about Named-entities from the Web. In Proceedings of the Nineteenth International FLAIRS Conference, 153–158, Melbourne Beach, FL, May 2006.

Sleator D, Temperley D (1991) Parsing English with a Link Grammar. Carnegie Mellon University Computer Science Technical Report CMU-CS-91-196, October 1991.

Chapter 12
Visualization of Temporal Semantic Spaces

Blaž Fortuna, Dunja Mladenić, and Marko Grobelnik

Abstract In order to realize the semantic web vision, the creation of semantic annotation, the linking of web pages to ontologies, and the creation, evolution and interrelation of ontologies must become automatic or semi-automatic processes. Natural Language Generation (NLG) takes structured data in a knowledge base as input and produces natural language text, tailored to the presentational context and the target reader. NLG techniques use and build models of the context and the user and use them to select appropriate presentation strategies.

In the context of Semantic Web or knowledge management, NLG can be applied to provide automated documentation of ontologies and knowledge bases. Unlike human-written texts, an automatic approach will constantly keep the text up-to-date which is vitally important in the Semantic Web context where knowledge is dynamic and is updated frequently. This chapter presents several Natural Language Generation (NLG) techniques that produce textual summaries from Semantic Web ontologies. The main contribution is in showing how existing NLG tools can be adapted to take Semantic Web ontologies as their input, in a way which minimizes the customization effort.

A major factor in the quality of the generated text is the content of the ontology itself. For instance, the use of string datatype properties with implicit semantics leads to the generation of text with missing semantic information. Three approaches to overcome this problem are presented and users can choose the one that suits their application best.

12.1 Introduction

Visualization is commonly used in data analysis to help the user in getting an initial idea about the raw data as well as a visual representation of the regularities obtained in the analysis. In the case of textual data, visualization can help understanding, discovery and summarization of the topics which are discussed in the text.

B. Fortuna(✉)
Jožef Stefan Institute, Slovenia
e-mail: Blaz.Fortuna@ijs.si

J. Davies et al. (eds.), *Semantic Knowledge Management*,

If the time dimension is present in the data, this can be used to further show how specific topics emerge or disappear through time.

Visualization can further help understand the context and relationships of objects (named entities), which are described in a text corpus. For example, given a stream of news articles, one can apply visualization on the named-entities that appear in the corpus to show their topical context (e.g., the topical context of Bill Clinton on a visit in China could be Asia, Trade, etc.) and topical relations between different named-entities (e.g., the topical relation between Chicago and Bill Clinton for a given time period could be Democratic convention that took place there and which Bill Clinton attended).

In this chapter we first present a set of methods for visualization of a document corpus based on calculation of the semantic space of the corpora documents. Then we propose a possible extension of the methods to enable visualization of the named-entities occurring in the text corpora in their temporal semantic space.

12.2 Dimensionality Reduction of Text

The first step of our approach to visualization of a document corpus is to map all the documents into a two-dimensional vector space so we can plot them on a computer screen. Ideally they would be positioned in such a way that the distance between two documents would correspond to the content similarity between them. In the next subsections we give a sequence of methods that transform a text document into a two-dimensional point.

12.2.1 Document Representation

The most often used representation of text documents, which can be successfully used for topical modelling, is the bag-of-words (Salton 1991) representation. In this representation the order of the words in a document is ignored and only the presence or count of the words in the document is considered. More technically, documents are represented as vectors in a high-dimensional vector space, where each dimension corresponds to one word from the vocabulary. The value of a specific dimension in the document vector reflects the count of appearances of the word in the corresponding document.

The similarity of two documents that are represented as vectors can be calculated using the cosine similarity measure, which is the cosine of the angle between the two vectors. Effectively this means that the more words the two documents have in common, the more similar they are to each other. Cosine similarity is a standard measure between documents used in the field of information retrieval.

However, not all the words are of equal importance for determining the similarity between two documents, for example, two documents sharing word "the" are not necessarily topically similar. By weighting the elements of the vectors we can bias the cosine similarity so it disregards topically non-related words (e.g., "the", "and", etc.) and puts more account onto the topically significant words. In this chapter we used the standard TFIDF weights where each element of the vectors (the count of the word or TF – term frequency) is multiplied by the importance of the word with regards to the whole corpus (the IDF weight – inverted document frequency). The IDF weight for the *i*-th word is defined as $IDF_i = \log(N/df_i)$, where *N* is total number of documents and df_i is the document frequency of the *i*-th word (the number of documents from the whole corpus in which the *i*-th word appears). The IDF weight was found to put lower weight on common words (the ones that appear in almost all the documents) and favour the words that appear scarcely in the documents.

12.2.2 Latent Semantic Indexing

A well known and used approach for extracting latent semantics (or topics) from text documents is Latent Semantic Indexing (Deerwester et al. 1990). In this approach we first construct a term-document matrix *A* from a given corpus of text documents. This is a matrix with vectors of documents from a given corpus as columns. The term-document matrix *A* is then decomposed using singular value decomposition, so that $A = USV^T$; here matrices *U* and *V* are orthogonal and *S* is a diagonal matrix with ordered singular values on the diagonal. Columns of matrix *U* form an orthogonal basis of a subspace in the bag-of-words space where vectors with higher singular values carry more information – this follows from the basic theorem about SVD, which tells that by setting all but the largest *k* singular values to *0* we get the best approximation for matrix *A* with matrix of rank *k*).

Vectors that form the basis can be also viewed as concepts and the space spanned by these vectors is called the *Semantic Space*. Each concept is a vector in the bag-of-words space, so the elements of this vector are weights assigned to the words coming from our documents. The words with the highest positive or negative values form a set of words that are found most suitable to describe the corresponding concept.

12.2.3 Multidimensional Scaling

Multidimensional scaling (Carroll and Arabie 1980) enables dimensionality reduction by mapping original multidimensional vectors onto two dimensions. Here the points representing documents are positioned into two dimensions so they minimize some energy function. The basic and most common form of this function is

$$E = \sum_{i \neq j} \delta_{ij} - d(x_i, x_j)^2 \tag{12.1}$$

where x_i are two-dimensional points, $d(x_i, x_j)$ denotes euclidian distance between the two points (we will refer to this with d_{ij} in short) and δ_{ij} represents the similarity between two vectors (in our case documents *i* and *j*). An intuitive description of this optimization problem is: the better the distances between points on the plane approximate real similarity between documents, the lower the value of the energy function *E*. Notice that function *E* is nonnegative and equals zero only when distances between points match exactly with similarity between documents.

A common way of applying multidimensional scaling is by using gradient descent for the optimization step. The problem with this approach is that the energy function is not convex: it usually has many local minima which are not that interesting for us. One could start this method repeated times with different initial states and then choose the results with the lowest energy.

We choose a slightly different approach which is based on reformulation of the energy function. Given a placement of points, we calculate for each point how to move it to minimize the energy function. We denote the current positions of points with (x_i,y_i) and the desired position with $(x_i',y_i') = (x_i + \delta x_i, y_i + \delta y_i)$. Then we have

$$\begin{gathered} d_{ij}^{2'} - d_{ij}^{2} = \\ (x_i - x_j)^2 + (y_i - y_j)^2 - (x_i + \delta x_i - x_j - \delta x_j)^2 + (y_i + \delta y_i - y_j - \delta y_j)^2 \approx \\ (x_i - x_j)\,\delta x_i + (x_j - x_i)\,\delta x_j + (y_i - y_j)\,\delta y_i + (y_j - y_i)\,\delta y_j = \\ [(x_i - x_j), (x_j - x_i), (y_i - y_j), (y_j - y_i)][\delta x_i, \delta x_j, \delta y_i, \delta y_j]^T . \end{gathered}$$

By writing this for each pair *(i,j)* and substituting d_{ij}' with the original distance δ_{ij} between *i*-th and *j*-th document we get a system of linear equations which has a vector of moves (δx and δy) for a solution. This is an iteration which finds a step towards minimizing energy function and is more successful at avoiding local minima. Each iteration involves solving a linear system of equations with a very sparse matrix. This can be done very efficiently using Conjugate Gradient (CG) method. Finally, the points are normalized to lie in the square $K = [0,1]^2$.

12.2.4 Visualization Using Dimensionality Reduction

The proposed visualization approach is based on a sequential combination of linear subspace methods and multidimensional scaling for reducing document space dimensionality. Both methods can be independently applied to any data set that is represented as a set of vectors in some higher dimensional space. Our goal is to lower the number of dimensions to two so that the whole corpus of documents can be shown on a computer screen.

Linear subspace methods like Latent Semantic Indexing focus on finding projections from the original bag-of-words space into a lower dimensional space while trying to preserve as much information as possible. By projecting data (text documents) only on the first two directions we can get the points that live in the two-dimensional space.

The problem with linear subspace methods is that only the information from the first two directions is preserved. In case of LSI it would mean that all documents are described using only the two main concepts.

In Fortuna et al. (2006) we have proposed combining the two methods (linear subspace and multidimensional scaling) in order to take advantage of the both. What follows is description of that algorithm:

Input: Corpus of documents to visualize in form of TFIDF vectors.
Output: Set of two-dimensional points representing documents.
Procedure:

1. Calculate *k*-dimensional semantic space generated by input corpus of documents,
2. Project documents into the semantic space,
3. Apply multidimensional scaling using energy function on documents with Euclidian distance in semantic space as similarity measure.

There is a parameter *k* in the first step which has to be given to this procedure. In order to select this parameter we use following heuristic. Let $\sum_k = S_1^2 + S_2^2 + \ldots + S_k^2$, where S_i is *i*th singular value. We know that $\Sigma_n = \mathrm{Trace}(A^T A)$, where *n* is the number of the documents in the corpus and *A* is the term-document matrix. From this we can guess *k* by prescribing the ratio Σ_k/Σ_n to some fixed value (e.g., 75% which was found to produce the best results).

12.2.5 Semantic Landscape

The procedure defined in the previous section can be used to make a map of topics from a given collection of documents. The map is calculated for each document collection separately with no connection to the other document collections. But in some cases one would like to visualize a given set of documents on a predefined set of topics, for example, for comparing two document collections on a predefined set of topics. To handle such scenarios we developed a notation of *semantic landscape* which defines the set of topics over which the documents will be visualized. Essentially it is a pre-given set of documents, which talk about the desired topics (e.g., news events), projected onto a two-dimensional map. The documents which define the semantic landscape are called *landmarks*.

Notice that the landmarks do not appear in the visualization, they are just used for positioning the given collection of documents on the map as follows. In order to position a document on a map, the top *N* most similar landmarks are calculated and the position of the document is then a convex combination of these landmarks' positions. More specifically, let μ_i be the cosine similarity of the document to the *i*th most similar landmark and A_i be the location of the landmark on the two-dimensional map. Than the location of the document is calculated as

$$\left(\frac{1}{\sum_{i=1}^{N}\mu_i}\right)\sum_{i=1}^{N}\mu_i A_i. \tag{12.2}$$

The positioning is computationally very inexpensive which in turn means that once the semantic landscape is calculated the visualization of new documents can be done in near real time.

Semantic landscape can be also use for more efficient visualization of larger corpora, when the multidimensional scaling (MDS) algorithm would take too long to finish. In such cases a semantic landscape is generated from landmarks, which are extracted from the corpora. In our experiments we used a *k*-means (Jain et al. 1999) clustering algorithm for clustering the corpora in a larger number of clusters, usually several hundreds. Since *k*-means is more efficient than MDS, this significantly increases the time performance.

12.2.6 Time Dimension

When visualizing named-entities from news articles, their topical context can heavily depend on the time period. For example, the context of Bill Clinton would contain different words during an election campaign than during his visit to China. Such changes of context can be visualized by introduction of the time dimension to the presented visualization framework. We do this by introducing frames, where one frame is a visualization of the named-entities using their context from a particular time period. Arranging such frames in a sequence sorted by time shows the changes of contexts through which the named entities occur.

Generation of a frame sequence works as following. First a semantic landscape is constructed from a collection of documents where the documents can be either whole articles, for example, news articles or the contexts of named-entities. The generated semantic landscape is used as the basis for each of the frames. Then each frame is assigned a time period, and the named-entities are projected (as described in the Sect. 12.2.5) on the frame's landscape using their context from the time period.

Using the same landscape across all the frames enforces consistency across all the frames since the semantic landscape defines which area of the map corresponds to which topic. If each frame were generated as a separate visualization, then the topic area would be located independently of the topic's location on the previous frame, resulting in confusion when sliding through the frame sequence.

12.3 Visualization

After the documents from the corpus are mapped onto a two-dimensional plane, some other techniques can be used to make the structure and the time dynamics of the documents more explicit for the users:

Landscape generation: Landscape can be generated by using the density of points. Each point from square K is assigned height using the formula

$$h(x,y) = \sum_i \exp(-\sigma \,||\, (x,y) - (x_i, y_i) \,||^2), \tag{12.3}$$

where σ defines how wide is the influence of one point. In the case of semantic landscape, the landscape can be calculated by either using the density of points or the density of landmarks.

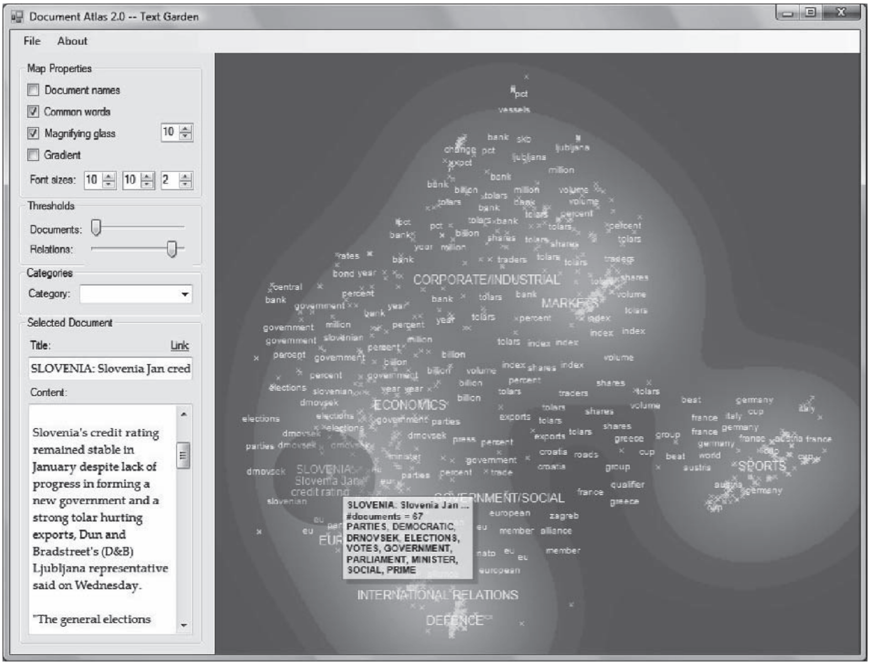
Keywords: Each point from square K can be assigned a set of keywords by averaging TFIDF vectors of documents which appear within a circle with centre in this point and radius R.
Gradient: For each point the direction in which it will move in the next frame of the visualization is calculated and can be displayed as an arrow next to the point to indicate the temporal dynamics of the point.
Trace: The whole trace of the point as it moves between frames can be shown.
Relations: Links between documents are shown as arrows on the visualization. In the case of named-entities the relations are topical and are extracted from the news articles based on the co-occurrences. The context of a relation can be showing as a list of keywords, extracted from the texts in which both named-entities co-occur.
We use these features when showing visualizations to make them more descriptive and to improve the overall user experience.

In our system, named Document Atlas (Fortuna et al. 2006; http://docatlas.ijs.si), that is implemented on the top of the Text Garden library (Grobelnik and Mladenić 2008; http://www.textmining.net), the documents are presented as yellow crosses on a map and the density is shown as a texture in the background of the map (the lighter the colour, the higher the density). The user moves through time with the slider located on the left side of the screen. The most common keywords are shown for the areas around the map. Positions, for which the keywords are computed, are selected randomly. Keywords are displayed using white colour font.

When the user moves a mouse on the map a set of the most common keywords is computed in real-time for the area around the mouse pointer. The area from which these keywords are extracted, is marked darker on the map, and the list of keywords is shown in the semi-transparent window next to the mouse as shown in Figs. 12.1 and 12.2. The user can also zoom-in to see specific areas in more details. By clicking on a document (yellow crosses on the map), more information about the selected document is shown on the left side of the screen (see Fig. 12.1).

12.4 Experiments

We applied the proposed visualization techniques to the task of visualizing named-entities extracted and contextualized from the Reuters RCV1 dataset (Lewis et al. 2004).



Document Atlas 2.0 -- Text Garden

Map Properties

☐ Document names

☑ Common words

☑ Magnifying glass 10

☐ Gradient

Font sizes: 10 10 2

Thresholds

Documents:

Relations:

Categories

Category:

Selected Document

Title: Link

SLOVENIA: Slovenia Jan cred

Content:

Slovenia's credit rating remained stable in January despite lack of progress in forming a new government and a strong tolar hurting exports, Dun and Bradstreet's (D&B) Ljubljana representative said on Wednesday.

"The general elections

SLOVENIA: Slovenia Jan ...
#documents = 67
PARTIES, DEMOCRATIC, DRNOVSEK, ELECTIONS, VOTES, GOVERNMENT, PARLIAMENT, MINISTER, SOCIAL, PRIME

Fig 12.1 The main window of the Document Atlas tool

12.4.1 News Article Visualization

In the first experiment we applied the proposed visualization to two different subsets of the Reuters news articles. The first subset consisted of the news articles, which included the word "*Slovenia*" and the second subset consisted of all the articles with the words "*Bill*" and "*Clinton*". For each subset we produced two different visualizations, the first showed all the articles from the subset and the second produced a sequence of visualizations, one for each month. Both types of visualization were produced using the same semantic landscape, which was generated using the landmarks extracted from the full subset. Results of the visualization are presented in the Figs. 12.2–12.5. Figures 12.2 and 12.4 include the visualization of the whole subset and the Figs. 12.3 and 12.5 include the visualization of one particular month.

12.4.2 Named-Entity Extraction and Visualization

In the second experiment we focused on visualization of the contexts in which named-entities appear in the news. The most frequent named-entities were extracted

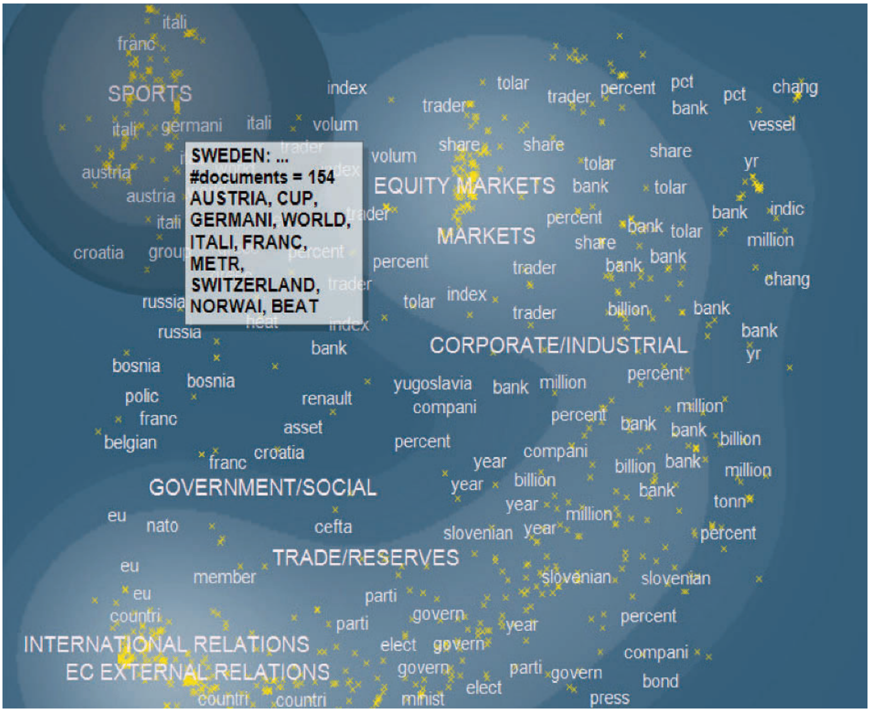


Fig 12.2 A visualization of the articles containing keyword "*Slovenia*". One can see from the visualization that Slovenia appeared in the articles related to sports (*upper left*), economy (*upper right*), international politics with focus on European Union and NATO (*bottom left*) and internal politics (*bottom right*)

from Reuters news articles and visualized based on their contexts, as extracted from the news articles.

The named-entity extraction approach (Grobelnik et al. 2006) we used is simple but very efficient at handling large amounts of data. Whenever one, two or three adjacent words in the document begin with a capital letter, this group of words is considered to be a named entity. There are two exceptions to this rule: firstly, the sequence of two or three adjacent capitalized words may be interrupted by a non-capitalized word if the latter belongs to a predefined list of exceptions (consisting of non-capitalized words that are often a part of named entities, such as "of" " and "von"). Secondly, a capitalized word that appears at the beginning of a sentence (and is not part of a group of two or three adjacent capitalized words) is only treated as a named entity if the same word appears capitalized somewhere else where it is not the first word of a sentence. This approach has obvious drawbacks such as not being able to handle named entities with names longer than three capitalized words.

When comparing two names (occurring in the same document) to see if they refer to the same named entity, the first word of any multi-word name is ignored.

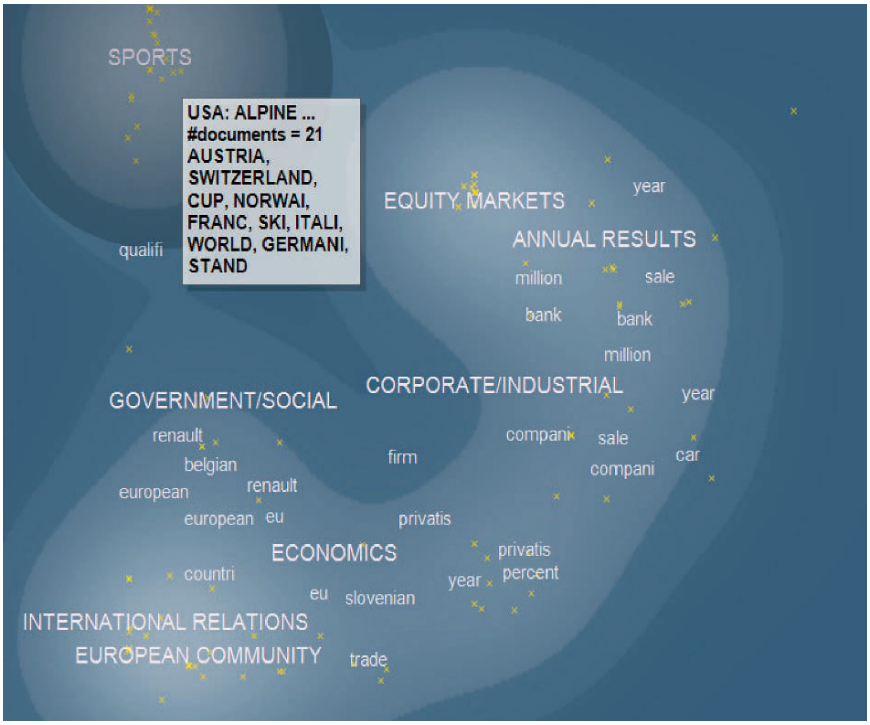
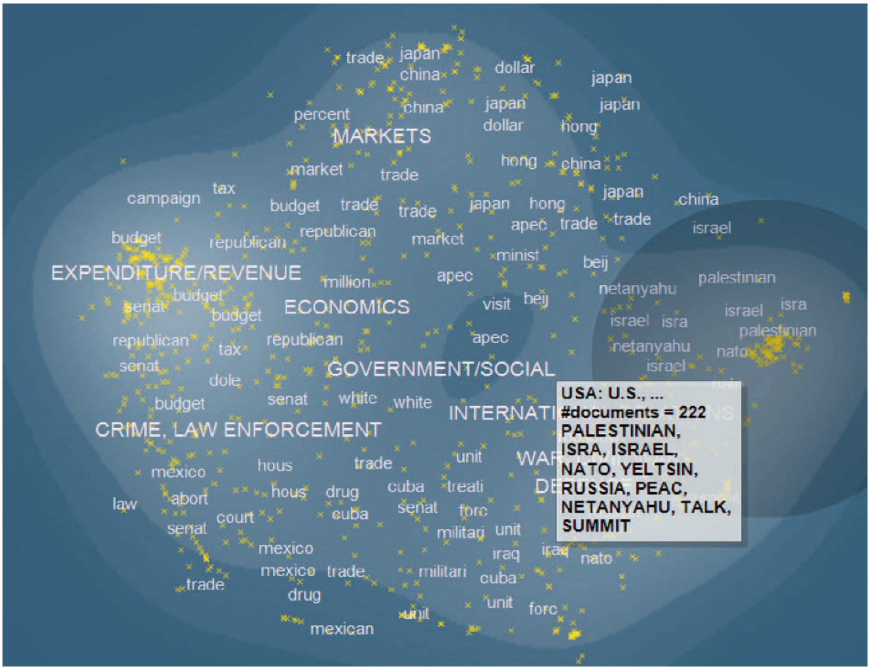


Fig 12.3 A visualization of the articles containing the keyword "*Slovenia*" focused just on the articles from March 1997, visualized on the right half. We can see, for example, that sports news was mostly about skiing and that international affairs were mostly focused on EU related topics

This allows for example "President Clinton", "Bill Clinton" and "Clinton" to be recognized as names referring to the same entity.

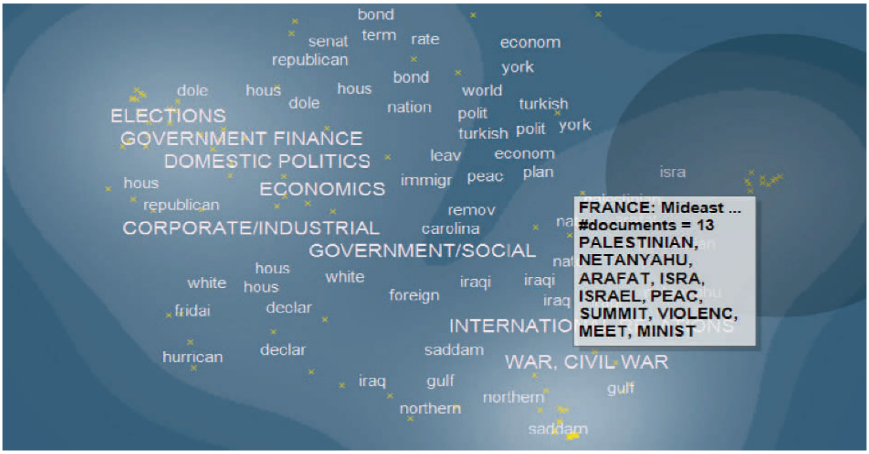
Once the named-entities are extracted, we generate the context for each entity and for all the entity pairs that co-occurred in any of the news articles. The context is simply the concatenation of texts of all the articles in which the named-entity or the pair of named-entities appears. This enables us to deal with the named-entities on the same way as text documents. Since all the news articles are annotated with time information, we can split the time period into several time windows and generate a separate context for each time window. By allowing some overlapping between the windows the transition between frames can be made smoother. In the experiments presented in this chapter we used 1 month time window with 1 week overlap on each side.

Table 12.1 shows examples of contexts for Bill Clinton and Chicago during the first 2 months of our data and the relation in which they appear. There is also the context of co-occurrences of Bill Clinton and Chicago, from which one can see how they were related in the first month due to the democratic convention which was happening in Chicago, while in the second month there is no relation between the two entities (i.e., they do not co-occur in any of the news articles).



USA: U.S., ...
#documents = 222
PALESTINIAN,
ISRA, ISRAEL,
NATO, YELTSIN,
RUSSIA, PEAC,
NETANYAHU, TALK,
SUMMIT

Fig 12.4 A visualization of the articles containing the keyword "*Bill Clinton*". From the visualization we can see that he mostly appears in the articles related to USA internal politics (*big cluster on the middle left*), global economic issues (*central and upper middle part*) and Middle East peace process (*big cluster on the middle right*). There are also many smaller topics spread around the map, such as illegal drugs (*bottom middle*), law enforcement (*bottom left*), etc.



FRANCE: Mideast ...
#documents = 13
PALESTINIAN,
NETANYAHU,
ARAFAT, ISRA,
ISRAEL, PEAC,
SUMMIT, VIOLENC,
MEET, MINIST

Fig 12.5 A visualization of the articles from September 1996 containing the keyword "Bill Clinton". We can see that the internal politics cluster (*middle left*) was mostly focused on elections, that the Middle East peace talks were happening at that time (*middle right*) and that he was also involved in some issues related to Iraq (*a small cluster at bottom right*)

Table 12.1 Context keywords for named-entities Bill Clinton, Chicago and keywords from co-occurrence of Bill Clinton and Chicago for two consecutive time periods

Named-entity	Time window	Main context keywords
Bill Clinton	From 1996-08-20 till 1996-09-20	President, republican, campaigned, election, presidential, missiles, attacked, democratic, troops, Kurdish
	From 1996-09-20 till 1996-10-20	President, republican, summit, presidential, debated, campaign, elections, peace, Palestinians, prime
Chicago	From 1996-08-20 till 1996-09-20	News desk, convention, democratic, cents, future, trader, printing, bushels, wheat, delivery
	From 1996-09-20 till 1996-10-20	News desk, cent, printed, traders, futures, wheat, soybeans, corn, prices,
Bill Clinton–Chicago	From 1996-08-20 till 1996-09-20	Conventional, democrat, training, day, campaign
	From 1996-09-20 till 1996-10-20	*No relation*

After we have extracted the contexts for the named-entities we can use them as inputs to the visualization. The visualization was created by first making a semantic landscape from the contexts of all the named-entities. Then for each month a separate visualization was created by projecting the contexts of the named-entities from that month onto the semantic landscape. Figure 12.6 shows example of such a visualization, also displaying how trace and gradients are visualized. Figures 12.7 and 12.8 show how relations are displayed, using the example of the relation between Bill Clinton and Chicago.

12.5 Related Work

In Fortuna et al. (2006) we have proposed a system for text corpora visualization which is based on a similar visualization pipeline and techniques as the ones presented in this chapter. The main difference is that the methods presented here can operate on larger corpora (Sect. 12.2.5), can visualize the time dimension (Sects. 12.2.6 and 12.3) and support visualization of named-entities.

Another approach for visualization of topics in document collections, as they evolve through time, is the system Theme River (Havre et al. 2000). The idea is that a document collection is represented as a river flowing from left to right and the river is composed of multiple streams, each stream representing one topic and depicted with different colours. The x-axis corresponds to time and the width of each stream at particular time represents the strength of the corresponding topic at that time. The main difference between this and our approach is that our approach

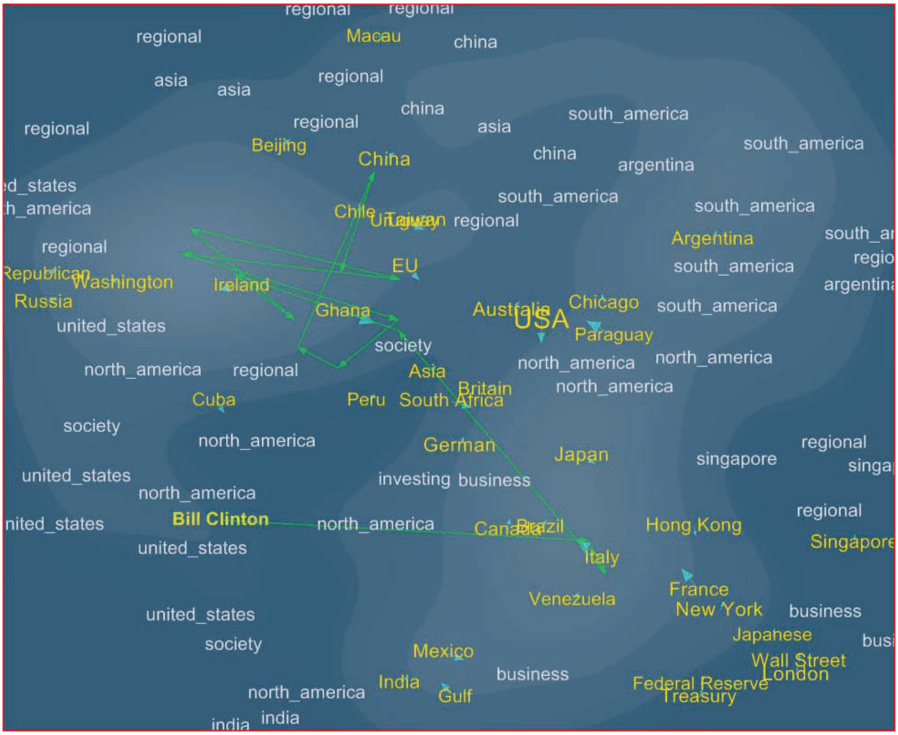


Fig 12.6 The map displays the visualization of the most frequent named-entities from the Reuters corpus for the first month. Next to each named-entity there is a blue arrow, indicating in which direction it will move in the next frame. For the selected named-entity (Bill Clinton) a trace of his travel across all the frames is displayed using a sequence of long green arrows. He was president of USA at that time which put him in many different contexts over different months and this is reflected in the visualization with an extensive trace around the map. Named-entities like China have a much more stable location on the map, since they do not appear in so many contexts. For example China is located on the top middle (location of Asia related news) location throughout the whole year

shows a much more detailed visualization for each time period and the relation of each document or named-entity to the topics; whilst the Theme River shows one global overview of the main topics and evolution of their strength over time, but does now show how the individual documents or smaller topics are positioned in relation to the major trends.

Somewhat related to our work is an approach to automatic topic identification in a document collection described in Wang and McCallum (2006). It uses Latent Dirichlet Allocation (LDA) for modelling the topics. The result of the method is a set of topics, where each topic is depicted by a set of keywords and their importance through time. The output is a visualization similar to Theme River visualization. The authors have illustrated their approach on papers from NIPS conference datasets, showing the evolution of the conference topics over the first 17 years of the conference.

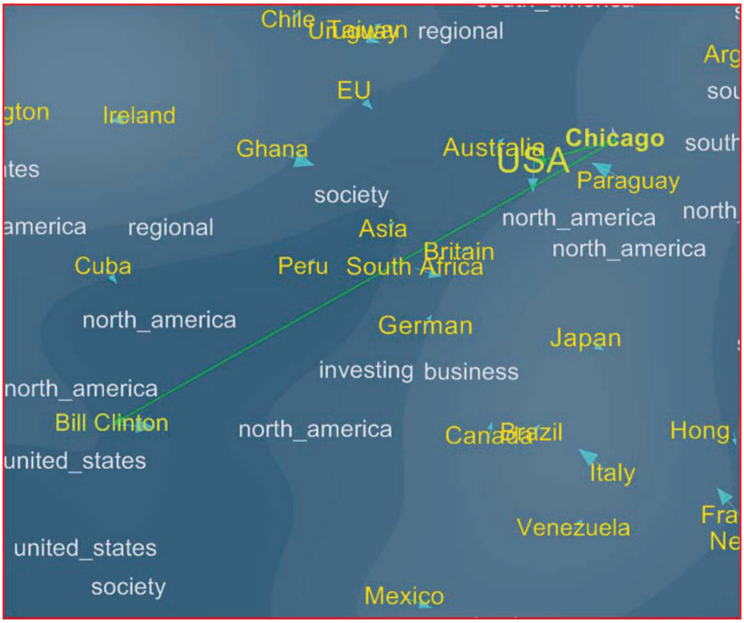


Fig 12.7 A visualization of the first month when there was a strong link between Bill Clinton and Chicago. The relation is visualized using a long green arrow and when the user moves with mouse over the arrow, a list of keywords extracted from the articles, and relating the two entities, is displayed

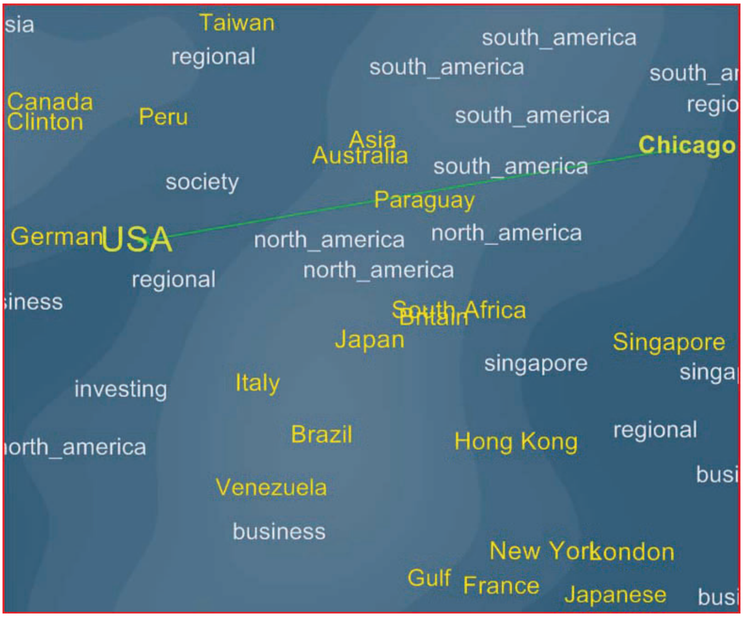


Fig 12.8 Visualization several months after the visualization shown in the Fig. 12.7. Note that in this picture Chicago is still related to the USA but no longer to Bill Clinton

12.6 Conclusions

In this chapter we proposed a serious of methods for visualization of a large collection of text documents. We show how we have combined them in a tool called Document Atlas. We applied the techniques to a specific type of data, namely news articles, and showed that it can extract and visualize the topical information that the articles contain. We propose a setting where the same methods can also be used to visualize the contexts in which named-entities appear in news, and show how these named-entities and their contexts evolve over time.

Acknowledgements This work was supported by the Slovenian Research Agency and the IST Programme of the European Community under SEKT (IST-IP-2003-506826), NeOn Lifecycle Support for Networked Ontologies (IST-4-027595-IP), TAO (IST-2004-026460) and PASCAL Network of Excellence (IST-2002-506778).

References

Carroll JD, Arabie P (1980) Multidimensional scaling. In: M.R. Rosenzweig and L.W. Porter (Eds.), Annual Review of Psychology, 31: 607–649.

Deerwester S, Dumais S, Furnas G, Landuer T, Harshman R (1990) Indexing by latent semantic analysis. Journal of the American Society of Information Science 41(6): 391–407.

Fortuna B, Mladenić D, Grobelnik M (2006) Visualization of text document corpus. Informatica 29: 497–502.

Grobelnik M, Brank J, Mladenić D, Novak B, Fortuna B (2006) Using DMoz for constructing ontology from data stream. International Conference on Information Technology Interfaces, Croatia.

Grobelnik M, Mladenić D (2008) Text Garden, Springer, Berlin Heidelberg.

Havre S, Hetzler B, Nowell L (2000) ThemeRiver: Visualizing theme changes over time. IEEE Symposium on Information Visualization, 2000, pages 115–123, Salt Lake City http://docatlas.ijs.si; http://www.textmining.net.

Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review, ACM Computing Survey.

Lewis DD, Yang Y, Rose T, Li F (2004) RCV1: A new benchmark collection for text categorization research. Journal of Machine Learning Research 5: 361–397.

Salton G (1991) Developments in automatic text retrieval. Science 253: 974–979.

Wang X, McCallum A (2006) Topics over time: A non-Markov continuous-time model of topical trends. Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 424–433, Philadelphia.

Chapter 13
Semantic MediaWiki

Denny Vrandečić and Markus Krötzsch

Abstract Wikis are web-based systems for the simple and collaborative management of online content, the best-known example being the free encyclopaedia *Wikipedia*. However, the open nature of a wiki, along with potentially changing and heterogeneous authorship, and sometimes chaotic or non-existent editorial control can also become a challenge for readers who want to access information. Classical full-text search helps to find possibly relevant pages, but cannot overcome the lack of structure that hinders powerful navigation or query answering features in classical wikis. Semantic MediaWiki described in this chapter is an extension to the well known wiki implementation MediaWiki that adresses this problem. As it is based on MediaWiki, it benefits from this stable, powerful, and yet scalable platform. The extension allows for the powerful yet simple annotation and reuse of the content inside a wiki, thus making it a *semantic wiki*. Semantic MediaWiki adds database-like structuring and querying capabilities on top of an existing wiki, without requiring users to develop or adhere to a rigid database schema when authoring content.

13.1 Introduction

Wikis are web-based systems for the simple and collaborative management of online content, the best-known example being the free encyclopaedia *Wikipedia*. There and in numerous other applications of wikis, large user communities – sometimes anonymous and often loosely organised – are concurrently modifying and extending a single wiki. To fit such scenarios, wikis have been tailored towards simplicity of use and easy collaboration in large groups. Due to their easy deployment and low learning curve, wikis spread not only in the open communities of the web, but also enhanced knowledge management and information sharing within groups of co-workers, within companies or research institutes.

D. Vrandečić (✉)
Institut AIFB, Universität Karlsruhe (TH), D-76128
Karlsruhe, Germany
e-mail: vrandecic@aifb.uni-karlsruhe.de

J. Davies et al. (eds.), *Semantic Knowledge Management*,

But the wiki's open nature, potentially changing and heterogeneous authorship, and sometimes chaotic or non-existent editorial control can also become a challenge for readers who want to access information. Classical full-text search helps to find possibly relevant pages, but cannot overcome the lack of structure that hinders powerful navigation or query answering features in classical wikis.

Semantic MediaWiki is an extension to the well known wiki implementation MediaWiki that addresses this problem. As it is based on MediaWiki, it benefits from this stable, powerful, and yet scalable platform. The extension allows for the powerful yet simple annotation and reuse of the content inside a wiki, thus making it a *semantic wiki*. Semantic MediaWiki adds database-like structuring and querying capabilities on top of an existing wiki, without requiring users to develop or adhere to a rigid database schema when authoring content.

The required features grew out of the case studies within the SEKT project, resulting in a wiki that enables users to express data in Semantic Web standards without actually having to study the formal syntax and semantics of Semantic Web knowledge representation languages.

This chapter starts by recalling the general advantages of wikis, giving a brief outline of their use within the SEKT project. We then describe the basic usage of Semantic MediaWiki, which has been developed to address various shortcomings of classical wiki systems. We continue by detailing the relationships between Semantic MediaWiki and current Semantic Web standards, and finally focus on the practical reuse of semantic knowledge within the system, especially in the form of advanced querying.

13.2 Wikis

Wikis were introduced by Ward Cunningham to allow a community of practice from heterogeneous institutions, backgrounds, and geographical locations to collaborate on a common resource. Wikis are systems that not only display content on the web, but also offer a simple editing mode for this content. This allows for a quick and easy correction of errors and omissions. Users add new content to the system based on their interest and expertise. This content is evaluated and edited by other users, and thus the content grows in collaboration of the whole community. The core idea is to offer a text on the web that everyone can edit (Cunningham and Leuf 2001).

"Everyone can edit" has two different aspects in this case: first, users need to have the rights to edit the content of the wiki. Wikis on the web often even allow for anonymous editing (although both editing and viewing rights may be restricted, which is often necessary in corporate settings). But the second aspect is actually more important: "everyone can edit" implies also that the system is so simple to use that everyone has the ability, not just the right, to edit the content.

Wikis therefore introduce a syntax, *wikitext*, that is far easier to read and edit than full fledged HTML. The actual syntax is different from implementation to implementation, but usually consists of simple text that can be partially marked up for special elements like headings, lists, tables, and, most importantly, links. A wiki typically consists of many interlinked pages (or articles).

Besides this link structure, the index of all pages, and a full-text search, there are usually no further means to structure the content of a wiki. Since wikis are edited by a potentially heterogeneous group of collaborators, it usually requires quite some effort to structure the wiki by hand (so called *wiki gardening*), in order to make the content of the wiki accessible and useful.

Within the SEKT project, a wiki was used in order to capture the argumentation while building the OPJK ontology (Casanovas et al. 2007). The initial build of the ontology suffered from long and repeating discussions. The DILIGENT methodology was then applied, and it suggested capturing the argumentation (Tempich 2006). In order to capture it in a way that was easy to access, correct, extend, and refer to, a wiki was used.

The wiki proved useful and helpful. The discussions soon grew shorter, as previous decisions and their rationales were captured. Users from different geographic locations could easily get an update on the current status of the ontology by visiting the wiki. New people starting on the project could trace the decisions and design rationales and easily follow up on the current discussion.

But soon also the problems of using the wiki for capturing the argumentation showed. First, the wiki required a lot of gardening in order to make the content easy to browse. In addition, although the decisions were described inside the wiki, there was no way to actually export this knowledge from the wiki and to reuse it. Therefore, the ontology engineers had to keep two independent resources up to date: the wiki, with the rationale and description of the ontology, and the ontology itself in an ontology engineering workbench. Visualisations of the ontology were copied from the engineering environment to the wiki in order to support the use of the wiki as a tool for comprehending the ontology. There was no automated interaction between the wiki and the ontology development environment.

13.3 Semantic MediaWiki

Based on our experiences with wikis, we decided to design a system that overcomes the identified problems. We aimed at integrating semantic features, while retaining as much as possible of the unstructured wiki workflow.

We have identified the link structure within the wiki as one of the major features that allow to access to the wiki content, and especially it was the only one that was provided explicitly by the user. We decided to enhance the user's ability to refine that link structure by allowing not just to add arbitrary links between pages, but also to *type* these links – an idea previously employed by the hypertext community. We further allowed to add attributes to pages, and to query and export the link and attribute structure. In the following sections, we will describe all of these features.

Instead of implementing a complete wiki system from scratch, we decided to reuse an existing one. This allows us to offer the usual capabilities of current wikis, without having to reimplement features like page histories, access rights, and edit conflict resolution. We decided to use the MediaWiki engine that is used by Wikipedia – the largest wiki in terms of content and users – as well as by thousands of other websites (Völkel et al. 2006). MediaWiki is an open source project with an active community of developers and a vast array of resources. It also offers sound mechanisms for implementing extensions, so that we could develop new features without having to fork the project. Users of the semantic extension, which we dubbed *Semantic MediaWiki*, therefore profit from any future improvements of MediaWiki, and existing wikis can easily be upgraded (Krötzsch et al. 2006).

MediaWiki and the Semantic MediaWiki extension are written in PHP and work on a MySQL database, and thus require only a minimal server environment. MediaWiki has the advantage of offering argumentation support within the wiki: each page in a MediaWiki installation has a discussion page. This offers an ideal environment for collaborative work, since the content and its discussion are, although tightly connected, still separated. It also maps naturally to our requirement for capturing the argumentation: whereas the decisions and the documentation of the ontology can remain short and succinct, the rationale and the argumentation that lead to that decision would be on the accompanying discussion page.

13.3.1 Relations

In MediaWiki, authors can link to a page by taking the page name and enclosing it into double square brackets, e.g. `[[SEKT]]`. This will create a link to the page called SEKT, where one will find the description of the SEKT project. If such a page does not exist, clicking on the link allows users to create a new of if this name.

A wiki with information about European projects would include pages about the projects, but also about the partners within these projects. Therefore, a page about the University of Karlsruhe would describe the university and mention, that they are partners in the SEKT project. Furthermore, reviewers may be linked to the SEKT project, events that were sponsored by it would link to it, as well as publications and deliverables. The SEKT project itself would link to the topics of its work, and describe its results, linking to further pages describing software systems developed within the project, etc. Now all these links between the different pages do have a certain meaning: the University of Karlsruhe is not just somehow linked to the SEKT project, but actually the relation may be well defined: the university is a *project partner*.

In Semantic MediaWiki, we can make this relation explicit by typing the link leading from the University of Karlsruhe page to the SEKT page. For this, we extend the link with a type: [[is partner in::SEKT]]. Within the link, a double colon separates the link type from the link target. Based on such types, the wiki engine is able to distinguish the relation between the university and the project e.g. from the relation between a publication and a project.

There is no set of predefined relations that editors are restricted to when describing the structure of the wiki content. Editors can introduce new relations as they deem fit, just as they can introduce new pages anytime. Since a consistent application of the emerging vocabulary provides useful, every relation has its own article that describes its proper usage.

13.3.2 Attributes

In order to add information to a page that cannot be expressed by relations, the user can use attributes. For example, in order to state the number of students at the University of Karlsruhe, we would not want to create a page describing that number. Instead we add an attribute value in a syntax very similar to the typed links introduced above: `[[number of students: = 16,000]]`. Note the equal sign. This does not create a link to the non-existing page called 16,000, but rather adds the metadata that the object described on this page has an attribute called "number of students" with the value of 16,000. As we will see shortly, such attributes enable the wiki to sort and filter.

As in the case of relations, the attribute "number of students" has a page of its own describing that attribute. Unlike relations, these pages are not optional: in order for an attribute to function, the *(data)type* of the values needs to be defined on the attribute's page. In order to properly parse the values, the wiki must know whether a value is a number, string, date, some physical quantity, or some other type. For instance, the values "16,000" and "16000" are equal as integer numbers, but not as strings. The list of available types is extensible.

The type system supports unit handling, and can convert values between different units of measurement. This means that editors can add the duration of the SEKT project as being 36 months, and the system knows that this is longer than another project that runs for ten weeks. It can also be used to display a value in different units. For example, the length of a river can be displayed both in miles and kilometres.

13.3.3 Categories

Categories allow to group pages. They have been introduced to MediaWiki in 2004, and its enthusiastic deployment within MediaWiki-powered wikis exposed the need for further abilities to structure the wiki content. In order to categorise the SEKT project, an editor may add the `[[Category:Project]]` to the page. A new page will describe the category and provide a list of all pages categorised with this category. It is further possible to create subcategories of categories. A reader can browse through the category tree in order to explore the content of the wiki.

Editors can introduce categories on the fly. Nevertheless, a usable and helpful category system emerged even on very large wikis like Wikipedia, with tens of thousands of unorganised authors. This also created a valuable resource, providing insights into how the category system evolved (and still is evolving), and which control and discussion mechanisms had to be established by the user community.

13.4 Browsing and searching

The purpose of adding relations, attributes, and categories in Semantic MediaWiki is to simplify common tasks for users of the wiki. Most importantly, SMW includes various features for browsing, searching, and aggregating the wiki's content. A simple example is the *Factbox* that summarises the semantic content of each article at its bottom. It also includes links by which users can browse the wiki's semantic content.

A more elaborate way of exploiting semantic data within the wiki is the semantic search. Semantic MediaWiki includes an easy-to-use query language which enables users to access the wiki's knowledge.

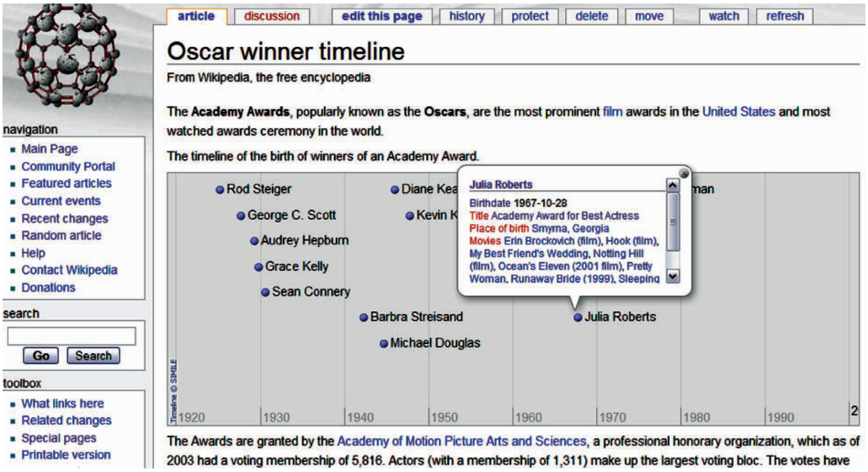
The syntax of the core query language is very similar to the syntax of annotations in Semantic MediaWiki. For example, the query [[Category:Project]] asks for all pages within the category "Project", and the query [[is partner in::SEKT]] asks for all partners of the SEKT project. It is possible to combine multiple such conditions, and to use more general descriptions instead of concrete article names. For example, consider the following query: `[[Category:University]] [[is partner in::+]] [[number of students: = > 10,000]]`.

It will return all articles that belong to the category "University", are partner in *something*, and have *more than* 10,000 students. Further expressivity is added by allowing subqueries, that could be used, e.g., to restrict the above query to universities which are partners within something that is an EU-funded project.

Queries as the above simply return a list of pages. It is possible to gather more information about these results by adding additional *print requests* to the query, usually described as a relation or attribute with an asterisk as its object. For example, the query `[[is partner in::SEKT]] [[location city::*]]` still returns all pages that describe partners in SEKT, but additionally shows for each of them the objects of the relation "location city" if any.

The above query language can be used in various ways within the wiki. On the one hand, knowledgeable users can of course directly enter queries and browse their results. It is also possible to directly link to result pages in order to save other users from having to type queries.

A much easier way of letting uninitiated users exploit the querying capabilities is the use of so-called inline queries. These are queries that wiki authors write directly into articles, enclosed into the special tag < ask >. Instead of those queries, the displayed article shows their up to date results, thus enabling all readers to view the latest query results without ever having to write queries. Many different output formats for query results enable a seamless embedding into wiki pages, so that casual readers do not notice that parts of the viewed page are generated dynami-



Oscar winner timeline

From Wikipedia, the free encyclopedia

The **Academy Awards**, popularly known as the **Oscars**, are the most prominent film awards in the United States and most watched awards ceremony in the world.

The timeline of the birth of winners of an Academy Award.

Julia Roberts
Birthdate 1967-10-28
Title Academy Award for Best Actress
Place of birth Smyrna, Georgia
Movies Erin Brockovich (film), Hook (film), My Best Friend's Wedding, Notting Hill (film), Ocean's Eleven (2001 film), Pretty Woman, Runaway Bride (1999), Sleeping

The Awards are granted by the Academy of Motion Picture Arts and Sciences, a professional honorary organization, which as of 2003 had a voting membership of 5,816. Actors (with a membership of 1,311) make up the largest voting bloc. The votes have

Fig. 13.1 Inline query with birth dates of Academy Award winners

cally (one example is given in Fig. 13.1). They might note, however, that the pages are always up-to-date and consistent with data stored elsewhere in the wiki.

13.5 Semantic MediaWiki in Terms of OWL DL

In this section, we relate the formal information gathered within the wiki system to the Web Ontology Language OWL (Smith et al. 2004). OWL DL is conceptually related to description logics, and one can divide ontological elements into *instances* that represent individual elements of the described domain, *classes* that represent sets of individuals, and *roles* that represent binary relations between individuals. Semantic MediaWiki's knowledge representation was inspired by OWL DL, and one can naturally relate the elements of the wiki, i.e. the individual content pages, to the basic vocabulary of OWL. Technically, the MediaWiki system employs namespaces to distinguish several types of content pages, and the semantic interpretation exploits this mechanism of "typing" pages (Vrandečić and Krötzsch 2006).

OWL individuals are represented by normal article pages. These pages typically constitute the majority of the wiki's contents, given by the pages of MediaWiki's main namespace, and most other namespaces such as *Image* or *User*.

OWL classes in turn have natural counterparts in the wiki in form of MediaWiki categories, which are represented as pages within the *Category* namespace. They can be organised in a hierarchical way, but it is not possible to make a category contain other categories. Thus MediaWiki's category system is more similar to the semantics of classes in OWL DL than to the semantics of classes in RDFS.

OWL properties do not have a counterpart in MediaWiki, and are introduced only by Semantic MediaWiki. OWL further distinguishes object-properties

Table 13.1 Typical representation of OWL constructs in Semantic MediaWiki

OWL	Semantic MediaWiki
OWL individual	Normal article page
owl:Class	Article in namespace Category
owl:ObjectProperty	Article in namespace Relation
owl:DataTypeProperty	Article in namespace Attribute
Statement about element page	Syntax in wiki-source of page
object-property	[[property_name::object_value]]
attribute-property	[[property_name: = value_string]]
rdf:type class_name	[[Category:class_name]] (on article pages)
rdfs:subClassOf class_name	[[Category:class_name]] (on category pages)

(describing relationships between two individuals), data-properties (associating individuals with values of a given datatype), and annotation-properties (having almost no impact on the model-theoretic semantics). Relations are closely related to object-properties between elements of the wiki. A minor issue is that they can be used on elements that do not represent individuals, e.g. by establishing a relation between two categories. In this case, the category is treated as an individual different from the class that it normally represents, a method known as *punning* that is consistent with all uses of annotations in the wiki.

Attributes may represent any kind of property in OWL DL, depending on the (wiki) datatype they have been given. In the vast majority of cases, they are used as data-properties with a single XML Schema type that is used consistently, but it is also possible to encode annotations or object properties to external objects as attributes.

An incomplete mapping from OWL language elements to the semantic wiki is given in Table 13.1. Various further OWL statements about some OWL individual/ class/role are specified in Semantic MediaWiki by annotating the respective wiki page. The formal correspondence to OWL DL is also utilised for exporting semantic data in the OWL/RDF format. For this purpose, Semantic MediaWiki associates URIs with wiki pages. These URIs bijectively correspond to concrete pages, but are not identical to the article URLs. This prevents confusion between ontological concepts (e.g. the SEKT project) and actual HTML documents (e.g. the article about SEKT).

13.6 Conclusions

Semantic MediaWiki influenced by the experiences of the DILIGENT methodology, and by experiments in wiki-based ontology engineering within SEKT project case studies. While wikis turned out as viable solutions for ontology engineering, off-the-shelf wikis could not fulfil this task appropriately, and other existing semantic wikis at that time (and even today) hardly provided the essential argumentation features. Semantic MediaWiki would have allowed to build the ontology within the wiki and then to export the constructed ontology from the wiki without having to maintain and rebuild it manually in a second development environment.

Due to the late development of Semantic MediaWiki within the project, we could not apply the implementation within the SEKT case studies any more. However, the system is now used on numerous wikis world-wide. Being free software that is fully compatible with one of the most successful wiki engines, it has been applied in many different scientific, industrial, and purely recreational scenarios. Following user requests, the system was translated into various languages, and is currently available in English, French, German, Spanish, Hebrew, and Slovak. Semantic MediaWiki is further developed very actively, both in the context of current and upcoming research projects, and through the contributions of many volunteers.

Semantic MediaWiki makes semantic knowledge available on the Semantic Web via its OWL/RDF export, thus facilitating reuse in external tools, such as semantic search engines or browsers. Since RDF processing libraries are available in all major programming languages, writing specialised tools that read and process a wiki's data is also possible. The availability of all data in standardised and open formats thus might, at least in the long run, turn out as being the most important advantage of using systems like Semantic MediaWiki.

Acknowledgements This work was supported by the IST Programme of the European Community under SEKT (IST-IP-2003-506826).

References

Casanovas P, Casellas N, Tempich C, Vrandečić D and Benjamins R (2007) OPJK and DILIGENT: ontology modeling in a distributed environment, Artificial Intelligence and Law, Vol. 15, No. 1.

Cunningham W and Leuf B (2001) The Wiki Way. Quick Collaboration on the Web. Reading, MA, Addison-Wesley.

Krötzsch M, Vrandečić D and Völkel M (2006) Semantic MediaWiki, International Semantic Web Conference ISWC2006, Athens, GA.

Smith M K, Welty C and McGuinness D L (2004) OWL Web Ontology Language Guide, W3C Recommendation, Boston, MA.

Tempich C (2006) Ontology Engineering and Routing in Distributed Knowledge Management Applications, Karlsruhe, Germany, PhD Thesis, Universität Karlsruhe (TH).

Völkel M, Krötzsch M, Vrandečić D, Haller H and Studer R (2006) Semantic Wikipedia, World Wide Web Conference WWW2006, Edinburgh, UK, ACM Press, New York.

Vrandečić D and Krötzsch M (2006) Reusing Ontological Background Knowledge in Semantic Wikis, First Workshop on Semantic Wikis – From Wikis to Semantics, Budva, Montenegro.

Chapter 14
Deploying and Evaluating Semantic Technologies in a Digital Library

Ian Thurlow and Paul Warren

Abstract Digital libraries have been the subject of considerable research since the 1990s. Their practical value in providing remote access to knowledge is beyond question. As they have developed, there have been numerous attempts to theorize about their future nature and role, and to lay down the challenges for further work. On a more practical level, individual organisations have developed their own digital libraries in response to their particular needs. This chapter describes research into the use of semantic technology in a Digital Library. The work draws on the technologies and tools described elsewhere in the book and puts them in the context of a particular application. The chapter also explains in some detail how user trials within the case study were used to validate our approach.

14.1 Introduction

Digital libraries have been the subject of considerable research since the 1990s. Their practical value in providing remote access to knowledge is beyond question. As they have developed, there have been numerous attempts to theorize about their future nature and role, and to lay down the challenges for further work. On a more practical level, individual organisations have developed their own digital libraries in response to their particular needs. One such is the BT Digital Library, which has evolved over the last dozen years in response to the needs of knowledge workers in BT. This chapter describes research into the use of semantic technology in the BT Digital Library. The lessons, however, are applicable to digital libraries in general. The work draws on the technologies and tools described elsewhere in this book and puts them in the context of a particular application. The chapter also explains in some detail how user trials within the case study were used to validate our approach.

I. Thurlow (✉)
Next Generation Web Research BT Research, Ipswich IP5 3RE, UK
e-mail: ian.thurlow@bt.com

J. Davies et al. (eds.), *Semantic Knowledge Management*,

14.2 The BT Digital Library

The BT Digital Library,[1] which formed the starting point for our case study, contains abstracts, and in many cases full text, of much of the literature of interest to BT's technical specialists, besides a significant amount of relevant non-technical literature. In all, this represents five million articles from over 1,200 publications, including journals, conference proceedings and IEEE Standards. This is provided in the form of two databases, Inspec[2] and ABI/INFORM from ProQuest.[3] Users search the content of the library using a keyword-based search engine. The search engine offers high recall and, although it is possible to narrow down the number of results returned through careful composition of the query, most users only formulate very simple queries; as a result being presented with a large number of search results. This was the starting point on which we hoped to build and improve through the use of semantic technologies.

14.2.1 The Challenges

In a previous book (Davies et al. 2006) the authors have described the challenges facing digital libraries. Most prominent amongst these is the goal of semantic interoperability (Chen 1999). In practical terms this means providing each user with a unified view of digital objects across libraries. Other challenges included the need for improved user interfaces to navigate large information collections (Lynch and Garcia-Molina 1995) and the need to match concepts, not just search strings (NSF 2003).

For us, semantic interoperability meant the ability to integrate relevant information on the Web or corporate Intranet with information in the Digital Library.

Improving the user interface and searching on concepts were particularly prominent challenges for us, as they gave us the opportunity to address some of the weaknesses of the search tools in use in the library. Our response to these two challenges is represented by the user tools *Squirrel*, a semantically-enabled search and browse tool, and *SEKTagent*, a semantically-enabled search agent application. These tools are described elsewhere in this book and in Duke et al. (2006). Their use in the Digital Library is summarised later in this chapter.

A long-standing goal of the BT Digital Library was to be not just a repository of information, but also an enabler of communities of interest, where knowledge could be exchanged and people of similar professional interests could make contact. The BT Digital Library was not alone in this. For example, the Perseus Digital Library

[1] The BT Digital Library has been developed under the leadership of the authors' colleague, David Alsmeyer

[2] http://www.iee.org/Publish/INSPEC/

[3] http://www.proquest.com

project[4] has a commitment to "connect more people through the connection of ideas". Early in the development of the Library, the concept of information spaces was introduced (Alsmeyer and Owston 1998). On one level each information space is a collection of documents, defined by a particular query agent. On another level, the information space is the group of people who register to receive regular alerts from the agent. A companion chapter of this book describes the use of semantic technology for knowledge sharing, in the form of the *Squidz* tool, and compares this with the pre-existing information spaces.

14.2.2 Understanding User Needs

Before beginning development we sought the views of users to validate our ideas. This is described in some detail in Davies et al. (2006), but a brief overview is given in this section.

Initially a questionnaire was used to obtain a general view of user requirements. There was then a focus group drawn from a small number of the Library users. Again, the questions posed to the group were quite general. We learned that users wanted improved ways of searching; and that they wanted searches to take into account their profile of interest.

A second questionnaire was used to test users' reactions to our proposed functionality. This confirmed the value of the planned functionality and gave us an understanding of the users' perceived priorities. The majority of the features valued by users were subsequently incorporated into *Squirrel*, a semantic search and browse application.

A final technique for understanding users' requirements was that of user preference analysis.[5] We used this to understand the trade-off, from the user's perspective, between precision and recall. Semantic technology can help us improve both. Precision can be improved by the capability to specify more precisely the entity sought, for example, that the string "BT" represents a company in the telecommunications sector. Recall is enhanced by the ability to recognise different words or phrases as synonyms. When the results of our preference analysis were analysed there were two clusters of users. In one cluster users had a clear preference for precision over recall. In the other cluster users gave equal weight to precision and recall. This led, in our user trials, to the adoption of a different measure of information quality, specifically oriented to the end-user perspective.

In all, the combination of questionnaires, focus group and user preference analysis provided an understanding of user requirements to guide the development of the end-user tools.

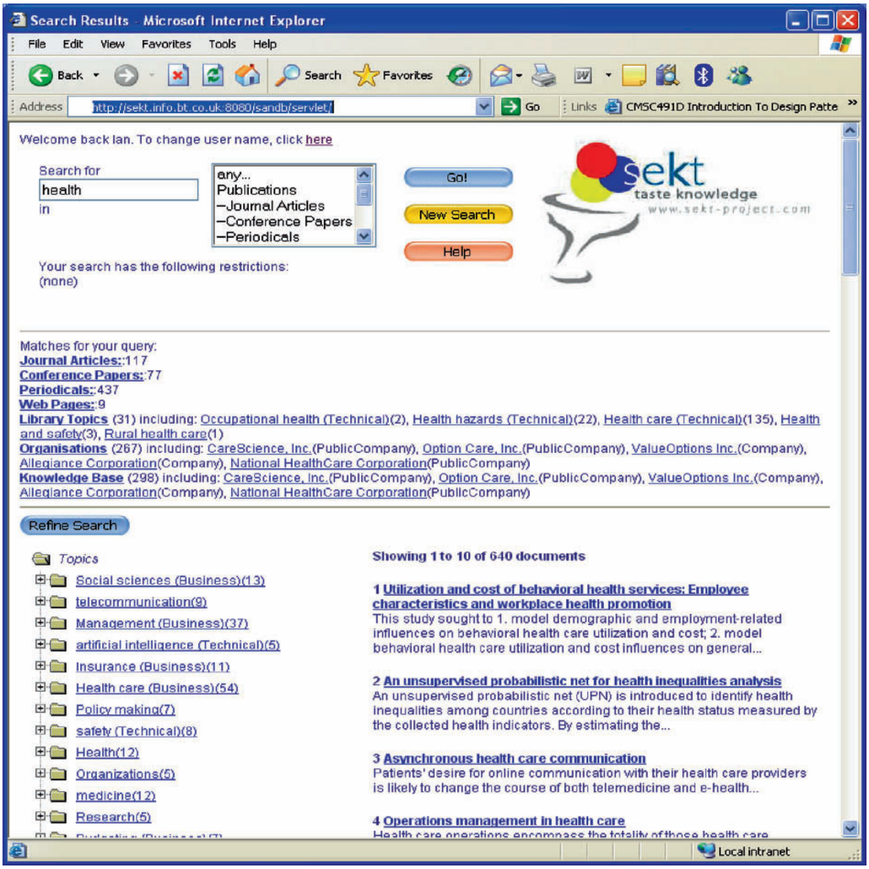
[4] http://www.perseus.tufts.edu

[5] The user preference analysis was undertaken with the guidance and assistance of Professor Tom Bösser, of kea-pro GmbH.

14.3 Squirrel – A Semantic Search and Browse Application

Squirrel was designed to improve access to digital library content, and to overcome some of the limitations presented by the search engine that is currently in use in BT's digital library. *Squirrel* combines free-text search with the means to query and browse ontology-based semantic meta-data in the form of a Web Ontology Language (OWL)[6] shared ontology. A collection of bibliographic records from Inspec and ABI is integrated with information sourced from the Web using a common topic ontology.

The user interface to *Squirrel* is shown in Fig. 14.1. A complete description of the main functions are described in a companion chapter in this volume. Users enter their


Search Results - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
Go
Links CMSC491D Introduction To Design Patte
Go!
New Search
Help
sekt
taste knowledge
www.sekt-project.com
Refine Search
Local intranet

Address: http://sekt.info.bt.co.uk:8080/sandb/servlet/

Welcome back Ian. To change user name, click here

Search for: health

in: any... / Publications / –Journal Articles / –Conference Papers / –Periodicals

Your search has the following restrictions:
(none)

Matches for your query:
Journal Articles::117
Conference Papers::77
Periodicals::437
Web Pages::9
Library Topics (31) including: Occupational health (Technical)(2), Health hazards (Technical)(22), Health care (Technical)(135), Health and safety(3), Rural health care(1)
Organisations (267) including: CareScience, Inc.(PublicCompany), Option Care, Inc.(PublicCompany), ValueOptions Inc.(Company), Allegiance Corporation(Company), National HealthCare Corporation(PublicCompany)
Knowledge Base (298) including: CareScience, Inc.(PublicCompany), Option Care, Inc.(PublicCompany), ValueOptions Inc.(Company), Allegiance Corporation(Company), National HealthCare Corporation(PublicCompany)

Topics
- Social sciences (Business)(13)
- telecommunication(9)
- Management (Business)(37)
- artificial intelligence (Technical)(5)
- Insurance (Business)(11)
- Health care (Business)(54)
- Policy making(7)
- safety (Technical)(8)
- Health(12)
- Organizations(5)
- medicine(12)
- Research(5)

Showing 1 to 10 of 640 documents

1 Utilization and cost of behavioral health services: Employee characteristics and workplace health promotion
This study sought to 1. model demographic and employment-related influences on behavioral health care utilization and cost; 2. model behavioral health care utilization and cost influences on general...

2 An unsupervised probabilistic net for health inequalities analysis
An unsupervised probabilistic net (UPN) is introduced to identify health inequalities among countries according to their health status measured by the collected health indicators. By estimating the...

3 Asynchronous health care communication
Patients' desire for online communication with their health care providers is likely to change the course of both telemedicine and e-health...

4 Operations management in health care

Fig. 14.1 The user interface to Squirrel

[6] http://www.w3.org/2004/OWL

search query in the search box, and optionally select the type of information they need from the drop-down menu, for example, users can search for publications authored by a particular person, for publications of a particular type (e.g., a conference paper), or for items which mention a specific company or one of its aliases. Alternatively, for users wanting to invoke a broader search, the search application offers a search which covers all types of content held in the digital library. The user's query is formed using concepts from the ontology. Search results are ranked, taking into account user profiles to give a degree of context to the user's search (the user profile is constructed from an analysis of a user's interaction with BT's digital library and other Web information sources, and is described through a set of interests that are defined in the ontology). Users can refine their query by selecting from a set of related topics returned with the results of a search.

Squirrel identifies and highlights *named entities* within the search results, for example, the names of people, names of companies, names of organisations, location names, etc. Supplementary information concerning those entities is presented to the user, for example, for an entity of type *company*, additional information such as the location of its headquarters, its web address, the sector it operates in, and details of its key personnel are presented to the user. The rationale here is that the identification and presentation of named entities should give the end user information additional to that in the original text, and help the user to more swiftly identify the context of the results from a search, thereby enabling the results to be filtered based on entities of interest.

Users can refine the search by specifying the type of document (e.g., journal article, conference paper, periodical, web page), library topic, or name of company identified in the document. Details of any organisations matching the search criteria are also presented to the user. In addition to searching against the metadata that describes the information entities, the user can also browse the topic hierarchy to find relevant documents. Users are able to navigate up or down the topic hierarchy, expanding or refining their search as they go, for example, should a search on a particular topic give no useful result, users can make use of super topics of the original topic to broaden their search. Conversely, if there are too many results, users can make use of subtopics to narrow their search.

Besides searching and browsing by topic, the user can also browse the library using other characteristics of a document, for example, by requesting other documents by the same author, or papers published in the same conference proceedings. The refine search option enables a set of results to be filtered by searching specific fields of the records, for example, the title or the abstract. A user can also filter the results set by date or by author name.

14.4 SEKTagent – A Search Agent Application

SEKTagent collects relevant content from a pre-indexed set of documents on behalf of a user. This set includes the ABI and Inspec abstracts (and full-text where available), Web pages and RSS items. The application highlights named entities identified

within the search agent's results, and links those entities to relevant supplementary information in the knowledge base. The *SEKTagent* application uses an API provided by the KIM platform (Kiryakov et al. 2003). End users configure each agent with a semantic query making use of the BT digital library ontology. Some example *SEKTagent* queries, shown in natural language, are given below:
Find all intranet pages that contain a named person holding a particular position within a user-specified organisation.

- Find all conference papers authored by a particular person.
- Find all articles where the author is affiliated with a named organisation.
- Find all publications mentioning a named company that is active in a particular industry sector.

This mode of searching for types of entity is complemented with a full text search, allowing the user to specify terms that should occur in the text of the retrieved documents. The results page for a *SEKTagent* that is searching for any *person* that holds the *job position* of *analyst* within an *organisation* that is *located* in *any location* is shown in Fig. 14.2.

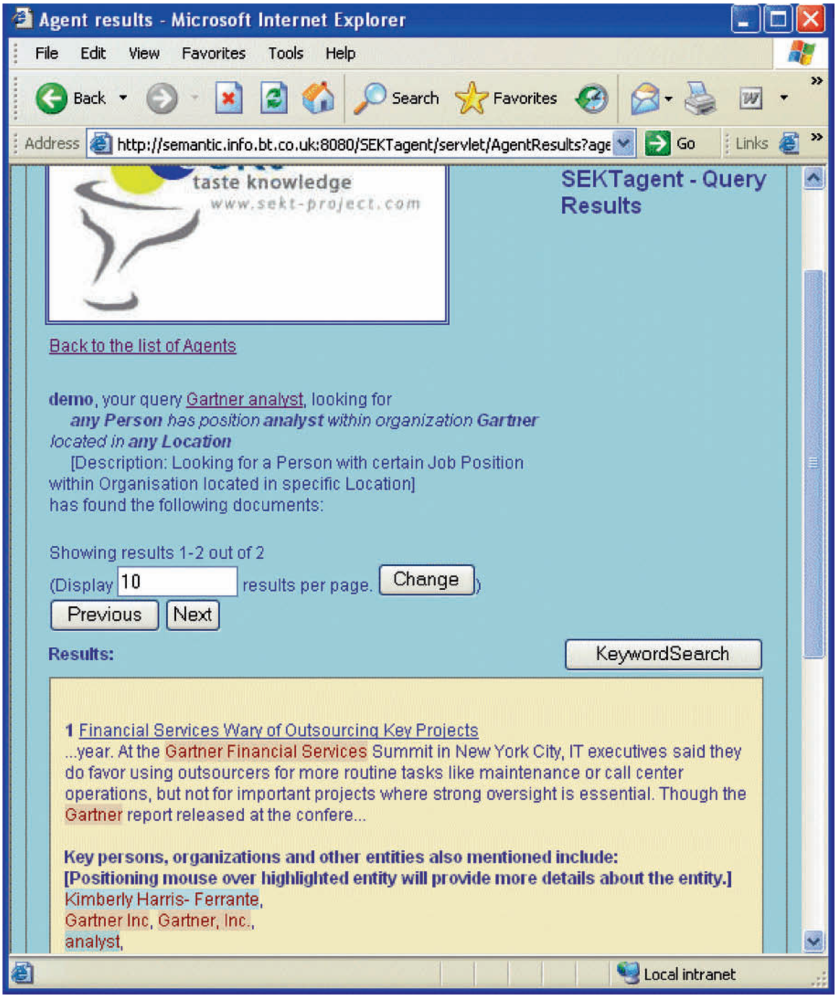
The title of the Web page, or title of the abstract/full-text for ABI/Inspec records, and a short summary of the content relevant to the query are displayed for each result. The summary highlights the occurrences of the named entities that satisfy the query. Other recognised named entities are also highlighted. In a similar way to the *Squirrel* search and browse application, the user is able to access further information about the entity from the knowledge base by placing their mouse pointer over any of the named entities that are highlighted.

14.5 Evaluation Process

The *Squirrel* search and browse and *SEKTagent* search agent applications were subjected to a three-stage user-centred evaluation process. The first two stages of the process, which comprised a *heuristic evaluation* (Nielsen and Molich 1990) and a *cognitive walkthrough* (Wharton et al. 1992), aimed to identify and correct any bugs or navigational problems with the applications. The final stage of the process comprised a set of *field tests* that aimed to evaluate the applications against the search engine currently in use in BT's digital library.

14.5.1 Heuristic Evaluation

An informal heuristic evaluation of the user interface was undertaken, with the primary objective to assure that the functions of the applications worked as intended. The aim was to identify and correct all significant defects and user interaction problems in the early stages of development.


Agent results - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://semantic.info.bt.co.uk:8080/SEKTagent/servlet/AgentResults?age
Go Links
taste knowledge
www.sekt-project.com
Local intranet

SEKTagent - Query Results

Back to the list of Agents

demo, your query Gartner analyst, looking for
*any **Person** has position **analyst** within organization **Gartner** located in **any Location***
[Description: Looking for a Person with certain Job Position within Organisation located in specific Location]
has found the following documents:

Showing results 1-2 out of 2
(Display 10 results per page. Change)
Previous Next

Results: KeywordSearch

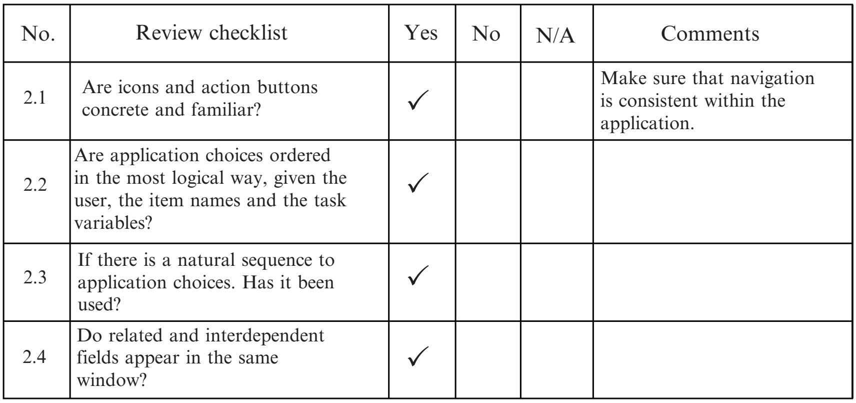
1 Financial Services Wary of Outsourcing Key Projects
...year. At the Gartner Financial Services Summit in New York City, IT executives said they do favor using outsourcers for more routine tasks like maintenance or call center operations, but not for important projects where strong oversight is essential. Though the Gartner report released at the confere...

Key persons, organizations and other entities also mentioned include:
[Positioning mouse over highlighted entity will provide more details about the entity.]
Kimberly Harris- Ferrante,
Gartner Inc, Gartner, Inc.,
analyst,

Fig. 14.2 SEKTagent results page

A group of five researchers, acting as usability experts, judged whether the user interface of early prototypes of *Squirrel* and *SEKTagent* adhered to a list of established usability principles known as the *usability heuristics*. The usability heuristics were based on a checklist adapted from the Xerox heuristic evaluation system checklist.[7] Each expert inspected the prototype during a session lasting ~1 h. A number of observations were made during each session, the majority of which tended to be concerned with minor interface problems and system performance issues. An example is shown

[7] http://www.stcsig.org/usability/topics/articles/he-checklist.html



No.	Review checklist	Yes	No	N/A	Comments
2.1	Are icons and action buttons concrete and familiar?	✓			Make sure that navigation is consistent within the application.
2.2	Are application choices ordered in the most logical way, given the user, the item names and the task variables?	✓			
2.3	If there is a natural sequence to application choices. Has it been used?	✓			
2.4	Do related and interdependent fields appear in the same window?	✓			

Fig. 14.3 Excerpt from system checklist

in Fig. 14.3. The findings of the inspection were collated and a measure of severity was assigned to each problem. The observations were discussed with the development team. The applications were then modified in accordance with the observations.

14.5.2 Cognitive Walkthrough

The heuristic evaluation provided a simple and effective method for identifying many of the problems with the user interface. It did not, however, reveal all problems that a user may experience when using an application. The primary aim of the cognitive walkthrough was to find out whether each application functioned as expected under simulated operational conditions. A group of five users were asked to use the applications to undertake a set of short information seeking tasks. Each task was designed to make use of one or more of the key functions of the application. Users were encouraged to explain their actions and their concerns as they undertook each task, thus giving the assessors additional information about their thought processes and an early indication on the overall usability of the application. The findings of the evaluation were discussed with the development team. Each application was then modified to resolve all major usability issues that were identified during the evaluation.

14.5.3 Field T~ests (Squirrel and SEKTagent)

The next part of the evaluation comprised a user test of *Squirrel* and *SEKTagent* applications against the current search engine in use in BT’s digital library. Twenty users with a wide range of experience of using search tools were invited to take part in these field tests. In the case of *Squirrel*, we also assessed whether the information search is more efficient in terms of information quality and time (information quality was assessed as a subjective measure of the relevance of the results to a user's information need). In essence, we wanted to find out whether the new technology would help people to get to the information they believed to be relevant to their search more easily and quickly.

A test environment, which comprised *Squirrel, SEKTagent*, and the current digital library search engine, was configured to give access to ~37,000 bibliographic records and 2,000 web documents. The field tests undertaken by each user comprised three distinct parts: (1) demonstration of the key search functions of *Squirrel, SEKTagent*, and the current search engine, followed by completion of a short questionnaire, (2) completion of information seeking tasks using *Squirrel* and the current search engine, and (3) completion of a Software Usability Measurement Inventory (SUMI) (Kirakowski and Corbett 1993) questionnaire for *Squirrel*.

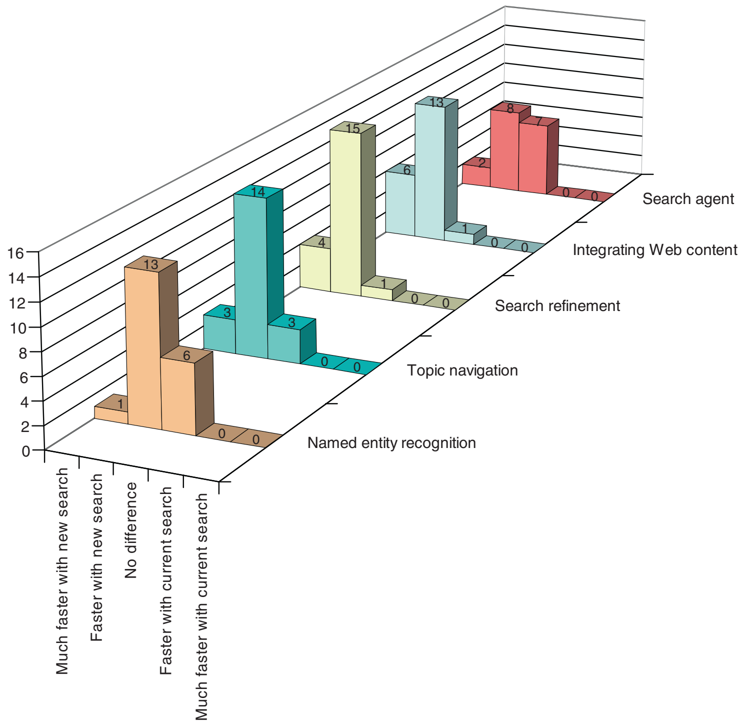
14.5.3.1 Demonstration of New Search Functions

The first part of the evaluation aimed to give the subjects some time to familiarise themselves with the main functions of *Squirrel, SEKTagent* and the current search engine. It also gave us the opportunity to gather some initial user feedback on the new semantic search functions in comparison with current search functions. The following functions were demonstrated: (1) named entity recognition in *Squirrel*, (2) navigation and browsing using the topic ontology in *Squirrel* against use of controlled indexing terms for navigation and browsing using the current search technology, (3) search refinement in *Squirrel* and the current search engine, (4) integration of Web content in *Squirrel*, and (5) the *SEKTagent* semantic search agent functions against use of the current information spaces implementation.

After each function was demonstrated, and after the subject had spent some time using the functions of the applications to complete a simple search task, each subject was asked to provide some qualitative feedback with respect to that task. For each function, we asked the subjects to answer to the following questions (by rating on a 4 or 5 point scale[8]):

- Does the new function offer an improvement compared to the equivalent function in the current search engine?
- Does the information seeking task become easier to complete with the new search and browse application (*Squirrel* and *SEKTagent*) in comparison with the current search engine?
- Do they expect to find information more quickly with the new search and browse application in comparison with the current search engine?
- Do they expect to find better quality information with the new search or the current search technology (where the relevance of results was to be taken as the main measure of quality)?

[8] Refer to Figure 14.4 through to Figure 14.7 for scales.



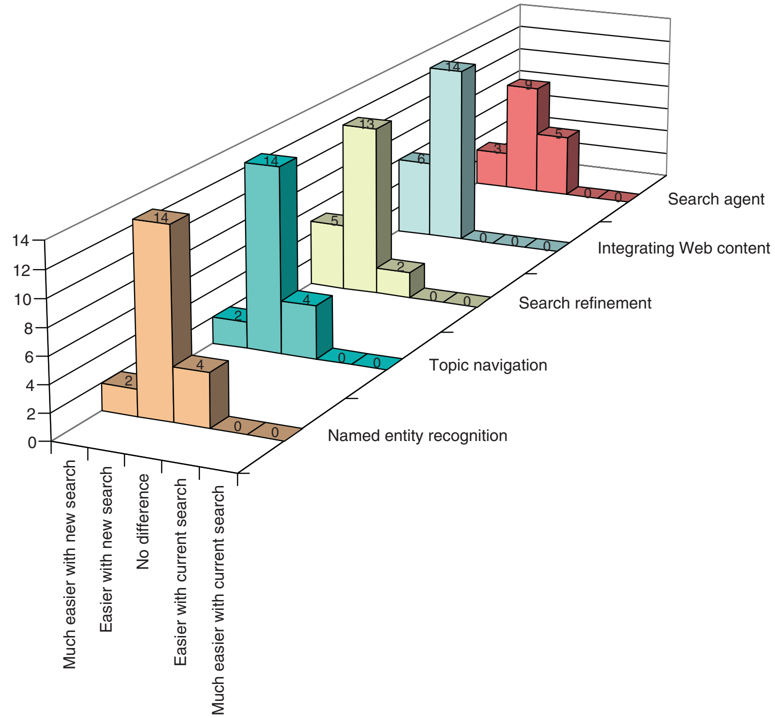
	Much faster with new search	Faster with new search	No difference	Faster with current search	Much faster with current search
🟧 Named entity recognition	1	13	6	0	0
🟩 Topic navigation	3	14	3	0	0
🟨 Search refinement	4	15	1	0	0
⬜ Integrating Web content	6	13	1	0	0
🟥 Search agent	2	8	7	0	0

Fig. 14.4 Speed of access to information for new search functions

The results of the evaluation are summarised in Figs. 14.4–14.7. Overall, there was a very positive response to the new functions provided in *Squirrel* and *SEKTagent*. In particular, the improved functionality for searching web content stands out, being rated very positively.

14.5.3.2 Information Search Quality Measures

One of the main quality dimensions that users will use to assess the value of a search and browse application is the effectiveness of their search. The fundamental value to the user is the quality of the results returned by the search engine in relation to their information need, that is, the problem that motivated their search. The effectiveness of their search also depends on their ability to carry out the search task efficiently, that is, successfully and timely, and without undue cognitive effort when using the application.

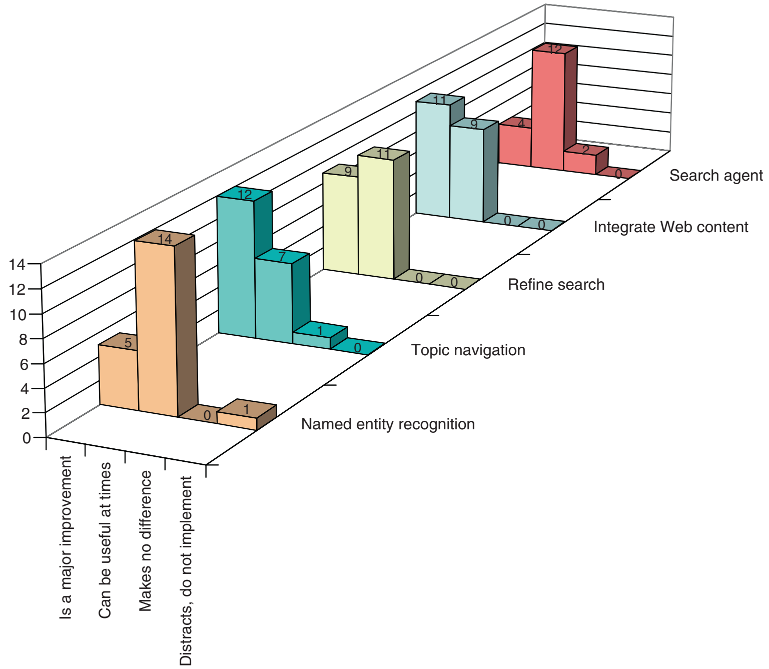


	Much easier with new search	Easier with new search	No difference	Easier with current search	Much easier with current search
🟧 Named entity recognition	2	14	4	0	0
🟦 Topic navigation	2	14	4	0	0
🟨 Search refinement	5	13	2	0	0
⬜ Integrating Web content	6	14	0	0	0
🟥 Search agent	3	9	5	0	0

Fig. 14.5 Ease of access to relevant information

The main goal of users is to maximise the efficiency of their information seeking process, that is, maximise the quality of the information returned by the search engine, whilst minimising any personal cost to them in terms of time and the cognitive effort required to obtain a set of results that they are satisfied with. Ideally, the gain in the quality of the information returned from the search and browse application in terms of relevance to a user's information need should increase as the user's search progresses.

Conventional measures of information retrieval systems (van Rijsbergen 1980) tend to focus on search precision (the proportion of relevant results that are retrieved to all the results that are retrieved), search recall (the proportion of relevant results that are retrieved to all relevant results available), and the F-measure (the weighted harmonic mean of precision and recall). The relevance of a set of search results is, however, a highly subjective measure. Search results that are highly relevant to one person's query are very likely to differ in terms of relevancy to another user using the same search engine and submitting the same query. Moreover, it is anticipated



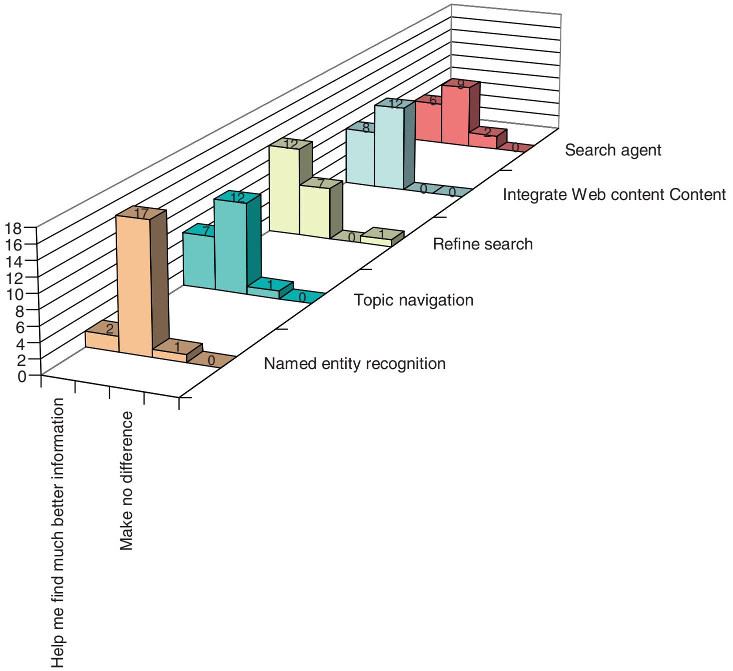
	Is a major improvement	Can be useful at times	Makes no difference	Distracts, do not implement
🟧 Named entity recognition	5	14	0	1
🟦 Topic navigation	12	7	1	0
🟨 Refine search	9	11	0	0
🟦 Integrate Web content	11	9	0	0
🟥 Search agent	4	12	2	0

Fig. 14.6 Improvement in search engine functions

that there is a shift in a user's perception of the relevance of a search result depending on the amount of relevant information that is accessible. When there is an abundance of relevant information, a user's acceptance criterion in terms of the quality of the search results is expected to be high, whereas in the opposite case, where only a few relevant or some marginally relevant articles are found, the acceptance level of the user is likely to be reduced. We therefore decided to take a different approach to measuring the effectiveness of the *Squirrel* search and browse application, replacing the more conventional measures of precision and recall in favour of an approach that enabled us to assess the quality of information returned from the search engine from an end user's perspective, that is, we concentrated on the subjective assessment of the information quality in our analysis.

14.5.3.3 Information Seeking Tasks

In order to gauge the effectiveness of *Squirrel*, we conducted a series of tests where users were asked to carry out defined tasks, with a defined objective, under controlled



	Help me find much better information	Might help me find better information at times	Make no difference	Preclude me from finding some information
🟧 Named entity recognition	2	17	1	0
🟦 Topic navigation	7	12	1	0
🟨 Refine search	12	7	0	1
⬜ Integrate Web content Content	8	12	0	0
🟥 Search agent	6	9	2	0

Fig. 14.7 Information quality expectations

conditions. Subjective user assessments were collected from users. Each user was asked to complete two similar, but separate, information seeking tasks from a set of six. One task was completed using *Squirrel*, the other was completed using the search and browse application currently in use in BT's digital library. The order in which a subject used the applications was alternated, so that half of the tasks were performed using the keyword-based search engine first, and the other half using the semantically enabled search engine first.

The user was presented with rating scales and asked to assess the subjective value of the results returned from the application as they completed their search. Users provided feedback on two key dimensions: (1) the perceived information quality (PIQ), that is, the quality of results returned from the search engine in relation to the task as perceived by the user, and (2) the perceived progress of search (PPS), that is, the progress of their search in relation to the task they were asked to undertake. PIQ was measured according to a 7-point scale, where *1 = unsatisfactory, 2 = poor, 3 = less than expected, 4 = as expected, 5 = above average, 6 = good* and 7 = *excellent*. PPS was measured according to a 15-point scale, where points 1–5 represented the beginning of the users search, points 6–10 represented revisions of their search, and 11–15 represented the refinement of their search as the task neared completion. Assessments of information quality were given by users at points in time that they determined themselves, for example, the moment before they submit a modified query rather than at a set time interval, so as to minimise interruptions to their search process.

Users typically provided four or five PIQ and PPS measures per task (average of 4.6). The average time for a user to undertake a search task was ~15 min (the minimum duration was 7 min; the maximum duration was 27 min). There was no significant difference in time between the two search tasks of users.

The average PIQ using the existing library system was 3.99 against an average PIQ of 4.47 using *Squirrel*. The PIQ scores given at various stages in the users' searches are shown in Fig. 14.8. The sign tests applied to the data shows the difference to be significant ($p < 0.01$). Note: in places where datapoints coincide (Fig. 14.8), one datapoint is shown inside the other.

14.5.3.4 Usability of Squirrel

Finally, the Software Usability Measurement Inventory (SUMI),[9] which gives a more detailed view of the subjective assessment of the usability of *Squirrel*, was

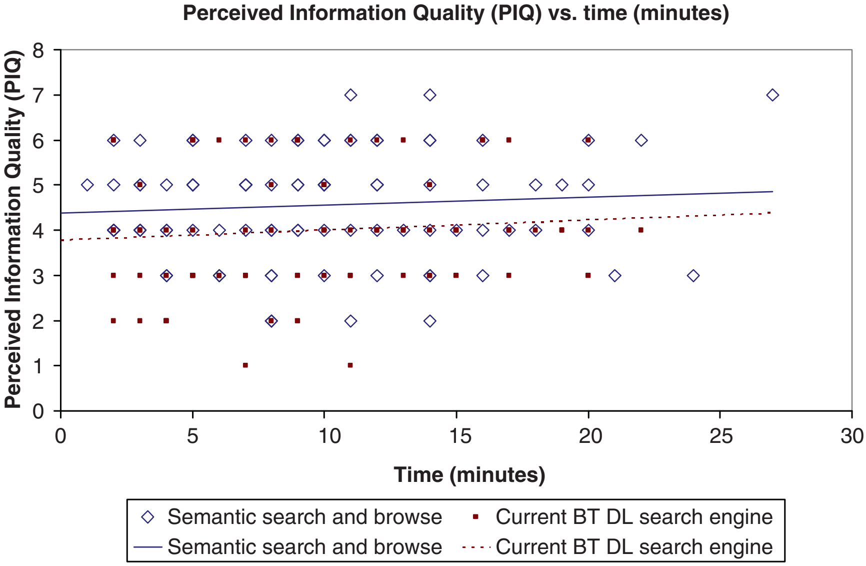


Fig. 14.8 Perceived information quality (PIQ) against time

[9] http://sumi.ucc.ie

administered to each subject. SUMI enables the following five separate aspects of user satisfaction to be measured: (1) efficiency, that is, the user's feeling that the software enables them to perform their tasks in a quick, effective and economical manner (Table 14.1), (2) affect, which refers to the positive user feeling of the user being mentally stimulated and pleased as a result of interacting with the software (Table 14.2), (3) helpfulness, which refers to the user's perceptions that the software communicates in a helpful way and assists in the resolution of operational problems, (4) control, which refers to the feeling that the software responds in an expected and consistent way to input and commands (Table 14.3), and (5) learnability, which refers to the feeling that it is relatively straightforward to become familiar with the software. A *global* metric describes the user's view of the overall usability of the application.

The SUMI questionnaire comprises fifty statements, each labelled with boxes *Agree, Undecided*, and *Disagree*. Users were asked to complete SUMI (for *Squirrel*) after they had completed the information seeking tasks (users are asked to answer all questions). Of the twenty SUMI questionnaires, one questionnaire was clearly incomplete and was excluded from the analysis. Three other questionnaires had

Table 14.1 SUMI item 1 (efficiency): This software responds too slowly to inputs

	Agree	Undecided	Disagree	
Profile	13	1	5	
Expected	3.61	2.9	12.49	
Chi Square	24.44	1.25	4.49	30.18[a]

[a]Indicates 99.99% confidence

Table 14.2 SUMI item 32 (affect): There have been times in using this software when I have felt quite tense

	Agree	Undecided	Disagree	
Profile	2	2	15	
Expected	7.19	2.89	8.93	
Chi Square	3.74	0.27	4.13	8.15[a]

[a]Indicates 95% confidence

Table 14.3 SUMI item 29 (control): The speed of this software is fast enough

	Agree	Undecided	Disagree	
Profile	6	1	12	
Expected	10.65	3.17	5.1	
Chi Square	2.03	1.48	8.99	12.51[a]

[a]Indicates 99% confidence

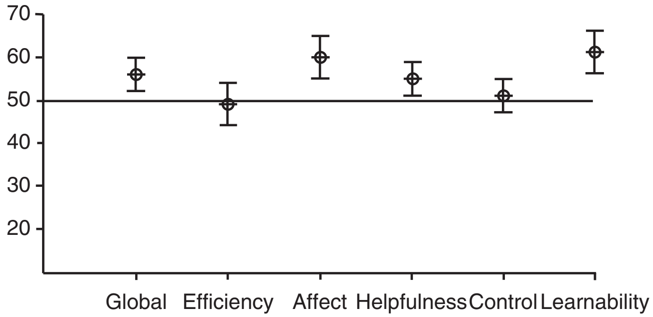


Fig. 14.9 Results of the SUMI analysis

between one and three statements unanswered (in the analysis, the responses to these questions was considered as being *Undecided*). An analysis of the responses given to the questionnaires was completed using the SUMI scoring program SUMISCO. The results of the analysis are shown in Fig. 14.9.

The vertical axis of Fig. 14.9 shows the normalized scale of the SUMI test, which is constructed to have a mean of 50, a total range of 100. The following tables show statements where the subjects made significant assertions about *Squirrel* (for brevity, only the statements where a significant statement was made are shown). Each table provides a description of the statement (e.g., *This software responds too slowly to inputs*), a *profile* of the user responses (the number of people who *agreed* or *disagreed* with the statement or who were *undecided*), the *expected* response (based on data from a standardisation database), and a *Chi Squared* measure (a measure of goodness of fit between the observed and the expected values). The standardisation database, which was produced from the responses of more than 1,000 users, is used to generate an expected pattern of response for each SUMI item. The expected pattern of response is then compared with the actual, obtained pattern. In summary:

- more users than expected agree that *Squirrel* responds too slowly to inputs, (99.99% confidence),
- more users than expected disagree that the speed of *Squirrel* is fast enough (99% confidence).
- more users than expected disagree that there have been times in using *Squirrel* when they have felt quite tense (95% confidence).

The users who completed the SUMI assessment were satisfied with the usability of *Squirrel* (Global usability was rated 4 scale points higher than the average). Furthermore, users liked using *Squirrel* (Affect was rated 8 scale points higher than the average), but do not seem to find the presentation of the user interface particularly attractive (a further analysis needs to be undertaken to identify the particular aspects of the presentation that the users were not satisfied with). *Squirrel* was considered easy to use (Learnability was rated 11 scale points higher than the average and Helpfulness was rated 4 scale points higher than the average). Control, the feeling that the software is responding in an expected and consistent way, was considered average. The majority of users considered *Squirrel* to be no more efficient than other comparable software systems (Efficiency was rated 4 scale points below the average). Users found the response to inputs, that is, the time to display a set of results, too slow.

14.6. Conclusions

The overall aim of the BT case study was to investigate how the semantic technologies researched and developed within the SEKT project could enhance the functionality of BT's digital library and address some of the challenges outlined at the beginning of this chapter. SEKT technology was used to integrate web content into our digital library prototype system.

The systematic inspection and the *cognitive walkthrough* analysis were found to be good tests of usability, enabling us to identify and correct a number of fundamental usability deficiencies prior to the field test phase of the evaluation.

The response of users to the new functions of the *Squirrel* and *SEKTagent* applications, when compared with the current technology in use in BT's digital library, was positive. The findings of a task-based evaluation of *Squirrel* revealed some promising gains in the average perceived information quality (PIQ) of the search results when compared with the current search engine.

A SUMI assessment of *Squirrel* showed that users rate the application positively and believe that it has attractive properties, but were concerned about its performance and speed.

Overall, the BT case study, in particular the positive results of the user evaluation, has given us confidence that the semantic technologies researched and developed in the SEKT project can be used to good effect in the next generation of semantically enabled search and browse and search agent applications. Some of these functions are expected to be integrated into BT's digital library in the near future.

Acknowledgement This work was supported by the IST Programme of the European Community under SEKT (IST-IP-2003-506826).

References

Alsmeyer D, Owston F (1998) Collaboration in Information Space. In: Proceedings of Online Information 98. Learned Information Europe Ltd., 31–37.

Chen H (1999) Semantic Research for Digital Libraries, D-Lib Magazine 5 (10), http://www.dlib.org/dlib/october99/chen/10chen.html.

Davies J, Studer R, Warren P (eds) (2006) Semantic Web Technologies: Trends and Research in Ontology-based Systems. Wiley, New York ISBN: 0-470-02596-4.

Duke A, Glover T, Stincic S (2006) Search and Browse Facility – Final Version. SEKT project, deliverable D5.2.2.

Kirakowski J, Corbett M (1993) SUMI: The Software Usability Measurement Inventory, British Journal of Educational Technology 24 (3): 210–212.

Kiryakov A, Popov B, Ognyanoff D, Manov D, Kirilov A, Goranov M (2003) Semantic Annotation, Indexing, and Retrieval, In: 2nd International Semantic Web Conference (ISWC2003), 20–23 October 2003, FL, USA. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence) vol. 2870, Springer, Berlin Heidelberg, 484–499.

Lynch C, Garcia-Molina H (1995) Interoperability, Scaling and the Digital Libraries Research Agenda, A report on the May 18–19th 1995 IITA Digital Libraries Workshop, http://dbpubs.stanford.edu:8091/diglib/pub/reports/iitadlw/main.html.

Nielsen J, Molich R (1990) Heuristic evaluation of user interfaces. In: Chew C Whiteside J (eds) Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Empowering People (Seattle, Washington, United States, April 01– 05, 1990). J. CHI '90. ACM Press, New York, 249–256.

NSF (2003) Knowledge Lost in Information, Report of the NSF Workshop on Research Directions in Digital Libraries, June 15–17, 2003. http://www.sis.pitt.edu/~dlwkshop/report.pdf.

van Rijsbergen CJ (1980) Information Retrieval. Butterworths, London.

Wharton C, Bradford J, Jeffries R, Franzke M (1992) Applying cognitive walkthroughs to more complex user interfaces: Experiences, issues, and recommendations. In: Bauersfeld P, Bennett J, Lynch G (eds) Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Monterey, CA, United States, 03–07 May, 1992). CHI '92. ACM Press, New York, 381–388.

Chapter 15
Semantic Technology for Professional Judicial Knowledge

Pompeu Casanovas, Núria Casellas, Joan-Josep Vallbé, Marta Poblet, Jesús Contreras, Mercedes Blázquez, and V. Richard Benjamins

Abstract An iFAQ (intelligent Frequently Asked Questions) system designed to provide assistance to young judges at their first appointment is described. The development and use of legal domain ontologies is described. The functionality of the system is set out, along with the system architecture. The methodology and results of a system evaluation are also reported.

15.1 Introduction

Iuriservice is an iFAQ (intelligent Frequent Asked Questions) system designed to provide assistance to young judges at their first appointment. In previous works (Rodrigo et al. 2004, Casanovas et al. 2004) we described the context in which this system has been built, the state-of-the-art of legal ontology building, and the judicial scenario in which the system had to be implemented. In this chapter we will describe the technical improvements of the system, the ontologies built up using SEKT methodology, and the performance of different tests, both with legal experts and real users.

First, in Sect. 15.2, we will describe some contextual issues such as agreements with the Judicial School and the Spanish Judicial General Council, and the changes in the users' technological skills.

In Sect. 15.3, we describe the progress in legal ontology building within the Legal Case Study. First, the development and completion of six OPJK (Ontology of Professional Judicial Knowledge) sub-domains is outlined. Second, the completion and construction of QTO (Question Topic Ontology) and JTO (Judgments Topic Ontology) with the use of OntoGen v 2.0 is detailed.

P. Casanovas(✉)
Institute of Law and Technology (IDT), Autonomous University of Barcelona,
e-mail: casanovas@uab.es

In Sect. 15.4 we explain the functionalities of the system, provide a high-level overview of the architecture, and offer some initial analysis of the results.

Section 15.5 explains some methods and results of the Legal Case Study usability tests. This section is followed by some final remarks.

15.2 Contextual Issues and Legal Situation

15.2.1 Legal Agreements

Legal issues matter in judicial research. On December 15 2005 the Plenary of the Spanish Judicial Council (CGPJ) issued an Order approving the extension until 2007 of the agreement on research between the CGPJ and the Autonomous University of Barcelona (AUB) (January 31, 2001). The Order states explicitly that researchers are under the restrictions on sensitive data established by the Spanish legislation (and art. 6 of the original agreement, concerning the confidentiality of judicial data).[1]

In addition, there is an agreement of research signed by the AUB, iSOCO and La Ley-Actualidad S.A. (Wolters Kluwer) (2004). In virtue of this agreement SEKT researchers have had access to the 450,000 judgments contained in La Ley legal database.

15.2.2 Contextual Issues: The Internet and Judges' Technological Skills

Judges entering into the Judicial School have improved significantly their technological skills over a 3-year period. As shown in Casanovas et al. (2006) there has been a consistent growth of judges' declared computer skills, declared use of the Internet, and number of people possessing a personal computer. Compared to the 52nd class members, new judges of the 58th class have more computer skills (47% vs 73.2% of affirmative answers) and a personal computer at home (35% vs 82%). However, on their arrival, only half of the students were used to work with legal databases.

Although this might seem surprising, it is consistent with the general growth and development of the Internet and its uptake in Spain. One of the most striking features

[1] It is worth mentioning that on November 27th, Iuriservice was officially introduced in a formal session to the CGPJ representatives, the CENDOJ (Centro de Documentación Judicial) representatives, judges of the 58th class, and Directors of the Latin-American Judicial Schools (who were present at that time at the Spanish School). Iuriservice will be implemented at the Judicial School during 2007.

of this development is that the Internet access is since the beginning related to the accessibility from personal computers at home, rather than at the work places or at schools, universities or centres of study. Young judges do not constitute an exception to this collective pattern.

Therefore, in order to interpret the results shown by the accumulated data available, they should be compared with the general growth of the Internet use in Spain. The accumulated statistical results on the eight last classes at the Judicial School show a judge profile in which females are prominent (63.71% vs 36.29% of males), mostly situated in a segment of age between 26 and 30-years old (70.77%).[2]

When interpreting the declared computer skills of judges entering the Judicial School, the important issues to take into account are:

- Internet access takes place mostly from personal computers;
- The Internet main functional utilities are limited to: (1) search and browse (72%); (2) e-mail (68%) and (3) e-news (53%) (Ministerio de Industria, Turismo y Comercio 2006b);
- Judges have not received a specific computer training as law students at the Law School;
- During the 4 years (average) of preparation for the official examination [*concurso-oposición*] to become a judge, they did not need any computer skill either.

The data may help to understand the low rate of legal databases use among those who access at the beginning of the semester. But do notice at the same time that they are becoming more familiar with technology and prone to learn and use new computer tools. This means that a significant change has been produced since our last report on judicial recruitment in Spain (Poblet and Casanovas 2005).

15.3 Legal Case Study Ontologies

In this section we will outline the current development and improvements of OPJK, the Ontology of Professional Judicial Knowledge, constructed by legal experts from a corpus of questions concerning on-duty problems provided by the judges in an ethnographic fieldwork study (Casanovas et al. 2004, 2005). These improvements are the addition of new concepts and instances and the possibilities given by MORE's versioning management (Huang et al. 2006). Finally, we will show the improvements made in the two topic ontologies used by this case study: QTO and JTO.

[2] Even for this segment between 24 and 35 years old corresponding to a working age, the Spanish access to the Internet is mostly home (50.2%), instead of work place (47.3%) or the university (14.2%). Other studies stress that there is a high rate of Internet penetration (43% inhabitants) and a high rate of broadband penetration (11.7% inhabitants) in Spain. The growth is very fast: for the first 3 months of 2006 the home connection rate was 33.9%, 1.4 million more (in absolute numbers) than a year before (Ministerio de Industria, Turismo y Comercio 2006a).

15.3.1 OPJK

The Ontology of Professional Judicial Knowledge (OPJK) has been built manually from a corpus of ~800 questions gathered during an array of ethnographic surveys (Casanovas et al. 2007, Casellas et al. 2005). During these surveys young judges at their first appointment identified a list of questions regarding their practical problems.[3] Therefore, the modelling of the content of these questions is directed to the formalization of professional judicial knowledge (Casanovas et al. 2004). *Professional knowledge* has been defined as the knowledge possessed by professionals, which enables them to perform their work with quality (Eraut 1992). In this regard, the design of OPJK requires the representation of not only the legal theoretical knowledge but also the knowledge produced and passed on among the judicial profession (as a subclass of the legal profession).

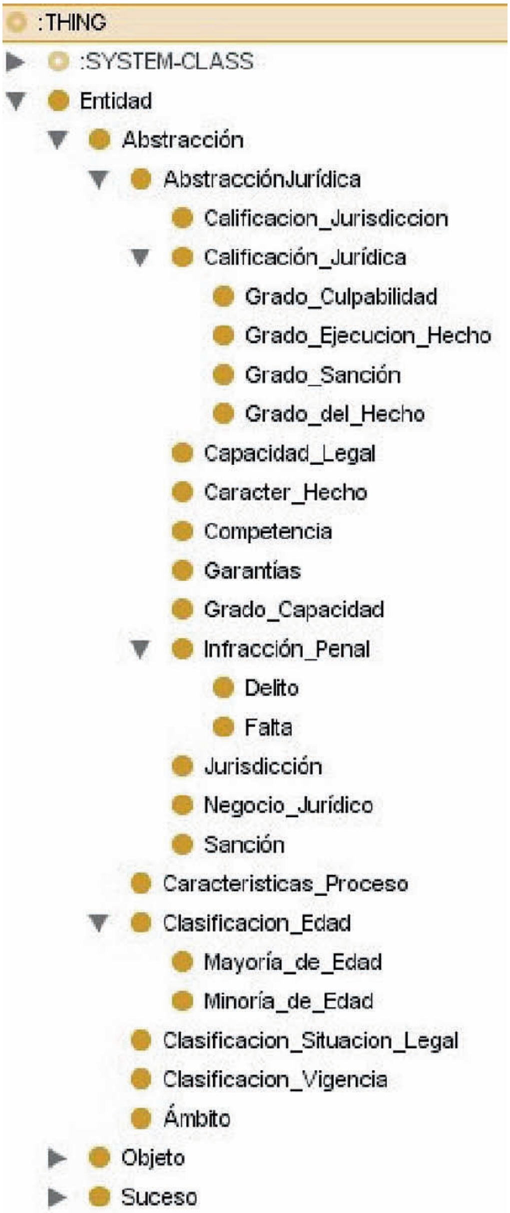
Furthermore, the questions were divided into different subdomains for their analysis. The on-duty, gender violence, marital issues, immigration and property domains have been analyzed and completed. Currently the ontology has 104 concepts and a total of 561 instances. More concepts and instances from the remaining questions regarding professional and procedural issues are yet to be formalized and included in the ontology. In addition to this formalization, also disjoint statements need to be identified for the detection of inconsistencies (Huang et al. 2006) (Fig. 15.1).[4]

During this manual formalization process different versions of the ontology were stored. As the concepts and instances of the ontology are being added taking into account the appearances in each frequently asked question, a new version of the ontology is created when the analysis of a question is completed.

This offered the possibility to use OPJK as test data to evaluate the system for ontology versioning and management (MORE). The MORE tool was developed in Amsterdam within SEKT, and described in detail in Huang and Stuckenschmidt (2005). It offers semantic versioning support for ontology development based on a combination of change detection and Linear Temporal Logic. Thus, the OPJK versioning space was built as a linear space with usually one new version per FAQ analyzed. In these tests, OPJK showed more stability with time, the addition of new individuals occurred more often than the addition of new concepts, the

[3] For the manual construction of the ontology the DILIGENT methodology was followed (Casanovas et al. 2007). Also, to aid in the construction of the judicial professional ontology, collaborative work was directed to the automatic or semiautomatic extraction of significant concepts from the corpus of questions containing judicial problems. At first TextToOnto was used for this, and subsequently Text2Onto (Völker et al. 2005).

[4] The OWL/DIG conversion and disjoint axiom enhancement could improve ontology versioning evaluation and management, a task initiated with the collaboration of the team from the Vrije Universiteit of Amsterdam (Huang et al. 2006).



- :THING
  - ▶ :SYSTEM-CLASS
  - ▼ Entidad
    - ▼ Abstracción
      - ▼ AbstracciónJurídica
        - Calificacion_Jurisdiccion
        - ▼ Calificación_Jurídica
          - Grado_Culpabilidad
          - Grado_Ejecucion_Hecho
          - Grado_Sanción
          - Grado_del_Hecho
        - Capacidad_Legal
        - Caracter_Hecho
        - Competencia
        - Garantías
        - Grado_Capacidad
        - ▼ Infracción_Penal
          - Delito
          - Falta
        - Jurisdicción
        - Negocio_Jurídico
        - Sanción
      - Caracteristicas_Proceso
      - ▼ Clasificacion_Edad
        - Mayoría_de_Edad
        - Minoría_de_Edad
      - Clasificacion_Situacion_Legal
      - Clasificacion_Vigencia
      - Ámbito
    - ▶ Objeto
    - ▶ Suceso

Fig. 15.1 Screenshot of current OPJK

data also showed that most of the changes so far were are small and, finally, that if the amount of instability inflicted by the addition or modification of subclasses could be assessed, different modelling options could be sought to promote stability (Huang et al. 2006).

15.3.2 QTO

We have used OntoGen to develop a topic ontology from the pre-lemmatized corpus of questions (Question Topic Ontology, QTO) as described in Blázquez et al. (2005) and also in Casellas et al. (2006). OntoGen is a "semi-automatic" and "data-driven" system for the construction of topic ontologies.[5] Topic ontologies consist of a set of topics (or concepts) and a set of relations between the topics which best describe the corpus. Ontogen discovers possible concepts and relations form the corpus.

An improved version of this tool has been released recently, OntoGen v2.0. This version of the tool is based on a novel ontology learning framework constructed especially for data-driven learning systems. The framework gives a basic ontology definition and defines concept and relation learning processes specially adjusted to include machine learning algorithms and takes into account the feedback received from the SEKT Legal and BT Digital Library case studies regarding OntoGen v1.0.

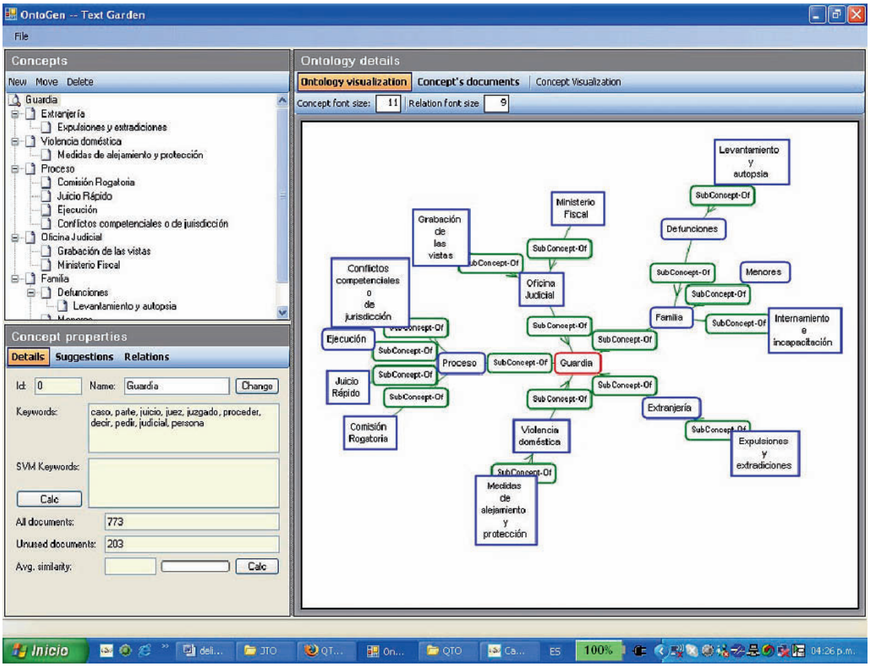
These new features are related to: (1) concept learning (the new version has a more extended list of keywords describing the suggested concepts and the user can also suggest concepts), (2) concept management (simplification of concept movements and pruning features), (3) ontology management (supports the addition of new documents into the already built ontology), (4) extended list of supported ontology formats (RDFS & OWL) and (5) an improved user interface (Fortuna et al. 2006).

In this case, a first QTO was developed using the first version of the tool and, after the public release, QTO was refined, developed and finished. The first topics identified were: *Oficina Judicial* [court office] with two subtopics (trial videotaping and prosecution), *Defunciones* [deaths] with two subtopics (corpse removing and autopsy), and *Violencia Doméstica* [gender violence] which has also two topics (protection orders and restraining orders) (Fig. 15.2).

It is important to note that the topic *Guardia* [on-duty] was considered to be the root of all the identified topics; on-duty was not considered anymore a topic but the moment in time when all the judicial questions were raised. Also the topics *Extranjería* [immigration], *Proceso* [process], *Internamientos* [internments] and *Familia* (menores) [family (minors)] were identified but were not developed further at the time.

For the development of QTO, the topic *Proceso* had to be subdivided due to its importance (quantity of questions contained). Also, it was important to be able to

[5] "**Semi-Automatic:** The system is an interactive tool that aids the user during the ontology construction process. The system suggests concepts, relations and their names, automatically assigns instances to concepts and provides a good overview of the ontology to the user through concept browsing and visualization. At the same time the user can fully adjust all the properties of the ontology by manually adding or deleting concepts, relations and reassigning instances. **Data-Driven:**Most of the aid provided by the system (concept, relation suggestion, etc.) is based on some underlying data provided by the user at the beginning of the ontology construction. The data reflects the domain for which the user is building ontology. Instance and instance co-occurrences are extracted from the data together with their profiles. Representation of profiles will be discussed later" (Fortuna et al. 2006).



- Guardia
  - Extranjería
    - Expulsiones y extradiciones
  - Violencia doméstica
    - Medidas de alejamiento y protección
  - Proceso
    - Comisión Rogatoria
    - Juicio Rápido
    - Ejecución
    - Conflictos competenciales o de juisdicción
  - Oficina Judicial
    - Grabación de las vistas
    - Ministerio Fiscal
  - Familia
    - Defunciones
      - Levantamiento y autopsia

Id: 0 Name: Guardia

Keywords: caso, parte, juicio, juez, juzgado, proceder, decir, pedir, judicial, persona

SVM Keywords:

All documents: 773

Unused documents: 203

Avg. similarity:

Concept font size: 11 Relation font size 9

Fig. 15.2 Screenshot of current QTO

differentiate and decide the boundaries of the topic *Familia*. In that development several sub-topics of *Proceso* were identified and also it was decided that Minors was a sub-topic of *Familia* that also contained the *Defunciones* sub-topic (usually related with inheritance issues).

15.3.3 JTO

Two topic ontologies have been created using OntoGen. The second topic ontology, the Judgment Topic Ontology (JTO) classifies in topics a concatenation of all the summaries that correspond to a particular topic (Blázquez et al. 2005, Casellas et al. 2006). A complete explanation of the methodology followed in order to collect and prepare the data for the construction of the topic ontology for the judgments is detailed in Blázquez et al. (2005). As a summary, 350,000 judgments were extracted from La Ley-Wolters Kluwer database (with their permission).[6] Due to the vast

[6] Official agreement: UAB-La Ley Actualidad, SA (Wolters Kluwer): Convenio marco de colaboración entre La Ley Actualidad, Universitat Autònoma de Barcelona y Intelligent Software Components (ISOCO). Madrid, December 21, 2004.

amount of data it was time-consuming to work directly on the whole document. Finally, after several analyses, it was decided that in order to produce the topic ontology for judgments the summary of the judgments was the best part of the structure to be explored.

The hand-made summary of a judgment provided a certain amount of information about that judgment. Although those summaries had some mistakes (misspelling, etc.), they contained "topics" describing what the judgment was about. Generally, they contained a "primary topic" which identified the major "topic" of the judgment and a "secondary topic" that made the summary more specific or refined the "primary topic". For that reason, only the primary topic was extracted and used for the construction of the JTO ontology. Also, generally, given that a number of summaries should belong to the same topic for that topic to be well-defined, we could only afford to extract frequently occurring topics. Thus, the Judgment Topic Ontology has been constructed from the document containing frequently occurring primary topics within the 350,000 judgments. There resulted six main topics: *Privado* [Private], *Procesal* [Procedural], *Fiscal* [Tax], *Social-Laboral* [Social-labour], *Contencioso-Administrativo* [Administrative or Public] and *Penal* [Criminal]. Furthermore, these main topics were themselves subdivided and unwanted contents were moved. Finally, QTO and JTO were aligned using OntoMap (Weiten et al. 2005).

15.4 Architecture

In this section, we briefly explain the functionalities of the system, provide a high-level overview of the architecture, and provide some initial analysis of the results.

15.4.1 Main Functionalities

The system can be best understood as an extended answer retrieval platform that allows users – judges in our case – to type a query in natural language, and to retrieve the most similar answered questions that best match the question input by the judge. The extension concerns two features: on the one hand, the usage of legal ontologies to perform semantic similarity matching and, on the other hand, the answer explanation system that offers complementary documentation (e.g., judgments) to support or expand the answer provided by the system. The key differential aspect of this system is its knowledge about the judicial professional domain, expressed as judicial domain ontology. Rather than matching, based on keywords, our system uses ontologies to both retrieve the most similar question and to provide the link to additional documentation. Figure 15.3 illustrates those two modes; on the left hand side there is the FAQ part (Expert System), while on the right hand side there is the Answer Explanation (Judgment) functionality. As it can be seen, the "answer explanation" part can also be used as a semantic meta-search engine over distributed legal sources.

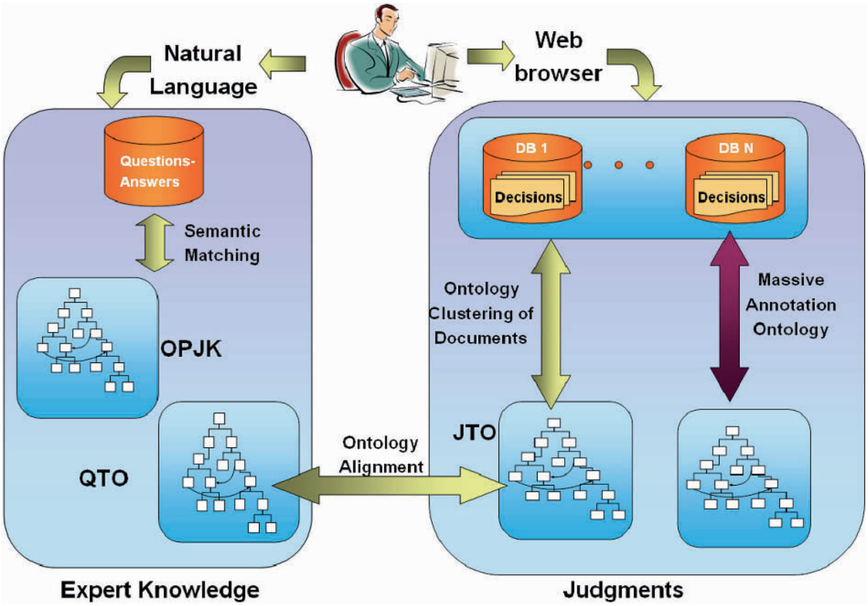


Fig. 15.3 High-level architecture of Iuriservice system. A FAQ system is combined with an answer explanation system that provides explanations for the answers provided by the FAQ part

15.4.2 Conceptual Architecture

In this section, we will provide an overview of the architectures of the two parts of the system.

15.4.2.1 FAQ System

Several search and score algorithms have been designed based on Natural Language Processing (NLP) and on Ontology Concepts Matching (Zhu et al. 2002). Algorithms have been organized around an architecture based on an adaptive multistage search chain, which is based on a variation of the "chain of responsibility"[7] software design pattern. In particular it is based on another pattern, called factory pattern that produces, on demand, a suitable search engine. This engine uses some search stage engine plug-ins and adapters to leverage on the main technologies used like NLP processing adapters, Ontology access and algorithms adapters. Each stage behaves

[7] "Avoid coupling the sender of a request to its receiver by giving more than one object a chance to handle the request. Chain the receiving objects and pass the request along the chain until an object handles it" (Gamma et al. 1995).

independently from previous stages. The stage starts with a FAQ subset as an entry, the goal being to reduce this subset with the constraint that the searched FAQ belongs to it. We have considered a three-stage search process, linking one outcome with the next entry, like a chain of responsibility. The *first stage* determines the domain of the question such as gender violence, criminal law, etc. The *second stage* uses keyword-based techniques to filter out FAQs that are dealing with other domains than that of the question. In the *last stage* the semantic distance is determined between the user question and the remaining FAQs. Since this is computationally an expensive process, it will be performed with those stored FAQs whose likelihood of appropriateness is above a certain threshold. Figure 15.4 illustrates this architecture. See Casanovas et al. (2005) for details.

The main technologies used in this architecture are: (1) *Natural Language Processing* (used at several search stages to get additional comprehension from the user's question; morphological and syntactical analysis of the user's question are also performed and the relevant words and grammatical patterns drawn from the question are used by other components in further stages); (2) *Thesaurus Processing* (used to match words based on synonymous relationships, attempting both exact and synonym matching); (3) *Ontology Processing* (the system uses several legal domain ontologies to obtain understanding of the user's question, trying to, first classify the question into a sub-domain [QTO ontology] and, also trying to find a match between fragments of the user's question and paths in the OPJK ontology,[8] see Fig. 15.4); and (4) *Cache proxy* (intermediate results of repetitive calculations that can be saved to avoid the repetition of computations, many of these calculations can also be recovered from a repository like a RDBMS and saved on cached memory).

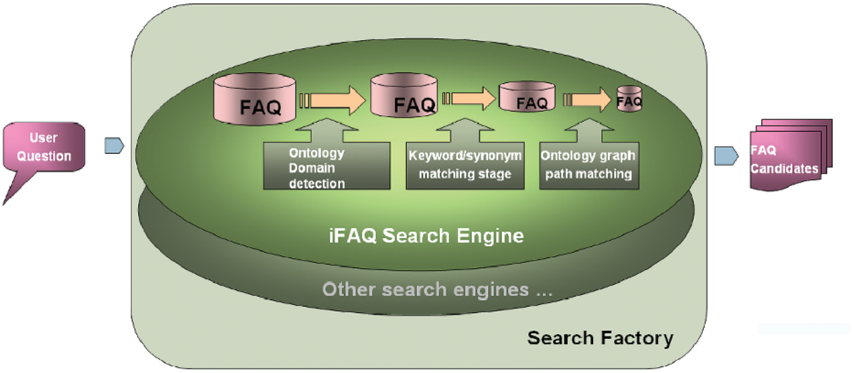


Fig. 15.4 Architecture of the Iuriservice FAQ subsystem

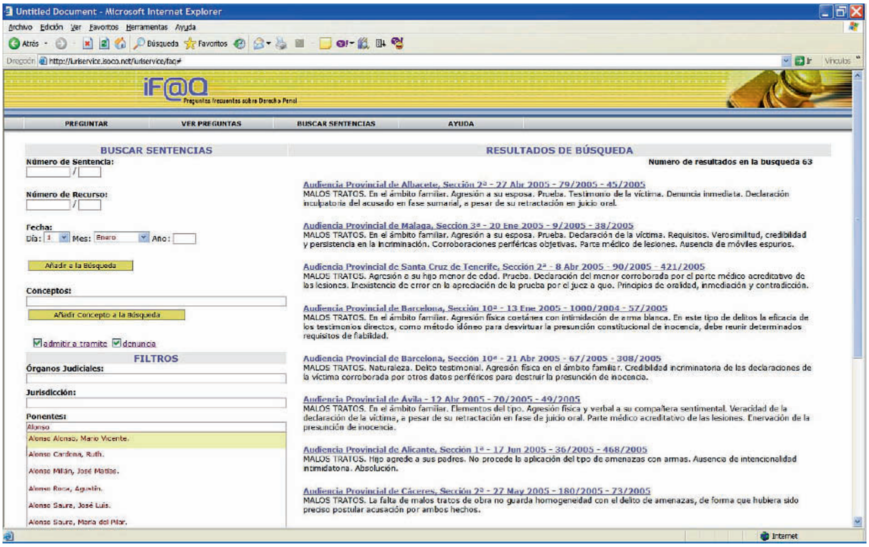
[8] It builds a graph path that is compared to each of the stored FAQ graph paths. Then the "semantic distance" between a new user query and the stored questions is calculated. Figure 4 illustrates the process of how two ontology fragments are matched to each other.

15.4.2.2 Answer Explanation System

In the Answer Explanation part of the system, the user can ask for supporting documents for the (or any) answer offered by the system. In this stage, the semantic search engine navigates the case law databases and offers references to relevant documents. This functionality allows the judge to learn more information from cases related to the answer provided. This functionality can also be used directly, as a meta-search engine for case law decisions.

These features are possible with automatic document processing and understanding based on ontology mapping and alignment technology. In this way each format representing a judgment is translated into a common schema and processed to establish links to stored FAQ answers. Besides the standard text interface, we are currently studying other more intuitive ways to visualise the results. Figure 15.5 shows the Use Case for this subsystem, including a meta-search engine that accesses the case law databases and extracts the relevant information and/or documents, and constructs the "explanation". The explanation consists of a set of automatically inserted hyperlinks into the question-answer pairs that point to relevant documents from the case law databases.

In order to connect the two kinds of judicial knowledge contained in the two subsections of the system, the knowledge from judicial experience (FAQ system) and the knowledge from judicial case-law (Answer Explanation), and to detect relevant documents (judgments, decisions) to justify the answers in the FAQ repository, the concepts in the two main ontologies had to be aligned. Therefore, if a user selects a justification, the system will check the OPJK concepts appearing in the


Untitled Document - Microsoft Internet Explorer
Archivo Edición Ver Favoritos Herramientas Ayuda
Atrás Búsqueda Favoritos
Dirección http://iuriservice.isoco.net/iuriservice/faq#
Ir Vínculos
iF@Q
Preguntas frecuentes sobre Derecho Penal
PREGUNTAR
VER PREGUNTAS
BUSCAR SENTENCIAS
AYUDA
Añadir a la Búsqueda
Añadir Concepto a la Búsqueda
Internet

BUSCAR SENTENCIAS

Número de Sentencia: /

Número de Recurso: /

Fecha:
Día: 1 Mes: Enero Año:

Conceptos:

☑ admitir a tramite ☑ denuncia

FILTROS

Órganos Judiciales:

Jurisdicción:

Ponentes:
Alonso
Alonso Alonso, Mario Vicente.
Alonso Cardona, Ruth.
Alonso Milán, José Matías.
Alonso Roca, Agustín.
Alonso Saura, José Luis.
Alonso Saura, María del Pilar.

RESULTADOS DE BÚSQUEDA

Numero de resultados en la busqueda 63

Audiencia Provincial de Albacete, Sección 2ª - 27 Abr 2005 - 79/2005 - 45/2005
MALOS TRATOS. En el ámbito familiar. Agresión a su esposa. Prueba. Testimonio de la víctima. Denuncia inmediata. Declaración inculpatoria del acusado en fase sumarial, a pesar de su retractación en juicio oral.

Audiencia Provincial de Malaga, Seccion 3ª - 20 Ene 2005 - 9/2005 - 38/2005
MALOS TRATOS. En el ámbito familiar. Agresión a su esposa. Prueba. Declaración de la víctima. Requisitos. Verosimilitud, credibilidad y persistencia en la incriminación. Corroboraciones periféricas objetivas. Parte médico de lesiones. Ausencia de móviles espurios.

Audiencia Provincial de Santa Cruz de Tenerife, Sección 2ª - 8 Abr 2005 - 90/2005 - 421/2005
MALOS TRATOS. Agresión a su hijo menor de edad. Prueba. Declaración del menor corroborada por el parte médico acreditativo de las lesiones. Inexistencia de error en la apreciación de la prueba por el juez a quo. Principios de oralidad, inmediación y contradicción.

Audiencia Provincial de Barcelona, Sección 10ª - 13 Ene 2005 - 1000/2004 - 57/2005
MALOS TRATOS. En el ámbito familiar. Agresión física coetánea con intimidación de arma blanca. En este tipo de delitos la eficacia de los testimonios directos, como método idóneo para desvirtuar la presunción constitucional de inocencia, debe reunir determinados requisitos de fiabilidad.

Audiencia Provincial de Barcelona, Sección 10ª - 21 Abr 2005 - 67/2005 - 308/2005
MALOS TRATOS. Naturaleza. Delito testimonial. Agresión física en el ámbito familiar. Credibilidad incriminatoria de las declaraciones de la víctima corroborada por otros datos periféricos para destruir la presunción de inocencia.

Audiencia Provincial de Ávila - 12 Abr 2005 - 70/2005 - 49/2005
MALOS TRATOS. En el ámbito familiar. Elementos del tipo. Agresión física y verbal a su compañera sentimental. Veracidad de la declaración de la víctima, a pesar de su retractación en fase de juicio oral. Parte médico acreditativo de las lesiones. Enervación de la presunción de inocencia.

Audiencia Provincial de Alicante, Sección 1ª - 17 Jun 2005 - 36/2005 - 468/2005
MALOS TRATOS. Hijo agrede a sus padres. No procede la aplicación del tipo de amenazas con armas. Ausencia de intencionalidad intimidatoria. Absolución.

Audiencia Provincial de Cáceres, Sección 2ª - 27 May 2005 - 180/2005 - 73/2005
MALOS TRATOS. La falta de malos tratos de obra no guarda homogeneidad con el delito de amenazas, de forma que hubiera sido preciso postular acusación por ambos hechos.

Fig. 15.5 Example for the "Answer Explanation" subsystem

answer, will transform them into the corresponding set of case law ontological concepts, and eventually retrieve the appropriate cases containing those concepts.

15.4.3 Test Results

At the current stage of the project, we have implemented the FAQ subsystem that uses the notion of semantic distance to calculate the similarity between user questions and stored FAQs (question-answer pairs). For this subsystem, we have performed tests to measure the effectiveness of the retrieval process. The goal of the measurement has been to verify whether retrieval, based on semantic distance technology obtains better results than traditional keyword-based retrieval. To focus the benchmark on the contribution of the "semantic" part, rather than comparing the results to simple keyword based approaches, we have compared it to an enhanced keyword approach by including a morphological analysis and synonyms. The test was performed over a corpus of more than 100 questions (Table 15.1).

15.5 Usability Tests

System evaluation with final users was made possible by the helpful intervention of the Magistrates from the School. In this way, ten trainee judges – nine the first day – (plus one School student) went to the School from the Courts of Rubí, Santa Coloma, Terrassa, Arenys de Mar, Tarragona and Barcelona to test the system. It is worthwhile to note that all of them made a real effort to come after their working day.

The objective of the SEKT case studies is to demonstrate that semantic technology provides benefits for users and organizations that operate information services for knowledge workers. As stated, users were involved in the entire development cycle of the Iuriservice application. In the early phases the needs of users were investigated, and the results were fed into the development process. During the actual development different approaches were used during the development phases, including the inspection of the prototypes by experts with standardised methods, and cognitive walkthroughs with early prototypes. In this way it was assured that the prototypes delivered to users are free of defects and correspond to the needs and requirements of users.

Table 15.1 Summary of tests performed

	Keyword	Keywords and semantic distance
Success	31, 25%	63, 23%
Failure	68, 75%	36, 77%

The completed prototype of the Iuriservice application was used to develop a knowledge base with information covering several legal domains. These systems were then used to carry out tests with a sample of end users.

A full analysis of the data is presented in Bösser et al. (2007), where the results of all user validation procedures carried out in the final phase of the SEKT project are reported.

The field tests of the Iuriservice application were designed to provide a test as realistic as possible of the application by judges and legal experts. As stated before, the user group of main interest for the study is young judges in their first appointment. These judges have a high workload, and they often require urgent support outside normal hours of work. They have access to legal information services (such as La Ley, Aranzadi and El Derecho), but would benefit greatly from more efficient support.

Tests were prepared for a group of judges, who participated in a test lasting for 2 days, and a group of legal experts. On the first day a number of typical cases were presented to the judges, and they were asked to solve the cases in a randomized order, using their usual tools, and their normal working procedures. These data served as a baseline for the tests on the second day: Firstly an introduction to Iuriservice was given, and then some exercises were carried out by all participants to familiarize them with Iuriservice. The subjects had all tools that they use in their work, and in addition Iuriservice available on their PC. After that the participants solved a number of cases from the same sample as those used on the first day, in randomized order again (such that each person solved all cases, half with and half without the use of Iuriservice).One of these cases is as follows:

I have given a restraining order to a woman and a few days after she wants me to withdraw the order. What should I do? Should I always withdraw it? What advice may I give to this woman?

The subjects recorded the time taken to solve the task and the number of queries executed. The decisions made were noted (and were assessed by independent experts). After the completion of all test cases, the subjects were asked to rate Iuriservice, and to complete the SUMI questionnaire.[9]

The same procedure was carried out with legal experts, but with a smaller number of cases.

As regards the results,[10] the comparison of the time taken for the same tasks either with or without the use of Iuriservice shows a significant difference. All tasks are solved in a much shorter time with the use of Iuriservice. This result may not be fully generalizable in quantitative terms because the limited number of judges available for the test did not allow the introduction of a further control group. However, as we discuss elsewhere, the results indicate strongly that Iuriservice provides an extra information service, complementing the existing databases, which allows the users to find the required information quickly and easily in many situations.

[9] Software Usability Measurement Inventory (SUMI). See http://sumi.ucc.ie/

[10] See Bösser et al. (2007) for further details details.

All subjects rate the Iuriservice application as positive and helpful, and see it as a desirable system. A more detailed analysis is available elsewhere, together with further analysis.

The summary of results of the SUMI questionnaire (Software Usability Measurement Inventory) shows that subjects assess the Iuriservice application as highly positive.

The measures are scaled to a mean of 50, and a standard deviation of 10. The five measures are defined as follows:

- Efficiency refers to the user's feeling that the software enables them to perform the task(s) in a quick, effective and economical manner.
- Affect is a psychological term for emotions. It refers to the positive user feeling of the user being mentally stimulated and pleased as a result of interacting with the software.
- Helpfulness refers to the user's perceptions that the software communicates in a helpful way and assists in the resolution of operational problems.
- Control refers to the feeling that the software is responding in an expected and consistent way to input and commands.
- Learnability refers to the feeling that the user has that it is relatively straightforward to become familiar with the software.

In summary the results demonstrate that the subjects were able to use Iuriservice without any apparent problems. It was easy to learn, and the application was rated as highly positive. The users expect considerable gains in the efficiency of their work, and reduction of their workload from an introduction of Iuriservice into their working environment.

15.6 Final Remarks

In this chapter, we described the context in which Iuriservice has been built: the judicial scenario in which the system is to be implemented, the state-of-the-art of legal ontology building, the technical improvements of the system, the ontologies built up using SEKT methodology, and the performance of different tests, both with legal experts and real users.

We may draw several final remarks from the work performed. In the first place, Iuriservice is now at the first stages of implementation at the Spanish Judicial School, therefore user needs compliance and positive attitude and feedback are of high importance. The agreements with the General Council of the Judiciary and the collaboration of the Spanish Judicial School demonstrate their interests in the development and implementation of Iuriservice.

Secondly, both the architecture and the ontologies were significantly improved. OPJK represents, now, six judicial sub-domains (Family, Procedure, etc.) and QTO and JTO have been completed.

In the third place, regarding the user tests, their preliminary results are promising and give us additional reasons to be positive about the system performance. Both young judges and young lawyers liked the system and their functionalities.

Concerning architecture, the performance of the FAQ System has been focused on improving ontology and the heuristic associated to find the best match between a user input question and the questions stored and the improvement of the average response time. The Answer Explanation System is still under development.

Acknowledgement This work was supported by the IST Programme of the European Community under SEKT (IST-IP-2003-506826).

References

Blázquez M, Peña-Ortiz R, Contreras J, Benjamins V R, Casanovas P, Vallbé J J, Casellas N (2005) D10.3.1 Prototype, EU-IST Project IST-2003-506826 SEKT Report.

Bösser T, Casanovas P, Casellas N, Melchior E M, Thurlow I, Vallbé J J (2007) D8.4.1 Results of User Tests and Completed Use Case Studies. Technical Report. EU-IST Project IST-2003-506826 SEKT.

Casanovas P, Poblet M, Casellas N, Vallbé J J, Ramos F, Benjamins V R, Blázquez M, Rodrigo L, Contreras J, Gorroñogoitia Cruz J (2004) D10.2.1 Legal Scenario. Case Study-Intelligent Integrated Decision Support for Legal Professionals. EU-IST Project IST-2003-506826 SEKT.

Casanovas P, Poblet M, Casellas N, Contreras J, Benjamins V R, Blázquez M (2005) Supporting newly-appointed judges: a legal knowledge management case study. Journal of Knowledge Management 9(5): 7–27.

Casanovas P, Vallbé J J, Casellas N, Poblet M, Blázquez M, Contreras J, Benjamins V R, López-Cobo J M (2006) D10.4.1 Legal Case Study After Analysis. EU-IST Project IST-2003-506826 SEKT Report.

Casanovas P, Casellas N, Tempich C, Vrandečić D, Benjamins V R (2007) OPJK and DILIGENT: ontology modeling in a distributed environment. Artificial Intelligence and Law 15 10.1007/s10506-007-9036-2. Springer, Berlin.

Casellas N, Blázquez M, Kiryakov A, Casanovas P, Poblet M, Benjamins V R (2005) OPJK into PROTON: legal domain integration into an upper-level ontology. WORM: 3rd International Workshop on Regulatory Ontologies, Part of the International Federated Conferences (OTM '05). 31 Oct–4 Nov 2005, Larnaca, Cyprus. http://www.cs.rmit.edu.au/fedconf/.

Casellas N, Jakulin A, Vallbé J J, Casanovas P (2006) Acquiring an ontology from the text. Ali M, Dapoigny R (eds.) Advances in Applied Artificial Intelligence, 19th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (IEA/AIE 2006) (Annecy, France, June 27–30, 2006), Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence) vol. 4031, Springer, Berlin Heidelberg, 1000–1013.

Eraut M (1992) Developing the knowledge base: a process perspective on professional education. Barnett, R. (Ed.) Learning to Effect. Open University Press, Buckingham, 98–118.

Fortuna B, Grobelnik M, Mladenić D (2006) Semi-automatic Data-driven Ontology Construction System. Proceedings of the 9th International Multi-conference Information Society IS-2006, Ljubljana, Slovenia.

Gamma E, Helm R, Johnson R, Vlissides J (1995) Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading, MA.

Huang Z, Stuckenschmidt H (2005) D 3.5.1 Reasoning with multiversion ontologies. EU-IST Project IST-2003-506826 SEKT Project Report.

Huang Z, Schlobach S, van Harmelen F, Casellas N, Casanovas P (2006) D3.5.2. Reasoning with Multi-version Ontologies: Evaluation, EU-IST Project IST-2003-506826 SEKT Project Report.

Ministerio de Industria, Turismo y Comercio (2006a) Red.es. XI Oleada: Las TIC en los Hogares Españoles. Estudio de la demanda de servicios de telecomunicaciones y sociedad de la información que se ofertan al segmento residencial en España. Enero-Marzo 2006. http://www.red.es [December 11th, 2006] [Public Report].

Ministerio de Industria, Turismo y Comercio (2006b) Red.es. Las TIC en los Hogares Españoles. Perfil Socio-demográfico de los Internautas. Actividades realizadas en Internet III Trimestre 2003-III Trimestre 2005. Observatorio de las Telecomunicaciones y de la Sociedad de la Información. Mayo 2006. http://www.red.es [December 11th, 2006] [Public Report].

Poblet M, Casanovas P (2005) Recruitment, Professional Evaluation and Career of Judges and Prosecutors in Spain. Di Federico G (ed.) Recruitment, Professional Evaluation and Career of Judges and Prosecutors in Europe: Austria, France, Germany, Italy, The Nederlands and Spain. IRSIG, Bologna, Ed. Lo Scarabeo, 185–214.

Rodrigo L, Blázquez M, Casanovas P, Poblet M (2004) D10.1.1. Legal Case Study Before Analysis, EU-IST Project IST-2003-506826. SEKT Report.

Völker J, Vrandečić D, Sure Y (2005) D3.3.2. Data Driven Change Discovery – Evaluation. Project IST-2003-506826 SEKT Project Report.

Weiten M, Wenke D, Meier-Collin M (2005) D4.5.3. Prototype of the Ontology Mediation Software V1. Project IST-2003-506826 SEKT Project Report.

Zhu H et al. (2002) An approach for semantic search by matching RDF graphs. Special Track on Semantic Web at the 15th International Flairs Conference (AAAI), May 2002, FL.

Chapter 16
Semantic Web for Knowledge Reuse in Business Processes

Jasmin Franz and Ralph Traphöner

Abstract This chapter describes the case study performed by empolis together with Siemens Business Services, whereas empolis provided the technological solution and Siemens Business Services acted as user partner. The users should deliver requirements to the technical work packages to stimulate their thinking about how semantic technology should be developed to enhance the benefit of user applications. Another task was to test out the technology being developed. In particular, test out if and how the various technologies can be integrated. The final objective was to gain feedback from user trials, both regarding the functionality proposed and also regarding the modes of user interaction.

16.1 Starting Position and Objectives

The overall goal of the specific Siemens Business Services case study was to optimize the analysed knowledge flows between Siemens employees, customers and partners which mean to make the contribution, relocation and reuse of information, documents and knowledge easier.

As a service organization employing more than 30,000 people, Siemens Business Services is in charge of knowledge resources to cover solutions, technologies, companies, industries and markets – knowledge that is increasing daily through projects, advanced training and the recruitment of new employees.

A fully implemented knowledge management system means that Siemens Business Services employees and local teams are not perpetually being forced to reinvent the wheel. Instead they can rely on approved knowledge by methods (Davenport and Probst 2002) and tested solutions as well as derived from project experience and expert contacts, all of which allows them to work more quickly and

J. Franz (✉)
Empolis GmbH, Europaallee 10, Germany
e-mail: jasmin.franz@empolis.com

J. Davies et al. (eds.), *Semantic Knowledge Management*,

more cost-effectively. For these purposes knowledgemotion®[1] has been implemented which forms a basis for effective knowledge sharing within Siemens Business Services. This browser-based portal represents the 'umbrella' of the overall KM solution and provides a single interface for connecting to the global knowledge network of Siemens Business Services. As the central point of contact, it ensures knowledge is accessible worldwide.

The main challenge of the SEKT project on behalf of Siemens Business Services was the development of superior new functionality which creates considerable benefit for the user and business process effectiveness (focussing sales & delivery) by means of higher quality and more efficient access to information on the basis of knowledgemotion®.

The requirements capture activity within this case study was done by a focus group to study user requirements in depth. It was apparent that for the IT Consultant in focus of the study there is a need for improved searching and retrieving, and that lack of time was a disincentive to knowledge sharing. There was also a general concern about the difficulty people found in retrieving knowledge during certain process steps having a specific role.

Another challenge of Knowledge Management per se is the difficulty of measuring the business impact. Therefore the selected business cases have also been seen as a test field to measure the impact both in employees acceptance and management key performance indicators. While the most important key figure will be time saved, it is important to show also what will be the additional business benefit based on the time savings (Nonaka 1991).

As the objective of the Siemens Business Services case study is to investigate and verify how semantically enabled technologies can improve the productivity of IT and business consultants, the scenarios focus on how to stimulate the emergence and creation of new knowledge and its capture along the Proposal- and Project Management Process in the Consulting Department.

16.2 The Adoption of Semantic Web Technology

We learned that the time spent either on searching or on storing information is too long and this is reflected by the users as basis of more barriers. Day-to-day operations along with high EBIT targets and constant changes are often raised as objections. The biggest hurdle to overcome with semantic enabled knowledge management is the effort to spend on classical attribution or search via metadata. Both normally require additional work from the IT consultant what in many cases people are not willing to invest.

The figure above (Fig. 16.1) shows the expected benefit when using semantic technologies. Especially the effort for the so called "Reinventing" phase and the "Searching for knowledge & information" could be decreased significantly, while concurrently there should be a gain of time for the essential tasks of innovation and

[1] Note: knowledgemotion® is a world-wide registered trademark of Siemens Business Services GmbH & Co. OHG.

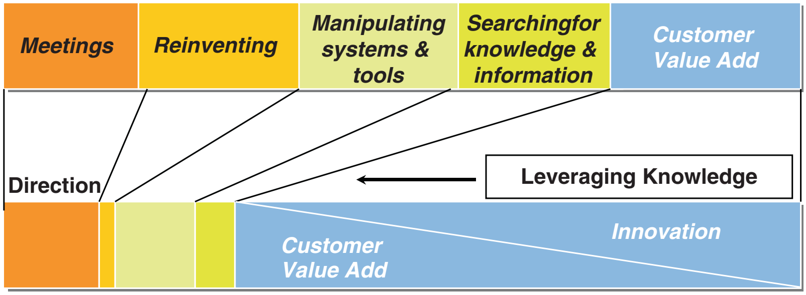


Fig. 16.1 Expected benefit by use of semantic enabled technologies

generating real customer value (Nonaka and Takeuchi 1995). Compared with the time spent searching for scattered information and knowledge sources and the frequent frustration at the lack of success, the time it takes to post a knowledge asset or respond to an urgent request is relatively small. The ability to reuse documents and access a large personal network, however, actually saves time and reduces operational pressure.

For efficient work knowledge management systems should enable codified knowledge reuse in consultants' business tasks everyday. Required are methods and tools that do not only provide access to knowledge repositories, but almost access to knowledge experience from projects. For this purpose the need for ontologies is stressed having implemented metadata. Thus the setup of ontology based knowledge management systems is recommended strongly.

Knowledge-based systems are "Meta-models" of expertise consisting of concepts in terms of human expertise. They are described by ontologies and implemented declaratively. The task ontology is needed to make knowledge-based systems aware of what task they are performing. Therefore specific conceptualization for the knowledge has to be defined before its use in the knowledge-based system.

An authoring system dealing with a tutoring task ontology knows what a tutoring task is and knows what type of domain knowledge is necessary to perform the task, which enables the authoring system to behave intelligently in the authoring support process.

For specific search by explorative navigation an IT Consultant is able to retrieve appropriate knowledge in which the content is localized by performing a sequence of selection steps within an organized content repository using semantically based retrieval algorithms.

Referring to knowledge discovery in corporate intranets and knowledge management systems, the context-awareness of a search engine may be defined as to include for example the current role of an employee performing a specific task in a specific process. Additionally it may also take into account the personal profile of this person including previous searches, manually set preferences and access-rights. To control the scope of the search the search engine offers options to apply search filters based for example on metadata attributes contained in the source documents of the different data pools (Siemens Business Services Core-Metadata Set). SEKT technology enables the enrichment of metadata by relations, additional concepts, automatically extracted from the text sources, such as named entities for example.

Therefore the process of information search from a user perspective is focused in detail in Use Case 1, taking especially into account the opportunities given by the assumed availability of an ontology based semantic search technology.

Covering the process steps of adding new content to the Knowledge Base and answering the request of automatic support for assigning metadata is subject matter within Use Case 2.

Reuse of existing knowledge within an IT Service Organization for providing competitive offerings in order to generate more business while achieving their productivity targets is discussed in the next chapter within Use Case 3 as forecast of further development (Reuse Initiative).

16.3 The SEKT Use Cases

Based on the analysis carried out, scenarios and use cases were developed on how work could actually be changed by applying the envisaged SEKT technologies.

16.3.1 Context of the Scenarios

The retrieval and content-creation process (Von Krogh 1998) is a specific process that turns a local document into a corporate knowledge asset (candidate) and regulates the worldwide exchange of knowledge. It ensures that the contents of the knowledge base are up to date and of a high quality.

When a new business problem arises for example a new task has to be faced by the IT Consultant and an appropriate process model has to be conceptualized, retrieved and instantiated. The execution of a new instance embodies the reuse of the process knowledge as well as of the knowledge contained in the attached information.

16.3.2 Use Case 1: Searching for Information

Standard "full text" and advanced searches do not meet user expectation, therefore it is necessary to also provide information based on the push principle (based on role, customer prospect, portfolio, industry, etc.) or do implicitly include those framework information into the search. Not all information will always be able to be gathered in internal sources; therefore an extension to external sources is necessary as well.

16.3.2.1 Analysis Profile: Searching for Information

Table 16.1 Description of use case 1

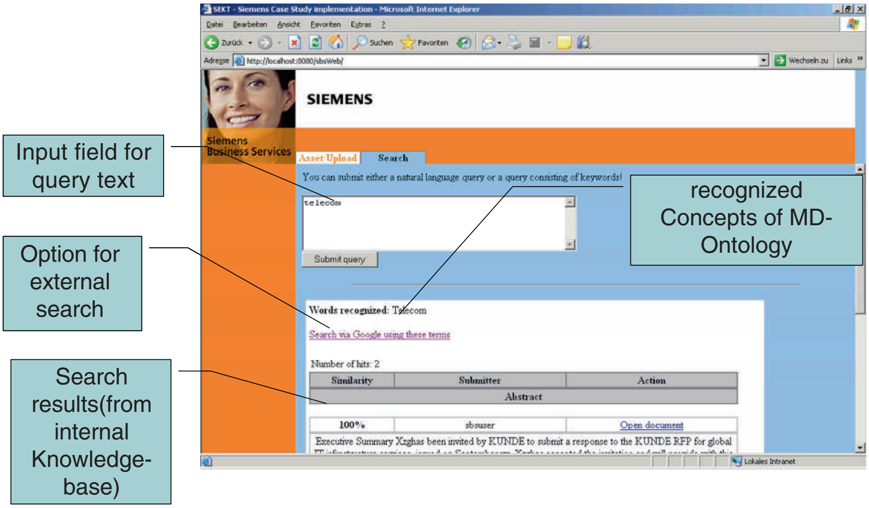
Use Case ID	UC1
Short – description:	To write a proposal a huge set of information is necessary, such as: templates methods knowledge about customer knowledge about market knowledge about competitors knowledge about solutions/products knowledge about own delivery competencies
Roles (involved):	Sales manager, project manager, project team members
Compelling event:	Process proposal development
Data:	See above
Result:	Proposal ready for proposal delivery (transfer to process proposal delivery and follow-up
Pre-conditions:	–
Relevance:	Very important, very often
Expected improvement:	Improved quality of information retrieval in terms of recall and precision leading towards increased productivity that is reduced proposal preparation time. Queries can be saved which enables the system to provide the user with relevant new information (push service).

16.3.2.2 SEKT Case Study Prototype for Use Case 1

The empolis prototype allows a Siemens employee to perform a search in the internal Knowledge Base as well as in external sources. The search form provides the following facilities for searching and is available in an expert search interface:

1. The system offers the option to perform a natural language search: The user may enter a complete paragraph or a question which the engine then analyses for similarity of content elements and returns associative related documents. Even spelling mistakes are corrected by the system.
2. The system answers which terms it has recognized and displays all relevant search results. The result set is automatically sorted based on the statistical relevance (percentage of match) of the retrieved documents.
3. Another option will allow the user to advise the engine to find "similar documents" related to a single document in the result set. This similarity is also based on associative, statistical relationships. The system also provides the option to search those terms externally, via Google.

4. To control the scope of the search the search engine takes the user profile into account.
5. To view one retrieved document the user has to click on the link displayed in the column Action. The document will be displayed and the user will see all recognized concepts marked up in the file (Fig. 16.2).


SEKT - Siemens Case Study implementation - Microsoft Internet Explorer
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück
Suchen
Favoriten
Adresse http://localhost:8080/sbsWeb/
Wechseln zu
Links
SIEMENS
Siemens Business Services
Asset Upload
Search
Submit query
Lokales Intranet
Input field for query text
Option for external search
Search results(from internal Knowledge-base)
recognized Concepts of MD-Ontology

You can submit either a natural language query or a query consisting of keywords!

telecom

Words recognized: Telecom

Search via Google using these terms

Number of hits: 2

Similarity	Submitter	Action
Abstract		
100%	sbsuser	Open document
Executive Summary Xzghas been invited by KUNDE to submit a response to the KUNDE RFP for global		

Fig. 16.2 Use Case 1, search results

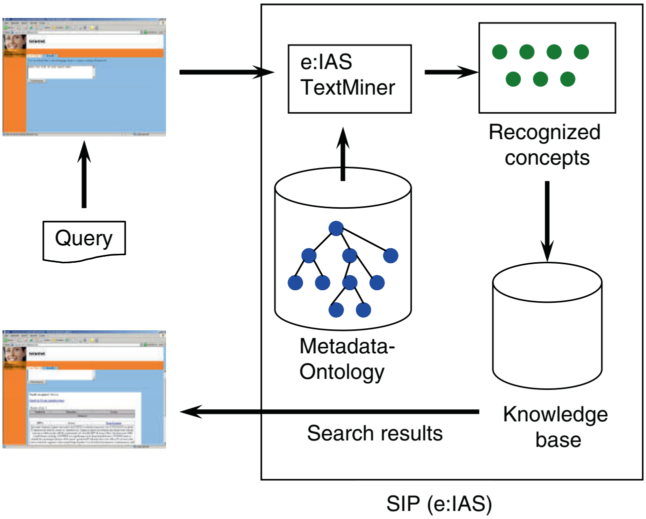


Fig. 16.3 Use Case 1, search process

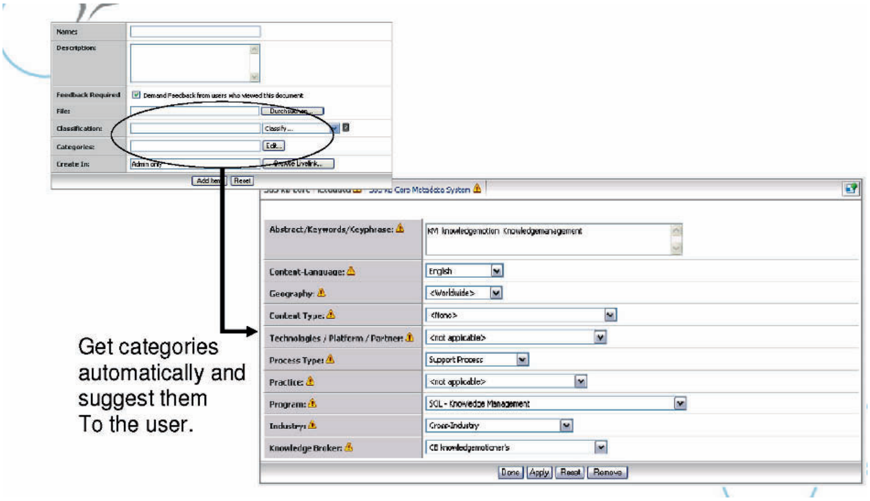
Figure 16.3 shows situation behind the scenes while performing a search. The text of the query is analysed by using e:IAS[2] TextMiner and the Metadata Ontology (Siemens Business Services Core-Metadata Set). Using all recognized concepts the engine queries the knowledge base and fetches matching items. The matching items are returned as the search result.

16.3.3 Use Case 2: Contribution of Knowledge Assets

The upload process described was mainly based on manual steps and contributions of additional information provided by the users. In many discussions we heard that this additional effort is considered as being too high.

The idea behind Use Case 2 is to automatically identify and gather all the semantic Meta information coming along with the document (e.g., type or format) as well as the role and the context of the individual user.

Most of this information describes the specific situation in which a user may want to upload a document and therefore it should be provided to the user by the system as pre-defined/proposed attributes that the user only has to confirm (see Fig. 16.4).



Name:	
Description:	
Feedback Required	☑ Demand Feedback from users who viewed this document
File:	
Classification:	Classify ...
Categories:	
Create In:	Admin only

Abstract/Keywords/Keyphrase:	KM knowledgemotion Knowledgemanagement
Content-Language:	English
Geography:	<Worldwide>
Content Type:	<None>
Technologies / Platform / Partner:	<not applicable>
Process Type:	Support Process
Practice:	<not applicable>
Program:	SCL - Knowledge Management
Industry:	Cross-Industry
Knowledge Broker:	CB knowledgemotioner's

Get categories automatically and suggest them To the user.

Fig. 16.4 Use Case 2, metadata suggestion

[2] e:IAS, empolis Information Access Suite is a product of empolis GmbH.

16.3.3.1 UC2 Analysis Profile: Contribution of Knowledge Assets Candidates

Table 16.2 Description of use case 2

Use Case ID	UC2
Short – description:	A user would like to upload a specific document/knowledge asset candidate to the knowledge base and therefore has to attach additional information to the document.
Roles (involved):	User as sales manager/proposal manager, project manager, project team member
Compelling event:	Perform knowledge asset upload
Data:	Role Situational context Personal profile (history of behaviour, project context, etc.), document
Result:	Document with added metadata, based on pre-configured/proposed metadata (attributes)
Pre-conditions:	–
Relevance:	Very important, very often
Expected improvement:	Metadata will be collected according to an improved siemens metadata core set definition. Applying SEKT and in particular Information Extraction and knowledge discovery capabilities must lead to an improved metadata quality in terms of richness and completeness when compared to purely manual annotation.

16.3.3.2 SEKT Case Study Prototype for Use Case 2

This use case covers the process steps of adding new content to the Knowledge Base and answering the request of automatic support for assigning metadata. Again the figure demonstrates the technical process behind when uploading a document to find relevant metadata (Fig. 16.5):

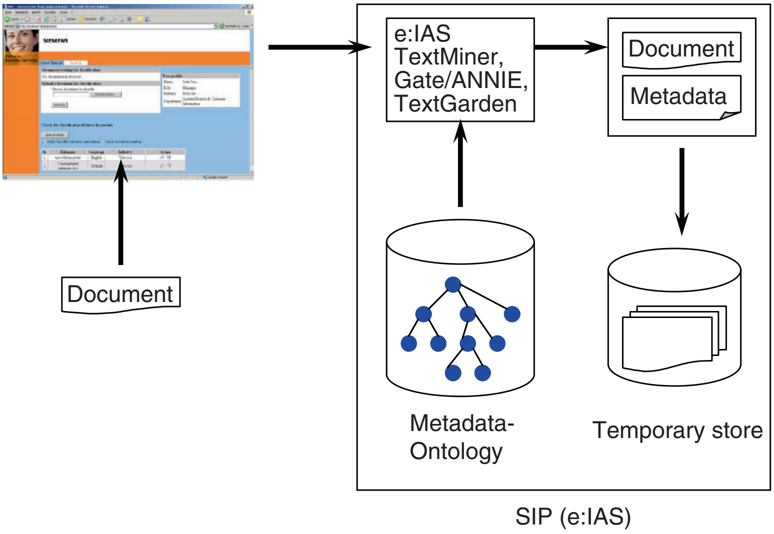
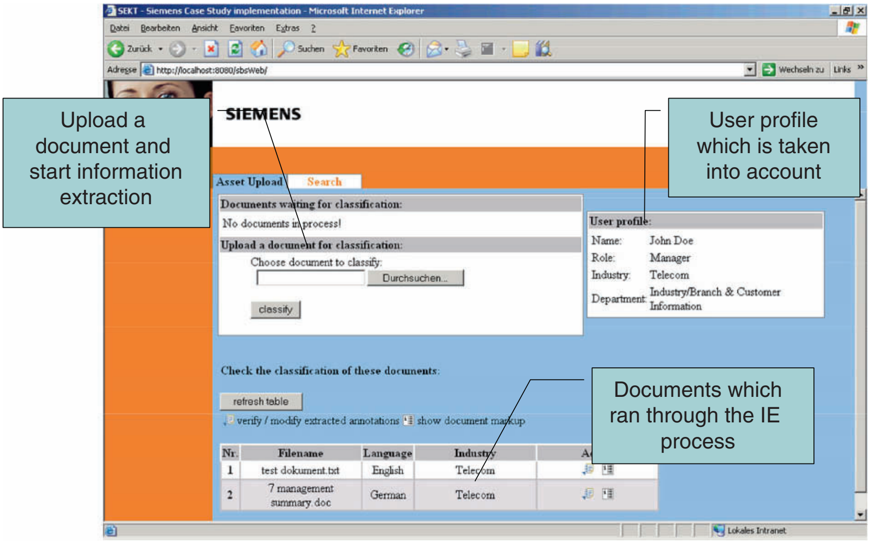


Fig. 16.5 Use Case 2, asset upload

After a document has been uploaded to the system it is analysed by Gate/ANNIE, TextGarden, and the e:IAS TextMiner. During the analysis phase the textual content of the uploaded document is used to extract relevant information based on the metadata ontology. Recognized concepts are extracted and added to the corresponding metadata. Finally, the extracted metadata and the document are stored in a temporary store.

The empolis prototype enables a Siemens employee to select a document for being analysed according to the SBS Metadata Scheme.

1. Once users have chosen a document for being classified, the system analyses the document against the metadata ontology and the specific user profile. The system returns a table with the checked document by displaying its defined metadata like language and branch (Fig. 16.6).



Upload a document and start information extraction

User profile which is taken into account

Documents which ran through the IE process

http://localhost:8080/sbsWeb/

Documents waiting for classification:

No documents in process!

Upload a document for classification:

Choose document to classify:

User profile:

Name:	John Doe
Role:	Manager
Industry:	Telecom
Department:	Industry/Branch & Customer Information

Check the classification of these documents:

verify / modify extracted annotations show document markup

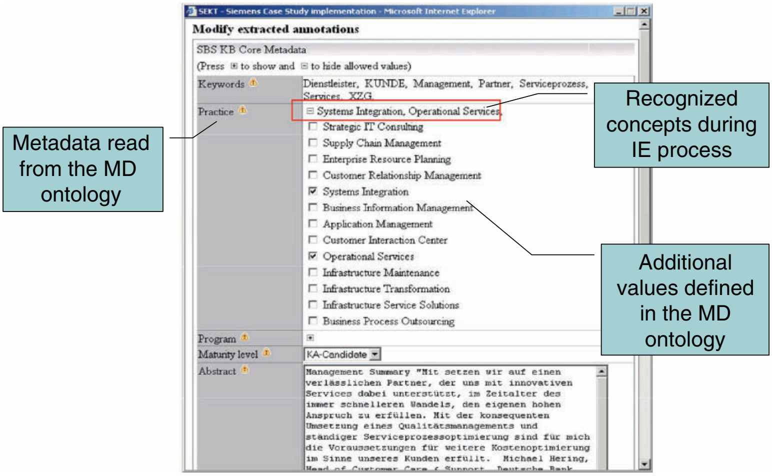
Nr.	Filename	Language	Industry	A
1	test dokument.txt	English	Telecom	
2	7 management summary.doc	German	Telecom	

Fig. 16.6 Use Case 2, asset upload procedure

Then the user has the choice to verify, modify or add particular attributes to the document on another screen:

2. By clicking on one of the Action buttons the user is requested to:
   a. Open the annotated document to view the markup of recognized concepts. The highlighted words made up the analysis result. Moving over a text fragment shows the corresponding type of this concept.
   b. Verify or modify the extracted annotations. The Siemens employee might want to check and modify the metadata extracted by the system. The user selects the attributes that best characterize his/her document just by checking the box right next to the value (Fig. 16.7).

To verify/complement the extracted metadata for a document the extracted metadata for this document is read from the temporary store and presented to the user (see screenshot). The user has the opportunity to add additional values from pick lists.


SEKT - Siemens Case Study implementation - Microsoft Internet Explorer

Modify extracted annotations

SBS KB Core Metadata

(Press ⊞ to show and ⊟ to hide allowed values)

Keywords: Dienstleister, KUNDE, Management, Partner, Serviceprozess, Services, XZG

Practice: ⊟ Systems Integration, Operational Services

- ☐ Strategic IT Consulting
- ☐ Supply Chain Management
- ☐ Enterprise Resource Planning
- ☐ Customer Relationship Management
- ☑ Systems Integration
- ☐ Business Information Management
- ☐ Application Management
- ☐ Customer Interaction Center
- ☑ Operational Services
- ☐ Infrastructure Maintenance
- ☐ Infrastructure Transformation
- ☐ Infrastructure Service Solutions
- ☐ Business Process Outsourcing

Program: ⊞

Maturity level: KA-Candidate

Abstract:

```
Management Summary "Mit setzen wir auf einen
verlässlichen Partner, der uns mit innovativen
Services dabei unterstützt, im Zeitalter des
immer schnelleren Wandels, den eigenen hohen
Anspruch zu erfüllen. Mit der konsequenten
Umsetzung eines Qualitätsmanagements und
ständiger Serviceprozessoptimierung sind für mich
die Voraussetzungen für weitere Kostenoptimierung
im Sinne unseres Kunden erfüllt.  Michael Hering,
Head of Customer Care & Support, Deutsche Bank
```

Metadata read from the MD ontology

Recognized concepts during IE process

Additional values defined in the MD ontology

Fig. 16.7 Use Case 2, metadata procedure

These pick lists are populated using the information from the metadata ontology. If the user decides to commit the document to the knowledge community the document and its metadata are inserted into the knowledge base and are removed from the temporary store. Referring to the SEKT technology, the system uses Text Garden and GATE components developed in SEKT Work Packages 1 and 2.

The document will now be in the responsibility of the community the user selected. If appropriate the relevant experts will contact the user about the quality of the document and its suitability as a knowledge asset. Of course the user may also contact the responsible community broker itself at any time.

16.4 Prospect of Further Development

The value of the organisational knowledge (Probst et al. 1999) is not based on its pure ownership, but on the multiplication of the knowledge (e.g., the re-use of knowledge). This chapter provides a perspective on a "third" use case (beyond the SEKT project) that focuses knowledge reuse by semantic means: the Reuse Initiative.

16.4.1 Reuse Initiative

Risk reduction occurs when Siemens Business Services can replicate project results and outstanding successes to as many customers and with as minimal effort as possible (reference selling) (Fig. 16.8).

Basis for replication is a completed project with the proven systems architecture of an implemented solution with as many stable, reusable knowledge assets and documented modules as possible.

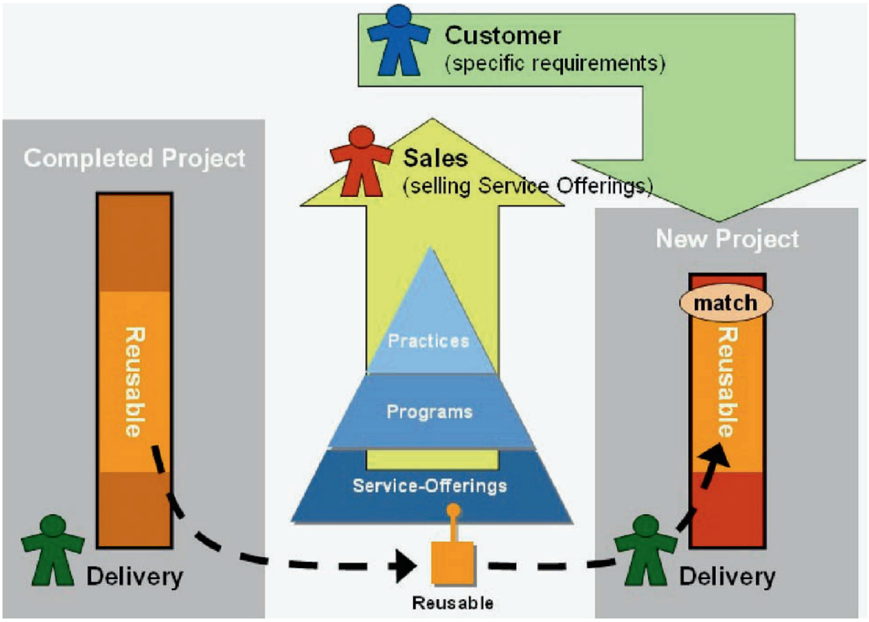


Fig. 16.8 Use Case 3, reuse course

16.4.1.1 Types of Reusable Knowledge Assets

Reusable Knowledge Assets originate out of a project. Their spectrum can vary from a single project element to pre-packaged projects. Accordingly the EBIT and sales impact of a Reusable Knowledge Asset differs:

- **Reusable Knowledge Assets of type A** enhance Siemens competence and capabilities and demonstrate this towards their customers, for example, through references. They are single project elements such as presentations, proposals, lessons learned, tool demos etc. Their direct EBIT impact is difficult to quantify.
- **Reusable Knowledge Assets of type B** are solution kernels (GUI, application modules), project approaches (business process models, architecture) and prepackaged projects (service offering on module level). As they have a direct measurable EBIT impact or reduce the delivery effort significantly, they improve Siemens internal performance (O’Dell and Grayson 1997).

16.4.1.2 Knowledge Transfer into a Project

Every new customer requirement, proposal and suggested solution architecture needs to be evaluated for the utilization of Reusable Knowledge Assets. This affects both proposal and delivery projects. All aspects of Reuse have to be planned by the

Sales, Proposal and Project Managers and Solution Architects at the very beginning of each opportunity/project (project ramp-up). Reusable Knowledge Assets are stored in the Knowledge Base and can be found for example via:

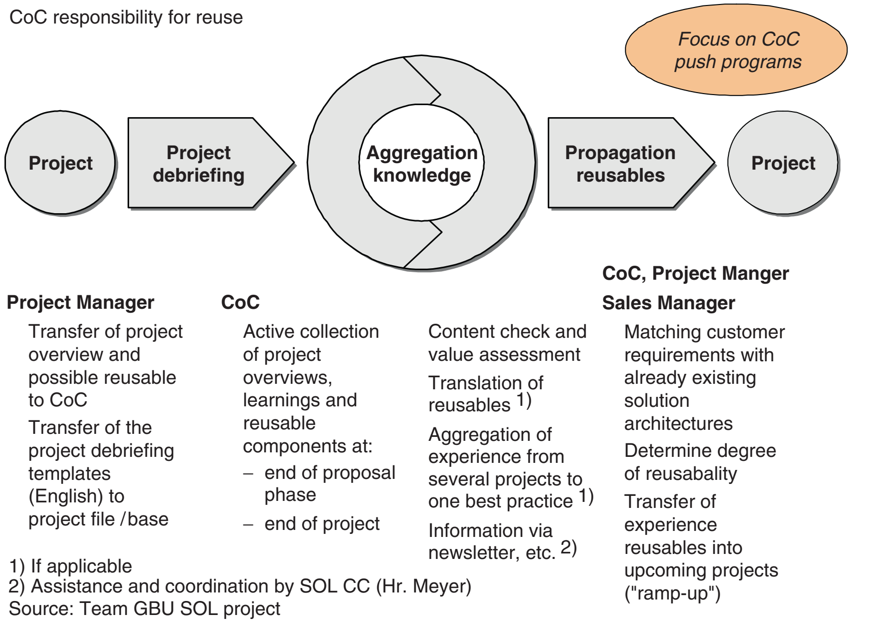
- a central find area
- the Reusable Catalogues for Reusable Components, Concepts and Solutions
- predefined queries or topic folders of the particular CoC
- the Cross CoC Search center

16.4.1.3 Knowledge Transfer Out of a Project

To ensure that knowledge is efficiently transferred out of a project (project wrap-up), two work products are mandatory:

- the Project Overview and
- the Project Debriefing Report

In the project wrap-up phase, new Reusable Knowledge Assets shall be identified and submitted to the Knowledge Base.



Project Manager

Transfer of project overview and possible reusable to CoC

Transfer of the project debriefing templates (English) to project file /base

CoC

Active collection of project overviews, learnings and reusable components at:
- end of proposal phase
- end of project

Content check and value assessment

Translation of reusables 1)

Aggregation of experience from several projects to one best practice 1)

Information via newsletter, etc. 2)

CoC, Project Manger
Sales Manager

Matching customer requirements with already existing solution architectures

Determine degree of reusabality

Transfer of experience reusables into upcoming projects ("ramp-up")

1) If applicable
2) Assistance and coordination by SOL CC (Hr. Meyer)
Source: Team GBU SOL project

Fig. 16.9 Use Case 3, CoC responsibility

16.4.1.4 Reuse in the Knowledge Management Process at the CoC

The CoC[3] is responsible for the collection, aggregation and propagation of Reusable Knowledge Assets (Fig. 16.9):

The person contributing the Reusable Knowledge Asset is responsible for its correctness. In general any employee of Siemens Business Services can identify Reusable Knowledge Assets and upload these to a CoC for Siemens Business Services-wide availability. For Reusable Knowledge Assets resulting from a project, the project manager takes responsibility for uploading the asset to the corresponding CoC community workspace.

16.4.1.5 UC3 Analysis Profile: Re-Use Initiative

Table 16.3 Description of use case 3

Use Case ID	UC3
Short – description:	Support for sales, proposal or project manager
Roles (involved):	Campaign manager Local unit manager SOL Segment manager/CoC segment responsibles Local sales manager/key sales people/multipliers Key project managers/team leader delivery/key systems architect Key communications people/multipliers
Compelling event:	Reuse of knowledge assets and reuse campaigns
Data:	Process input
Result:	When starting a proposal or a delivery project, the sales, proposal or project manager is responsible for considering and transferring reusable knowledge assets into the project. The corresponding CoC community broker shall assist upon his/ her request.
Pre-conditions:	Transfer of experience and applicable reusable knowledge assets into upcoming projects
Relevance:	Very important, very often

16.5 Conclusion

One of the goals of the SEKT project was the development of context-aware search engines. This new ontology based semantic search technology aims to significantly improve the success of searches and offers several opportunities to overcome the

[3] Center of Competence: organisational units that collect and disseminate knowledge and best practices within SBS.

current problems of mainly associative and statistical based information retrieval. The results of this case study arise from two perspectives. One is the subjective view of the user company (Siemens Business Services) itself, one is the objective view derived from the evaluation process performed by the methodological partner (Kea-Pro) of the SEKT project.

16.5.1 Benefits Seen by the Company

- **Lower costs:** Knowledge sharing generates savings not just by enabling knowledge to be reused and by its documented learning effects, but also by allowing new employees to be integrated more rapidly into the company and by its unified, integrated IT structure.
- **Time savings:** Time savings can be expected mostly through the avoidance of redundancies, fast access to information and knowledge, and rapid communication channels.
- **Improved quality:** The worldwide exchange of knowledge and experience improves the quality of service.
- **Increased sales and profits:** Knowledge sharing contributes to the enterprise's economic success. Problems can be solved more quickly and processes can be accelerated, which improves margins. Coming along with increased customer satisfaction is a better chance of winning follow-up business, for example through cross-selling.

16.5.2 Evaluation of the Application

The aim of user tests with the SBS SEKT application was to demonstrate the benefit created by SEKT in the form of improvements over the existing knowledgemotion® system. User tests were prepared by enabling a comparison of the existing "knowledgemotion®" versus the system enhanced with SEKT components (K verso S).

Tests were prepared by building an ontology for the S system, selecting and indexing a relevant dataset. After inspecting the system, a number of tasks were defined, taking account of the specifics of the installation, and additional data were uploaded.

In a second iteration, tests were carried out with the set of defined tasks, by three experts. Results of querying the knowledge base were: All queries which retrieved documents from the S set of information provided clearly shorter lists of results than the K system – higher "precision" as demanded by users in the user needs studies. It could be seen that this is due to the data model in S (ontology). For the user, clearly reduced user time results (fewer queries, reduced browsing in the results list), which, as we know from the user needs analysis, is a high priority for the users.

The support for preparing knowledge assets in a semi-automatic fashion is a SEKT specific functionality which is not available in the K system. It is assessed as highly positive (reducing annotation effort and enhancing the quality of metadata), but would require a realistic test context to be assessed quantitatively.

Tests were not continued further at this stage, because the organization was undergoing changes with consequences expected for the knowledge management organization and processes. Furthermore very large datasets would have to be integrated (and used) in order to make further tests meaningful.

We concluded from the expert evaluation that the SEKT application for SBS corresponded to the needs of users, as they were determined in the earlier SEKT user needs analysis.

Acknowledgements This work was supported by the IST Programme of the European Community under SEKT (IST-IP-2003-506826). The author wishes to acknowledge gratitude and appreciation to all the SEKT project partners, especially to Siemens Business Services, for their contribution during the development of various ideas and concepts presented in this chapter.

References

Davenport T, Probst G (eds.) (2002) Knowledge Management Case Book. Siemens Best Practises. Erlangen: Publicis MCD Verlag.

Nonaka I (1991) The knowledge-creating company. Harvard Business Review 69(6): 96–115.

Nonaka I, Takeuchi H (1995) The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation. New York, Oxford: Oxford University Press.

O’Dell C, Grayson C (1997) If Only We Knew What We Know: Identification and Transfer of Internal Best Practices, Houston: American Productivity and Quality Center (APQC).

Probst G, Raub S, Romhardt K (1999) Wissen managen. Wie Unternehmen ihre wertvollste Ressource optimal nutzen, 3. Auflage, Verlag Dr. Th. Gabler GmbH, Wiesbaden.

Von Krogh G (1998) Care in knowledge creation. California Management Review, 40(3).

Chapter 17
User Quality and Business Benefit of Semantic Knowledge Management Applications

Tom Bösser, Núria Casellas, Elke-Maria Melchior, Ian Thurlow, and Joan-Josep Vallbé

Abstract A novel user-centred approach to evaluation of semantic knowledge management applications is described. Application of the approach to two case studies – fully functional prototypes for digital library users, and support for legal decision making – is discussed. The results show that users appreciate the enhancement to their working environment provided by semantic technology, improving both the efficiency of work, and the quality of the result. Estimation of cost and benefits Indicates that semantic knowledge management systems can be highly cost-effective for large and medium-size organizations.

17.1 Introduction

Users were involved in the SEKT project in all phases, including the analysis of user needs and the tasks carried out in their context of work. Prototypes were tested with experts and users in various stages, including field tests with fully functional prototypes for digital library users, and support for legal decision making.

The results show that users appreciate the enhancement to their working environment provided by semantic technology, improving both the efficiency of work, and the quality of the result. Estimation of cost and benefit indicates that semantic knowledge management systems can be highly cost-effective for large and medium-size organizations. The decisive success factor is a high rate of usage and acceptance of the application, which can be expected to be influenced positively by the very positive response of users of SEKT in the field tests.

Technology per se is a necessary, but not a sufficient condition for the creation of value. In addition to the demonstration of technology it must be shown that benefit can be created in a realistic application context. The semantic knowledge technology components developed in the SEKT project were integrated into three prototype applications to show the viability of the technology, the usability of the

T. Bösser(✉)
Kea-pro GmbH, Tal, Switzerland
e-mail: tom@bosser.net

J. Davies et al. (eds.), *Semantic Knowledge Management*,

applications, and to demonstrate the benefits created by the technology for users and for the organizations which adopt semantic knowledge technology.

A user-centred development strategy was introduced into the project process from the start, and all teams involved were made familiar with the required techniques. The three main phases of user centred activities in the SEKT project were:

1. Analysis of user needs and requirements
2. Usability testing of prototypes
3. Field tests of fully functional application prototypes with prospective users under realistic conditions

17.2 Analysis of User Needs and Requirements

The objective of the user needs analyses in SEKT was to understand the functionality which users require, the preference of users for specific functional components, and to understand the requirements imposed by the specific context of work. One aspect studied was the value which users attach to the quality of information, defined in information retrieval terminology as recall versus precision as proposed by van Rijsbergen (1980), which they obtain when using the SEKT technology components.

The analysis of user needs and requirements was conducted early in the project in each of the case studies (described in this volume). Prospective users, knowledge management experts, and decision makers on various levels were involved. The methods to elicit user needs and requirements included site visits, focus-groups, discussions in design meetings, and questionnaires distributed to different user groups. In total, more than 200 users participated. While some of the user groups employ tools for information search which correspond to the current state of the art, some of the user groups so far do not use computerized knowledge management tools, but will be introduced to these over a period of years.

The needs and requirements resulting from the analysis were used by the development teams to update the design specifications and implemented in the applications. This is described in other chapters in this volume. Main results which influenced the application development, and which serve as criteria for assessing the applications and semantic knowledge technology in the field tests which conclude the project, are the following:

Most of the innovative functionality implemented (named entity recognition, topic navigation, search refinement, integration of web content and knowledge base content, and search agent functionality) was uniformly welcomed by users – confirming that the initial project concept is built on solid understanding of the needs of users. There were some interesting differences between groups:

> Use of *portable devices for knowledge access* (e.g., PDA): Although there was a general feeling that knowledge access at anytime anywhere can be an important feature in the future, a majority of users in all groups did not consider this important for their own work – which does not exclude the possibility that a minority of users value this feature highly.

Natural language access to knowledge repositories was not considered an important feature by the digital library and business information users, who currently use information systems, whereas the users for the legal system were more positive.

There was an indication that users did not like the idea of automation (e.g., regular background information search and automatic generation of user profiles), possibly reflecting a concern of losing control.

Recall and precision as qualities of the information retrieved was investigated in choice-based preference analyses in different user groups of all three case studies. High precision (relevance) and recall (completeness) of information was valued higher than new technical features by all groups of users. There was a clear preference for precision of information over recall (i.e., more information). This means that users do not want just more information, but would like to obtain help to select the right information for their problem. However, different groups can be identified: Both for the digital library users and the business information services two clusters can be distinguished, one with a less pronounced preference for precision than the other. The legal users considered precision essential.

Note that at an early stage of the development of the new applications it may be rather abstract to users what it will mean for their actual work.

There were a number of application specific requirements which the different user groups emphasized. Digital library users emphasized access to as many titles as possible as most important, while the business users considered support for annotation of knowledge assets with meta-data and support for the submission of knowledge objects as a priority.

17.3 Usability Tests During the Development Phase

In all case studies usability tests were carried out at intermediate development stages when functionality was sufficiently concrete for users to evaluate. Firstly heuristic evaluations were carried out by experts, using checklists and guidelines. Then design adjustments were made, and further user tests were carried out where needed. Standard methods were used, which are described by Bösser and Melchior (2002) or Rubin (2001).

17.4 User Tests with the Three Case Study Applications

The case study applications were becoming mature at different times, and the limited availability of professional users as subjects demanded that specific and efficient tests were carried out. User tests were scheduled when technical integration was completed, when experts had reviewed the systems and concluded that the status made the applications meaningful for users, including the availability of sufficiently large knowledge bases which made sense for users. This status was not achieved in time for the business application. The quantitative situation would have demanded that on the order of several hundred thousand documents would have to be selected, processed and integrated into a meaningful application. For administrativel reasons this was not possible.

The digital library application was populated with a selection of documents from a relevant technical domain. The IURISERVICE system was fed with a set of documents and the question-answer pairs for a legal domain defined in the area of family law. These systems allowed the definition of meaningful information search tasks in domains with which users were familiar.

17.5 Test Procedures in the Field Tests

The user tests were designed to assess the impact of semantic knowledge technology in knowledge management applications on the efficiency, effectiveness of task completion, on the satisfaction of users, and on the quality of the information obtained by users. Although different test procedures had to be designed for each case study, the same methods were used as much as possible. Essentially all tests were carried out in a task based manner: The principle was to demonstrate the functionality of the SEKT prototype to users, and then to ask them to carry out a number of tasks, exercising the relevant functionality of the prototypes.

For two of the case study applications (digital library application and IURISERVICE) extensive tests were carried out.

17.6 User Tests of the Digital Library Application

The SEKT digital library prototype is a new front-end for professionals searching for information in the extensive digital library which provides access to several million documents, comprising two components, the search-and-browse front-end, and the *SEKTagent*, which partly automates the search task.

The following functions enabled by SEKT were demonstrated to the users, practiced in examples, and then assessed by each subject:

1. named entity recognition in the new search and browse application,
2. navigation and browsing using the topic ontology in the new search and browse application compared to the use of controlled indexing terms for navigation and browsing using the current technology,
3. search refinement in both applications,
4. integration of Web content in the new search and browse application
5. the *SEKTagent* semantic search agent function (in comparison with the use of the current implementation using "information spaces").

The main question is whether the SEKT front-end provides decisive advantages for users, compared to the existing digital library system.

Two analyses were carried out, the first one assessing the SEKT functionality component by component, the second one to assess the efficiency of information search in terms of information quality with and without semantic technology.

The SEKT functionality components were new to the 20 users which participated in the study. Each user was tested individually in the two consecutive tests lasting for a total of 2 h.

In the first part of the test, components were introduced to the test user one by one. Each component was assessed by the user after they received a demonstration of the new functionality, and after they carried out predefined tasks with the new search-and-browse interface, and with the existing tools. Functionality was assessed by users according to the following criteria on a 4- or 5-value scale (the 4-value scale was used where we wanted to force the subjects to avoid a neutral statement):

- What is the impact which you expect the new functionality to have on your work?
- Do you expect to find information faster with the new search and browse application in comparison with the current search engine?
- Is it easier to complete the information search task with the new search and browse application, in comparison with the current search engine?
- Do you expect to find better quality information with the new, or with the current technology (where precision, i.e., the relevance of results was understood as the main measure of quality)?
- Does the new function offer an improvement over the functions available for solving your task in the current search engine?

17.7 Results: Evaluation of Semantic Technology Components

The results are summarized in a table where "++" denotes overall very positive, "+" positive, and "0" neutral responses (Table 17.1).

Table 17.1 Summary of assessments of semantic technology components by digital library users

The search process	Faster	Easier	Information quality is better	The function is an improvement
Named entity recognition	+	+	+	+
Topic navigation	+	+	+	++
Search refinement	+	+	++	+
Integration of web content	+	+	+	++
Search agent	0/+	0/+	+	+

Overall, there was a very positive response to the new functionality presented, there were no negative assessments. The functionalities for topic navigation, search refinement, and for searching related web content stand out by the very positive assessment by subjects.

17.8 Self-Paced Search-and-Browse Task: Efficiency of Search and Information Quality

In a second task, the twenty subjects completed two tasks from a set of six either with the new, or with the classis user interface, according to their own pace. The time taken to complete each task was recorded. In addition, after each search-and-browse cycle the subjects were asked to assess the quality of the results which they had obtained up to that point in their search process. A search cycle is defined as the formulation of a query and the review of results, that is, normally just before a further modified query is formulated. Information quality was rated on a 9-valued scale.

Dimensionality of information quality: The choice of the single dimension rating scale was supported by an independent study of search progress carried out with students in a variety of search tasks. In order to understand which factors users employ to assess information quality, 284 subjects completed a questionnaire with a total of 24 questions to evaluate information quality from different perspectives during a search-and-browse process. The subjects were visitors of a large digital library of a university, persons working in a call centre which provides technical support, and persons searching travel information.

The results did show that most of the variance was explained by a general quality factor (40%). The second important factor is "progress" (42% of the variance). A third factor tentatively named "quality of information presentation" accounted for 9% of the variance, but this factor is statistically not reliable with 284 subjects.

The data confirm that it makes sense to describe information quality in search processes by a limited number of factors, and allows us to configure effective tests, using the most reliable and valid items from the questionnaire.

17.9 Results: Time to Complete the Digital Library Search Task

The difference in time between the users' search tasks with the two search engines is not statistically significant ($n = 20$ users) (Table 17.2).

Table 17.2 Mean execution times for the digital library search task (SD – standard deviation), $n = 20$ subjects

Using SEKT search engine	Classic digital library search engine
Average: 16.7 min SD = 4.71	Average: 14.7–min SD = 4.83

17.10 Results: Quality of the Information Obtained

Users gave an average of 4.6 assessments per search task, that is, one about every 3 min. This means that a new search cycle (with a reformulated query) was carried out about every 3 min.

The average rating of Information Quality using the existing library system was 3.99 against an average rating of IQ of 4.47 using the semantically enabled search and browse application. For each subject the average value for information quality was compared for both experimental conditions (with SEKT or with the old search engine). The rating for information quality is significantly higher when the SEKT search engine was used for the task (sign test, two-sided, $p < 0.01$).

Progress in the search process: In a two-dimensional plot of information quality (IQ) versus search progress (time or successive queries) the more effective tool for information search should be visible as a steeper curve towards the result of the search process. We obtain a characteristic function for the search process by plotting successive IQ measures against progress. As anchor point we do not use the starting point, but the point in time when the subject terminates search (the "cut"). We consider this the right perspective to regard progress, because subjects define this point independently, while the starting point depends on the prior knowledge of the subject in the domain of search. We use a rating of IQ on a 9-valued scale by users in this study because professionals define the quality criteria entirely on their own.

The plot of IQ versus search cycle shows that with the SEKT search engine the IQ is rated higher throughout the successive search cycles. When comparing the IQ versus time, the data indicate that with the SEKT search engine information quality is rated higher early in the search process, and does not increase as steeply as without. This indicates that a main advantage of the SEKT tools is to provide high quality information for the user early and easily in the search process (Fig. 17.1).

The bubble plots show the frequency distributions for the information quality ratings (IQ) ratings versus the search cycle in a search process. Size of the bubble indicates the observed frequency of values for each IQ – search cycle pair. The solid lines are the regression lines calculated for the data in each graph, while the dotted lines represent the regression lines from the other graph. Quadratic trend components were calculated, but are small and not statistically significant.

17.11 User Perceived Quality of the SEKT Digital Library Search Engine

The widely used Software Usability Measurement Inventory, a standardized and validated instrument, was administered to each subject. SUMI (Kirakowski and Corbett 1993) gives a detailed view of the subjective assessment of the SEKT enabled search and browse application by five independent factors of user satisfaction:

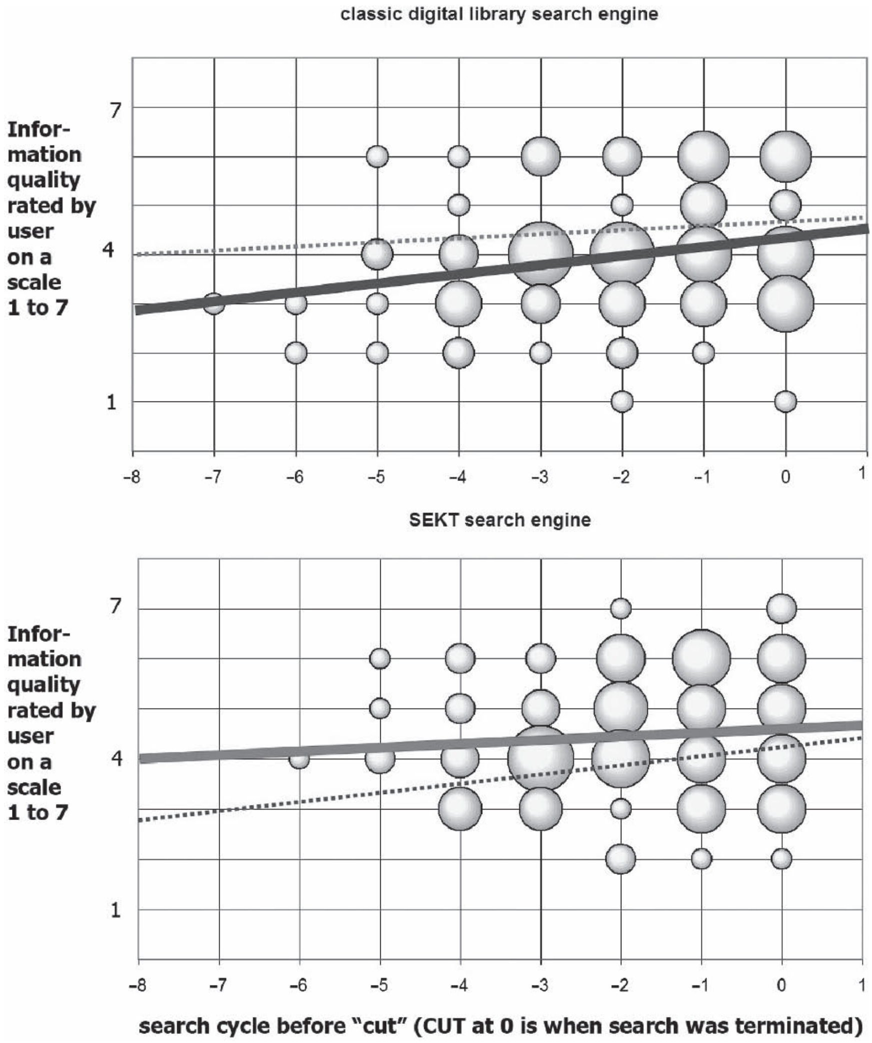


Fig. 17.1 Information quality ratings after search cycle before the termination of search. The size of circles indicates frequency

Efficiency *refers to the user's feeling that the software enables them to perform their tasks in a quick, effective and economical manner.*

Affect *refers to the positive user feeling of the user being mentally stimulated and pleased as a result of interacting with the software.*

Helpfulness *refers to the user's perceptions that the software communicates in a helpful way and assists in the resolution of operational problems.*

Control *refers to the feeling that the software responds in an expected and consistent way to input and commands.*

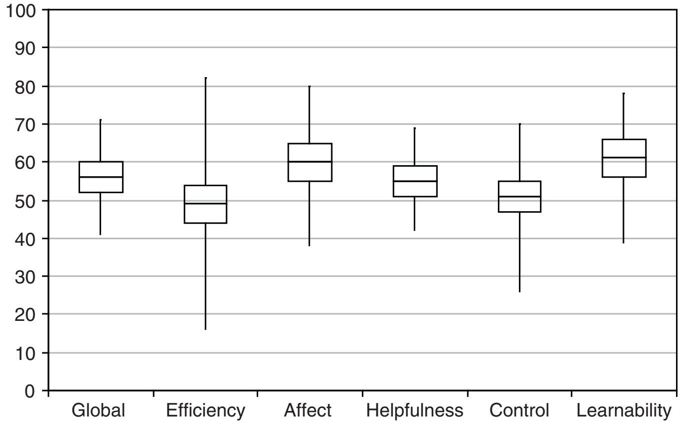


Fig. 17.2 Subjective assessment of the digital library search engine with the SUMI instrument. Mean of the standard scale is 50, plots show means, 95% confidence limits and range

Learnability *refers to the feeling that the user has that it is relatively straightforward to become familiar with the software.*

One subject did not complete part of the questionnaire. The results for 19 subjects are shown on the standardized scale with median, upper and lower 95% confidence intervals, and upper and lower limits of the distributions (Fig. 17.2).

Results show that the overall assessment of the SEKT search engine by users is positive. They find the application easy to learn, and see it as pleasing. The efficiency of work with SEKT, and the ability to exercise control over the application are rated as average.

17.12 User Tests of the IURISERVICE Application

The IURISERVICE application based on SEKT technology is intended to support judges with appropriate information while they work under time pressure. For validation of the application a number of workplaces were equipped with the IURISERVICE application, and a group of ten judges and a second group of ten legal experts from academia participated as subjects. The tasks were to elaborate immediate answers to legal questions from the domain of family law which are typical for those asked when judges are on 24 h duty, and are required to make urgent decisions. A written response has to be prepared by judges in a standard format. The tests consisted in completing a task, and subsequently answering a number of questions. From a set of six tasks the subjects completed three in random order on the first day, and three on the second day of testing.

On the first day baseline performance values were established: Judges solved three cases using their normal tools and procedures, including access to legal databases and the www. On the second day IURISERVICE was introduced by presenting examples, and subsequently the subjects worked through three further cases, now using IURISERVICE in addition to their usual databases. (It should be noted that a fully controlled experiment would have required a further control group, but this was not considered possible in a situation where a highly qualified group of professionals must be motivated to participate in a meaningful task.)

The data collected are performance (time to complete the task), assessment of the IURISERVICE functionality in a questionnaire, and the SUMI analysis of IURISERVICE. In addition, the quality of each solution – the legal decision of the judge – was assessed by an independent senior legal expert.

17.13 Results of the IURISERVICE Field Tests

Performance: Time to complete tasks: All subjects were able to solve all tasks. Two subjects participated on 1 day only. The time taken to provide a solution was significantly shorter in both groups (sign test, $p < 0.01$, two-sided test). The data are means for nine judges and nine legal experts under the condition without IURISERVICE (data missing from one subject in each group), and ten judges and ten legal experts with IURISERVICE (Table 17.3).

Quality of the solution: The quality of the solutions to all cases was assessed by a senior expert, in a manner comparable to career related assessments, according to the legal validity, the soundness of the decision, the richness of the argumentation, and the number of legal sources used. The data do not indicate any significant differences between the two test conditions. All subjects attain the same high level of quality in their decisions and legal argumentation.

Assessment of the IURISERVICE functionality by the subjects: Four important new technical features of IURISERVICE were rated by the subjects on a 5-valued scale.

Natural language queries, ordering of the results according to relevance to the question formulated in the query, clustering of results according to themes, and the suggestion of concepts to use for further refinement of the search were all rated as useful and helpful, most subjects used the most positive assessment (very helpful and useful) for the features.

Table 17.3 Mean times to solve one case without/with IURISERVICE

	Without IURISERVICE	With IURISERVICE
Judges	23 min per case	9 min per case
Legal experts	20 min per case	5 min per case

17.14 Assessment of the Quality of IURISERVICE with SUMI

The subjective assessment of IURISERVICE is very positive throughout, the highest scores are obtained for affect and efficiency. This means that users are satisfied with the application, and see IURISERVICE uniformly as positive (Figs. 17.3 and 17.4).

The additional comments received from subjects emphasize that specifically for the judges the improvement is significant and enables them to find answers to their

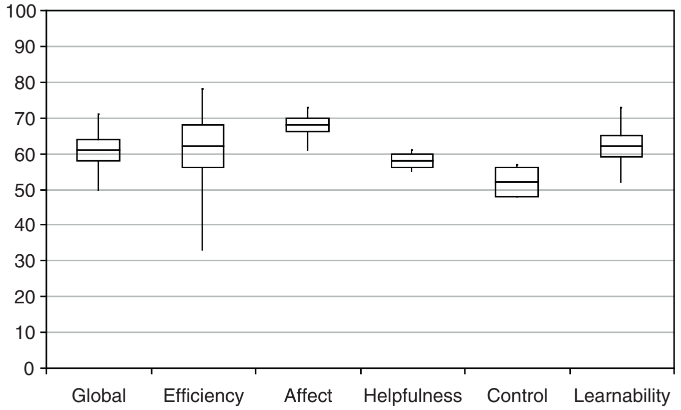


Fig. 17.3 Subjective analysis of IURISERVICE with the standardized SUMI instrument by nine judges. The graph shows the median, the upper and lower 95% confidence intervals, and range and maxima of the distributions

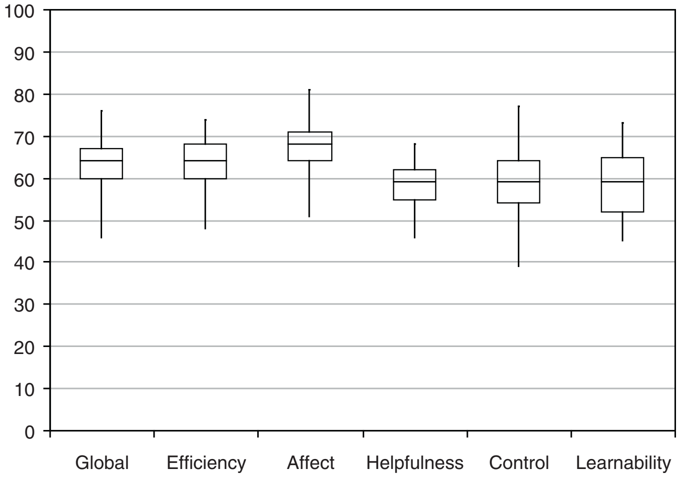


Fig. 17.4 Subjective analysis of IURISERVICE with the standardized SUMI instrument by t legal experts. The graph shows the median, the upper and lower 95% confidence intervals, and range and maxima of the distributions

questions more quickly. Finding solutions faster reduces the difficulty of the task and the stress generated by working under time pressure considerably.

17.15 Business Benefit of SEKT-Based Applications

The business benefit is derived by comparing the cost of implementing SEKT-based knowledge management solutions with potential quantitative benefits for the organization which adopts semantic knowledge technology. This requires estimates of cost and benefit in monetary terms.

Cost estimates were made for applications similar to the three SEKT case studies, including

- implementation of a SEKT based knowledge management solution in an organization in terms of cost per user
- continuous maintenance cost, also per user

Traditional technologies and semantic technology compete for the implementation of new KM systems. Under most conditions a varying degree of transfer of resources from a legacy system would be expected, but cost can only be estimated for concrete cases. (The digital library application is an upgrade, while IURISERVICE is a new system.) The cost for licensing and configuration is not considered to be significantly different for a knowledge management system based on SEKT per se. The main difference is due to the comparative internal effort required in the user organization to maintain a specific taxonomy or ontology. This accounts for roughly 50% of the cost of the KM system, the estimates were ranging from 30% to 60%. Total lifecycle costs were estimated on the basis of previously completed projects by experts from the SEKT partner organizations in separate analysis sessions, taking into consideration the particular benefits indicated by users in our analyses. Cost parameters, based on previous experience were obtained for organizations with either 3,000 or 30,000 users. For smaller organizations insufficient experience exists, but the cost per user would be higher (Table 17.4).

The difference in cost between a traditional taxonomy-based knowledge management system, and a system using semantic technology was estimated to be in the range of €0–20 per year and per user. It must be taken into account that an application integrating SEKT technology would include more functionality than existing KM systems, and the comparison would be made between not strictly equivalent systems.

Table 17.4 Estimated cost of the implementation of organization-wide knowledge management systems using semantic knowledge technology

Total lifecycle cost over a 5-year lifetime (€ per user per year)		3,000 users	30,000 users
	Mean upper/lower limits	50 30–100	35 25–60

Benefits created by semantic technology in knowledge management systems derive from the results obtained in the SEKT case studies: In business and sales terms the benefits of semantic knowledge technology would be

- reduced task time, gain in efficiency in task performance, and improved quality of the solution
- Satisfaction – users would be expected to appreciate the improved quality of their working environment

Reduced workload and stress

Further benefit should be expected from enhanced collaboration by making results and existing experience more widely available within organizations, which we did not address in this study. There is an opportunity to achieve a higher quality of task execution by using better and more information, as the digital library case study indicated. Again, this was not analyzed in detail. Professional users would be expected to exercise professional judgement when deciding about using new tools to improve either efficiency or quality of the result of their work, both of which would create benefits.

In summary, the added cost of semantic knowledge management technology over traditional solutions is small, in comparison to the potential benefits. The decisive factor which will determine the return on investment in semantic technology is the usage rate of the implemented KM system.

Risks and potential negative side effects were mentioned, but neither systematically investigated nor observed. A main concern was that increased level of reuse of existing solutions generates more stereotypical solutions, leading potentially towards standardization. This may be desirable in many contexts, but obviously is an issue to be considered as part of the strategy for introducing knowledge management systems.

The long-term goal of knowledge management is to increase the level of reuse of existing and previously elaborated information. An essential condition is that a culture of collaboration and cooperation exists, including the readiness of individuals to contribute time to collaboration. We were rather surprised by the large differences in this respect which appear to exist in different organizations where we were able to make relevant observations. The readiness to collaborate and share information actively is a precondition, not an expected outcome of the introduction of KM technology.

17.16 Conclusions

The new features introduced into knowledge management systems with semantic technology are appreciated by users, with reservations concerning mobile devices and natural language. Mobiles and PDAs are limited by the small amount of information they can display, while the case study applications display rather extensive information, which users scan to select the relevant results. Using mobile devices thus introduces additional costs for users, which may be acceptable in specific contexts only.

Natural language is regarded as a positive feature, but users who work with existing systems do not need them, and benefit may evolve slowly.

The field tests show that users recognize decisive advantages in the SEKT applications, corresponding to the specific needs of their work tasks. The advantages include gains in efficiency, reduction of workload and stress, or improved quality of the result. Regarding the quality of information for users, precision means for users that they receive precisely the information which they need in their context of work, not simply more information. Further investigations of specific quality aspects in information search should allow better understanding of this aspect in the future in order to allow the configuration of KM systems to support these needs of users.

The realization of business benefit created by semantic technology depends heavily on the ratio of active users. Costs per user are small in large organizations, and will be outweighed by benefits easily if the usage rate is high. The decisive success factor for knowledge management systems is acceptance and active participation of a large share of potential users. Acceptance would be influenced positively by the high subjective appreciation shown by users for the SEKT applications.

A note on user involvement in the SEKT project: Site visits and analysis of the tasks which users carry out were very valuable. Repeated review of the task analysis and user preferences, involving users as well as experts, show that user needs evolve as the new applications become more concrete.

Acknowledgement This work was supported by the IST Programme of the European Community under SEKT (IST-IP-2003-506826).

References

Böser T, Melchior E M (2002) User-Centred Product Creation. Best Practice in Interactive Electronic Publishing. ISBN 3-00-009652-3, http://www.vnet5.org.

Kirakowski J, Corbett M (1993) SUMI: the software usability measurement inventory. British Journal of Educational Technology 24: 210–212.

Rubin J (2001) Handbook of Usability Testing. How to Plan, Design, and Conduct Effective Tests. Chichester: Wiley.

van Rijsbergen CJ (1980) Information Retrieval. London: Butterworths.

Chapter 18
Challenges of Semantic Knowledge Management

John Davies, Marko Grobelnik, and Dunja Mladenić

18.1 Semantic Knowledge Management

Forrester (Moore 2007) estimate that more than 80% of all corporate information is unstructured. Knowledge workers are increasingly overwhelmed by information from a bewildering array of information sources: emails, intranets, the web, etc. and yet still find it hard to access the specific information required for the task at hand. This implies that knowledge worker productivity is reduced and that organisations may be making decisions on the basis of incomplete knowledge. Furthermore, an inability to access key information can lead to compliance failure.

As we have described in this volume, semantic technology is helping address these issues by associating unstructured information with domain ontologies. This makes possible more intelligent information access facilities by annotating documents (and parts of documents) with semantic meta-information – information, formally expressed, which tells the machine what the document or subdocument is about. This allows more sophisticated analysis of documents: for example, named entity recognition is a language processing technique which can identify particular locations, organisations or people mentioned in textual documents with ontological descriptions of those entities. Similarly, knowledge discovery techniques can be used to analyse content and classify it against an ontology, or indeed to derive new ontologies from content. Ontology management is then the set of tools, techniques and technology for managing the resulting ontologies and associated metadata.

This book has provided an overview of some of the best current research on Semantic Knowledge Technologies. Specifically, the book has presented basic research in the areas of ontology management, knowledge discovery and human language technology, and methods and tools for the integration of these technologies for use in semantics-based knowledge management. Two case studies applying these tools in real world applications are presented. Finally, the case studies are

J. Davies (✉)
Next Generation Web group, BT Research, Adastral Park, Ipswich, IP5 3RE, UK
e-mail: john.nj.davies@bt.com

J. Davies et al. (eds.), *Semantic Knowledge Management*,

evaluated using novel techniques appropriate to semantic applications which are also described.

All this demonstrates that semantic technology is ready to be, and is being, applied to knowledge management applications today. In the next section, we conclude with a brief look at some emerging trends and outstanding challenges.

18.2 Some Emerging Trends and Challenges

18.2.1 Microformats and the Semantic Web

An important recent development which will impact the uptake of semantic technology has been an initiative by the W3C to provide a link between the Semantic Web and microformats communities. With GRDDL ("Gleaning Resource Descriptions from Dialects of Languages"),[1] software can automatically extract information from structured Web pages to make it part of the Semantic Web (essentially by converting it to RDF). Those accustomed to expressing structured data with microformats in XHTML, for example, can thus increase the value of their existing data by porting it to the Semantic Web, at very low cost.

18.2.2 Web 2.0 & the Semantic Web

In recent years, we have seen in parallel with semantic web developments, the emergence of a group of technologies enabling web-based community and participatory systems such as flickr and del.icio.us. Known collectively as "Web 2.0," Web 2.0 systems can be seen as complementing the semantic web: many Web 2.0 systems are based around the notion of user-generated content (e.g., photographs in flickr) and user tagging, involving the annotation of data (photographs, web pages, etc.) with user-supplied descriptive words or phrases (tags). This latter is reminiscent of the semantic web's tagging of data with metadata, with the important difference that, in the case of the semantic web, this metadata is associated with a formal ontology, ensuring the consistent usage of particular tags across users and applications and formalising the data representation.

It may be that the Web 2.0 approach will prove most useful in systems where ease of use is key, while when representing mission-critical information the more formal ontological approach will tend to be adopted. Certainly it is hard to imagine a health application whereby clinicians are able to supply their own tags which may or may not be shared and understood by other health professionals and, conversely, it is doubtful that information sharing sites such as flickr would have attracted as

[1] http://www.w3.org/TR/grddl-primer/

many participants if each user had been constrained to the use of a shared ontological vocabulary.

Recently, building on the popularity of wiki technology, a number of "semantic wiki" systems have emerged (see, e.g., Voelkel et al. 2006). In such systems, semantic links between pages can be created but, importantly, there is no restriction on what the semantic links are called (i.e., there is no pre-defined ontology). One user might create a link between Cardiff and Wales annotated with "is capital of" whilst another use might create a link between Paris and France annotated with "capital city of". This informal approach, borrowed from the world of Web 2.0, is crucial to encouraging the creation of these links. Of course, users are encouraged to reuse semantic terms rather than invent their own. Users can also be encouraged to create equivalences between links where appropriate and also to indicate where one link type is a special case of another, for example, "is capital of" is a special case of "is located in". The semantic wiki is an example of the fusion of the methods of the Semantic Web, with its formally defined ontologies, and the Web2.0 approach, with informally defined folksonomies. We are likely to see more such examples in the future, along with attempts to automatically analyse folksonomies and either associate them with existing ontologies or derive more formal ontologies from them.

18.2.3 Trust and Provenance

With information and knowledge being increasingly discovered and shared across and between organisations and individuals linked via ever wider networks, the issues of trust and provenance come to the fore. The Semantic Web's current technical focus has been mainly at the ontology and logic layers. However, work in the academic community on trust inference calculi across distributed information systems is established, and work on trust and the Semantic Web is beginning to appear. See, for example, the *Semantic Web Trust and Security Resource Guide*,[2] a collection of resources on this topic. This is an area for ongoing research both into formal mechanisms for establishing trust and into the human and psychological aspects of how we can use such mechanisms.

References

Moore C (2007) The World of Content Management is Changing. Information Indepth, Oracle. http://www.oracle.com/newsletters/information-insight/content-management/feb-07/forrester-future.html.

Voelkel M, Kroetzsch M, Vrandečić D, Haller H, Studer R (2006) Semantic Mediawiki. Proceedings of the 15th International Conference on World Wide Web, Edinburgh, UK: 585–594.

[2] http://www4.wiwiss.fu-berlin.de/bizer/SWTSGuide/

Index