

Ladjel Bellatreche · Mária Bieliková ·
Omar Boussaïd · Barbara Catania ·
Jérôme Darmont · Elena Demidova ·
Fabien Duchateau · Mark Hall et al. (Eds.)

Communications in Computer and Information Science

1260

ADBIS, TPDL and EDA 2020 Common Workshops and Doctoral Consortium

International Workshops:

DOING, MADEISD, SKG, BBIGAP, SIMPDA, AIMinScience 2020
and Doctoral Consortium

Lyon, France, August 25–27, 2020, Proceedings

Communications in Computer and Information Science

1260

Commenced Publication in 2007

Founding and Former Series Editors:

Simone Diniz Junqueira Barbosa, Phoebe Chen, Alfredo Cuzzocrea,
Xiaoyong Du, Orhun Kara, Ting Liu, Krishna M. Sivalingam,
Dominik Ślęzak, Takashi Washio, Xiaokang Yang, and Junsong Yuan

Editorial Board Members

Joaquim Filipe 


Polytechnic Institute of Setúbal, Setúbal, Portugal

Ashish Ghosh

Indian Statistical Institute, Kolkata, India

Igor Kotenko 

*St. Petersburg Institute for Informatics and Automation of the Russian
Academy of Sciences, St. Petersburg, Russia*

Raquel Oliveira Prates 

Federal University of Minas Gerais (UFMG), Belo Horizonte, Brazil

Lizhu Zhou

Tsinghua University, Beijing, China

More information about this series at <http://www.springer.com/series/7899>

Ladjel Bellatreche · Mária Bieliková ·
Omar Boussaïd · Barbara Catania ·
Jérôme Darmont · Elena Demidova ·
Fabien Duchateau · Mark Hall ·
Tanja Merčun · Boris Novikov ·
Christos Papatheodorou ·
Thomas Risse · Oscar Romero ·
Lucile Sautot · Guilaine Talens ·
Robert Wrembel · Maja Žumer (Eds.)

ADBIS, TPDL and EDA 2020 Common Workshops and Doctoral Consortium


International Workshops:

DOING, MADEISD, SKG, BBIGAP, SIMPDA, AIMinScience 2020
and Doctoral Consortium

Lyon, France, August 25–27, 2020

Proceedings

Editors

Ladjet Bellatreche 
ISAE-ENSMA
Poitiers, France

Omar Boussaïd 
Université Lumière Lyon 2
Lyon, France

Jérôme Darmont 
Université Lumière Lyon 2
Lyon, France


Fabien Duchateau 
Université Claude Bernard Lyon 1
Lyon, France

Tanja Merčun 
University of Ljubljana
Ljubljana, Slovenia


Christos Papatheodorou 
Ionian University
Corfu, Greece

Oscar Romero 
Universitat Politècnica de Catalunya
Barcelona, Spain

Guilaine Talens
University of Lyon
Lyon, France

Maja Žumer 
University of Ljubljana
Ljubljana, Slovenia

Mária Bieliková 
Slovak University of Technology
Bratislava, Slovakia


Barbara Catania 
University of Genova
Genoa, Italy


Elena Demidova 
Leibniz University of Hannover
Hannover, Niedersachsen, Germany

Mark Hall 
The Open University
Milton Keynes, UK

Boris Novikov 
National Research University Higher School
of Economics
St. Petersburg, Russia

Thomas Risse 
Goethe University Frankfurt
Frankfurt am Main, Hessen, Germany

Lucile Sautot 
AgroParisTech
Montpellier, France

Robert Wrembel 
Poznań University of Technology
Poznań, Poland

ISSN 1865-0929 ISSN 1865-0937 (electronic)
Communications in Computer and Information Science
ISBN 978-3-030-55813-0 ISBN 978-3-030-55814-7 (eBook)
<https://doi.org/10.1007/978-3-030-55814-7>

© Springer Nature Switzerland AG 2020

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This volume contains a selection of the papers presented at all Workshops and Doctoral Consortium of the 24th East-European Conference on Advances in Databases and Information Systems (ADBIS 2020), the 24th International Conference on Theory and Practice of Digital Libraries (TPDL 2020), and the 16th Workshop on Business Intelligence and Big Data (EDA 2020), held during August 25–27, 2020, in Lyon, France.

This year, ADBIS, TPDL, and EDA 2020 attracted 10 workshop proposals. After evaluation of these proposals by the workshop chairs based on transparent criteria, nine workshop proposals were selected. Each workshop chair established their International Program Committee and advertisement strategy. After the paper submission deadlines, three workshops were canceled due to their low submission rates. In the end, six workshops were officially co-located with ADBIS, TPDL, and EDA 2020.

This volume contains 26 papers long papers and 5 short papers selected to be presented in these workshops and doctoral consortium (DC). An introductory chapter summarizes the main issues and contributions of all the events whose papers are included in this volume. The volume is divided into seven parts, where each part corresponds to a single satellite event and its organizers and International Program Committee are described.

The selected papers span a wide spectrum of topics related to intelligent *design* and *exploitation* of AI-driven systems, software, and applications to produce useful *insight* and tangible business *value*. This design takes into account different V's brought about by the Big Data Era (Volume, Velocity, Variety, Veracity, Value, Vocabulary). The exploitation of this amount of data/knowledge is empowered by machine/deep learning techniques and innovative hardware. One of the main particularities of our workshops is that they cover several case studies such as French cultural heritage, climate change, crowd votes, banking, health, environment, etc.

In the following, a short description and goals of each workshop and DC are given.

The First Workshop on Intelligent Data - From Data to Knowledge (DOING 2020) focuses on transforming data into information and then into knowledge. It gathered researchers from natural language processing (NLP), databases, and AI. DOING 2020 focuses on all aspects concerning modern infrastructures that support these areas, giving particular attention, but not limited to, data on health and environmental domains. The DOING workshop received 17 submissions, out of which 8 were accepted as full papers and 1 as a short paper, resulting in an acceptance rate of 50%. The workshop program also featured an invited keynote talk by Professor Marie-Christine Rousset, from the Laboratoire d'Informatique de Grenoble (LIG), France.

The Second Workshop on Modern Approaches in Data Engineering and Information System Design (MADEISD 2020) aims at addressing open questions and real potentials for various applications of modern approaches and technologies in data engineering and information system design to develop and implement effective

software services that support of information management in various organizational systems. The intention was to address the interdisciplinary character of a set of theories, methodologies, processes, architectures, and technologies in disciplines such as Data Engineering, Information System Design, Big Data, NoSQL Systems, and Model-Driven Approaches in development of effective software services. After a rigorous selection process, the MADEISD 2020 accepted four papers out of nine submissions.

The First Workshop on Scientific Knowledge Graphs (SKG 2020) provides a forum for researchers and practitioners from different fields such as Digital Libraries, Information Extraction, Machine Learning, Semantic Web, Knowledge Engineering, NLP, Scholarly Communication, and Bibliometrics that aims at exploring innovative solutions and ideas for the production and consumption of scientific knowledge graphs. SKG 2020 selected 3 full papers and 2 short papers out of 10 submissions, which corresponds to an acceptance rate of 50%. The workshop received submissions from authors of eight countries in four continents (Europe, Asia, America, Australia).

The Second Workshop of BI & Big Data Applications (BBIGAP 2020) focuses on BI and big data applications. BBIGAP 2020 aims at providing a forum to discuss recent advancements, exchange ideas, and share experiences on new issues and challenges in BI and big data applications in several domains, mainly in humanities and social sciences but also in medicine and agriculture. BBIGAP 2020 received seven submissions, out of which three were accepted as full papers and one as a short paper, resulting in an acceptance rate of 50%.

The IFIP 2.6 International Symposium on Data-Driven Process Discovery and Analysis (SIMPDA 2020) aims at offering a unique opportunity to present new approaches and research results to researchers and practitioners working in business process data modeling, representation, and privacy-aware analysis. The Program Committee chairs of SIMPDA 2020 accepted two papers covering the topics of converting relational data into processable event logs and of comparing approaches using vector space modeling for trace profiling.

The First International Workshop on Assessing Impact and Merit in Science (AIMinScience 2020) aims to bring academics, researchers, and other professionals from diverse fields together to share ideas and experiences about research assessment, relevant technologies, and applications. AIMinScience 2020 highlights the issues of developing reliable and comprehensive metrics and indicators of the scientific impact and merit of publications, data sets, research institutions, individual researchers, and other relevant entities. AIMinScience 2020 accepted for presentation two full papers and three short papers. The program of the workshop also included three invited talks by Dr. Roberta Sinatra (IT University of Copenhagen, Denmark), Prof. Yannis Manolopoulos (Professor and Vice-Rector of the Open University of Cyprus, Cyprus), and Dr. Rodrigo Costas (Centre for Science and Technology Studies at Leiden University, The Netherlands) and one special session that presented the results of a hackathon.

The ADBIS, TPD, and EDA 2020 DC is a forum where PhD students from the database and digital library communities had a chance to present their research ideas. They could gain inspiration and receive feedback from their peers and senior researchers, and to tie cooperation bounds. The DC papers aimed at describing the current status of the thesis research. The DC Committee accepted five presentations,

two of which were included in the satellite events proceedings. The topics discussed at the DC included data management, data analysis, social aspects of information technologies, and digitization of cultural heritage.

We would like to express our gratitude to every individual who contributed to the success of ADBIS, TPDL, and EDA 2020 Satellite Events. Firstly, we thank all authors for submitting their research papers to the different workshops and DC. However, we are also indebted to the members of the community who offered their precious time and expertise in performing various roles ranging from organizational to reviewing ones – their efforts, energy, and degree of professionalism deserve the highest commendations. Special thanks to the Program Committee members and the external reviewers for their support in evaluating the papers submitted to ADBIS, TPDL, and EDA 2020 Satellite Events, ensuring the quality of the scientific program. Thanks also to all the colleagues, secretaries, and engineers involved in the conference organization, as well as the workshop organizers. The committee and local organizers did a great job managing the review process and the conference, despite the COVID-19 pandemic. A special thank you to the members of the Steering Committee, in particular, its chair Yannis Manolopoulos, for all their help and guidance.

Finally, we thank Springer for publishing the proceedings containing invited and research papers in the CCIS series. Different Program Committee works relied on EasyChair, and we thank its development team for creating and maintaining the software; it offered great support throughout the different phases of the reviewing process. The conference would not have been possible without our supporters and sponsors: Eric Laboratory; Université Lyon 2; Université Lyon 3; Université de Lyon; OnlyLyon; and CNI.

Last, but not least, we thank the participants of the first online ADBIS, TPDL, and EDA 2020 Workshops and DC for sharing their works and presenting their achievements, thus providing a lively, fruitful, and constructive forum, and giving us the pleasure of knowing that our work was purposeful.

June 2020

Ladjet Bellatreche
 Mária Bielíková
 Omar Boussaïd
 Barbara Catania
 Jérôme Darmont
 Elena Demidova
 Fabien Duchateau
 Mark Hall
 Tanja Mercun
 Boris Novikov
 Christos Papatheodorou
 Thomas Risse
 Oscar Romero
 Lucile Sautot
 Guilaine Talens
 Robert Wrembel
 Maja Žumer

Organization

General Chair

Jérôme Darmont Université Lumière Lyon 2, France

Workshop Co-chairs

Ladjel Bellatreche ISAE-ENSMA Poitiers, France
Mária Bieliková Slovak University of Technology, Slovakia
Christos Papatheodorou Ionian University, Greece
Guilaine Talens Université Lyon 3, France

Doctoral Consortium Co-chairs

Barbara Catania University of Genoa, Italy
Elena Demidova L3S Research Center, Germany
Oscar Romero Universitat Politècnica de Catalunya, Spain
Maja Zumer University of Ljubljana, Slovenia

Proceedings Technical Editors

Fadila Bentayeb Université Lyon 2, France
Előd Egyed-Zsigmond INSA Lyon, France
Nadia Kabachi Université Lyon 1, France

Local Organizing Committee

Fadila Bentayeb Université Lyon 2, France
Omar Boussaïd Université Lyon 2, France
Jérôme Darmont Université Lyon 2, France
Fabien Duchateau Université Lyon 1, France
Előd Egyed-Zsigmond INSA Lyon, France
Mihaela Juganaru-Mathieu Ecole des Mines de Saint-Etienne, France
Nadia Kabachi Université Lyon 1, France
Omar Larouk ENSSIB Lyon, France
Fabrice Muhlenbach Université de Saint-Etienne, France
Habiba Osman Université Lyon 2, France
Muriel Perez Université de Saint-Etienne, France
Pegdwendé Sawadogo Université Lyon 2, France
Guilaine Talens Université Lyon 3, France
Caroline Wintergerst Université Lyon 3, France

DOING 2020 – First Workshop on Intelligent Data - from Data to Knowledge

Co-chairs

Mirian Halfeld Ferrari Université d'Orléans, INSA CVL, LIFO EA, France
Carmem S. Hara Universidade Federal do Paraná, Brazil

Program Committee

Cheikh Ba University of Gaston Berger, Senegal
Javam de Castro Machado Universidade Federal do Ceará, Brazil
Yi Chen New Jersey Institute of Technology, USA
Laurent d'Orazio IRISA, Université de Rennes, France
Vasiliki Foufi Division of Medical Information Sciences (SIMED),
 Geneva University Hospitals, University of Geneva,
 Switzerland
Michel Gagnon Polytechnique Montréal, Canada
Sven Groppe University of Luebeck, Germany
Jixue Liu University of South Australia, Australia
Shuai Ma Beihang University, China
Anne-Lyse Minard-Forst Université d'Orléans, France
Damien Novel ERTIM, INALCO, France
Fathia Sais LRI, Université Paris-Sud (Paris-Saclay), France
Agata Savary LIFAT, Université de Tours, France
Rebecca Schroeder Freitas UDESC, Universidade Estadual de Santa Catarina,
 Brazil
Aurora Trinidad Ramirez Universidade Federal do Paraná, Brazil
 Pozo

MADEISD 2020 – Modern Approaches in Data Engineering and Information System Design

Co-chairs

Ivan Lukovic University of Novi Sad, Serbia
Slavica Kordić University of Novi Sad, Serbia
Sonja Ristić University of Novi Sad, Serbia

Program Committee

Paulo Alves Instituto Politécnico de Bragança, Portugal
Moharram Challenger University of Antwerp, Belgium
Boris Delibasic University of Belgrade, Serbia
Joao Miguel Lobo University of Minho, Portugal
 Fernandes
Kresimir Fertalj University of Zagreb, Croatia
Krzysztof Goczyła Gdańsk University of Technology, Poland

Ralf-Christian Härting	Aalen University, Germany
Dušan Jakovetić	University of Novi Sad, Serbia
Miklós Krész	InnoRenew CoE and University of Primorska, Slovenia
Dragan Maćos	Beuth University of Applied Sciences Berlin, Germany
Zoran Marjanović	University of Belgrade, Serbia
Sanda Martincic-Ipsic	University of Rijeka, Croatia
Cristian Mihaescu	University of Craiova, Romania
Nikola Obrenovic	University of Novi Sad, Serbia
Maxim Panov	Skolkovo Institute of Science and Technology, Russia
Rui Humberto Pereira	Polytechnic Institute of Porto, Portugal
Aleksandar Popovic	University of Montenegro, Montenegro
Patrizia Poscic	University of Rijeka, Croatia
Adam Przybyłek	Gdańsk University of Technology, Poland
Kornelije Rabuzin	University of Zagreb, Croatia
Igor Rozanc	University of Ljubljana, Slovenia
Nikolay Skvortsov	Russian Academy of Sciences, Russia
William Steingartner	Technical University of Košice, Slovakia
Vjeran Strahonja	University of Zagreb, Croatia
Slavko Zitnik	University of Ljubljana, Slovenia

SKG 2020 – Scientific Knowledge Graphs

Co-chairs

Andrea Mannocci	Italian Research Council (CNR), Italy
Francesco Osborne	The Open University, UK
Angelo A. Salatino	The Open University, UK

Program Committee

Danilo Dessi	FIZ Karlsruhe, Germany
Ahmad Sakor	L3S Research Center, Germany
Alejandra Gonzalez-Beltran	Science and Technology Facilities Council, UK
Mohamad Yaser Jaradeh	L3S Research Center, Germany
Allard Oelen	L3S Research Center, Germany
Sepideh Mesbah	TU Delft, The Netherlands
Marilena Daquino	University of Bologna, Italy
Mehwish Alam	FIZ Karlsruhe, Germany
Drahomira Herrmannova	Oak Ridge National Laboratory, USA
Leonardo Candela	Italian Research Council, Italy
Patricia Feeny	Crossref, USA
Thanasis Vergoulis	IMSI, Athena Research Center, Greece
Jodi Schneider	University of Illinois at Urbana-Champaign, USA
Vladimir A. Fomichov	Moscow Aviation Institute (National Research University), Russia
Michael Färber	Karlsruhe Institute of Technology, Germany
Shubhanshu Mishra	Twitter, USA
Tirthankar Ghosal	Indian Institute of Technology Patna, India

BBIGAP – Second International Workshop on BI and Big Data Applications

Co-chairs

Fadila Bentayeb	Université Lyon 2, France
Omar Boussaid	Université Lyon 2, France

Program Committee

Thierry Badard	Laval University of Quebec, Canada
Hassan Badir	University of Tanger, Morocco
Ladjel Bellatreche	ISAE-ENSMA Poitiers, France
Nadjia Benblidia	Université Saad Dahleb, Algeria
Sandro Bimonte	INRAE, France
Azedine Boulmalkoul	University of Mohammadia, Morocco
Doukifli Boukraa	Jijel University, Algeria
Laurent d’Orazio	Rennes University, France
Gérald Gavin	Université Lyon 1, France
Abdessamad Imine	University of Lorraine, France
Daniel Lemire	University of Quebec in Montreal, Canada
Rokia Missaoui	University of Quebec in Gatineau, Canada
Rim Moussa	University of Carthage, Tunisia
Abdelmounaam Rezgui	Illinois State University, USA
Olivier Teste	University of Toulouse, France
Gilles Zurfluh	University of Toulouse, France

SIMPDA 2020 – 10th International Symposium on Data-Driven Process Discovery and Analysis

Co-chairs

Paolo Ceravolo	Università degli Studi di Milano, Italy
Maurice van Keulen	University of Twente, The Netherlands
Maria Teresa Gomez Lopez	University of Seville, Spain

Program Committee

Pnina Soffer	University of Haifa, Israel
Kristof Böhmer	University of Vienna, Austria
Luisa Parody	Universidad Loyola, Spain
Gabriel Tavares	Rutgers University, USA
Roland Rieke	Fraunhofer Institute for Secure Information Technology, Germany
Angel Jesus Varela Vaca	University of Seville, Spain
Massimiliano de Leoni	University of Padua, Italy
Faiza Allah Bukhsh	University of Twente, The Netherlands
Robert Singer	FH JOANNEUM, Austria

Christophe Debruyne	Trinity College Dublin, Ireland
Antonia Azzini	Consorzio per il Trasferimento Tecnologico, Italy
Mirjana Pejic-Bach	University of Zagreb, Croatia
Marco Viviani	Università degli Studi di Milano-Bicocca, Italy
Carlos Fernandez-Llatas	Universitat Politècnica de Valencia, Spain
Richard Chbeir	University of Pau and Pays de l'Adour, France
Manfred Reichert	Ulm University, Germany
Valentina Emilia Balas	Aurel Vlaicu University of Arad, Romania
Mariangela Lazoi	Università del Salento, Italy
Maria Leitner	AIT Austrian Institute of Technology, Austria
Karima Boudaoud	University of Nice Sophia, France
Chiara Di Francescomarino	Fondazione Bruno Kessler-IRST, Italy
Haralambos Mouratidis	University of Brighton, UK
Helen Balinsky	Hewlett Packard Laboratories, UK
Mark Strembeck	Vienna University of Economics and Business, Austria
Tamara Quaranta	40Labs, Italy
Yingqian Zhang	Eindhoven University of Technology, The Netherlands
Edgar Weipl	SBA Research, Austria

AIMinScience 2020 – First International Workshop on Assessing Impact and Merit in Science

Co-chairs

Paolo Manghi	CTO of the OpenAIRE infrastructure and ISTI-CNR, Italy
Dimitris Sacharidis	TU Wien, Austria
Thanasis Vergoulis	Athena Research Center, Greece

Program Committee

Alessia Bardi	ISTI-CNR, Italy
Nikos Bikakis	Atypon Inc., Greece
Lutz Bornmann	Max Planck Society, Germany
Guillaume Cabanac	University of Toulouse, France
Rodrigo Costas	Leiden University, The Netherlands
Christos Giatsidis	LIX, École Polytechnique, France
John P. A. Ioannidis	Medicine - Stanford Prevention Research Center, USA
Adam Jatowt	Kyoto University, Japan
Ilias Kanellos	ATHENA Research Center, Greece
Georgia Koutrika	ATHENA Research Center, Greece
Anastasia Krithara	NCRC Democritos, Greece
Andrea Mannocci	ISTI-CNR, Italy
Yannis Manolopoulos	Aristotle University of Thessaloniki, Greece
Giannis Nikolentzos	LIX, École Polytechnique, France
Paraskevi Raftopoulou	University of the Peloponnese, Greece
Maria Jose Rementeria	BSC, Spain

Angelo A. Salatino	The Open University, UK
Roberta Sinatra	IT University of Copenhagen, Denmark
Cassidy R. Sugimoto	Indiana University Bloomington, USA
Christos Tryfonopoulos	University of the Peloponnese, Greece
Giannis Tsakonas	University of Patras, Greece
Ludo Waltman	Leiden University, The Netherlands

Doctoral Consortium

Co-chairs

Barbara Catania	University of Genoa, Italy
Elena Demidova	L3S Research Center, Germany
Oscar Romero	Universitat Politècnica de Catalunya, Spain
Maja Zumer	University of Ljubljana, Slovenia

Contents

ADBIS, TPDL and EDA 2020 Common Workshops

Databases and Information Systems in the AI Era: Contributions from ADBIS, TPDL and EDA 2020 Workshops and Doctoral Consortium.	3
<i>Ladjel Bellatreche, Fadila Bentayeb, Mária Bielíková, Omar Boussaid, Barbara Catania, Paolo Ceravolo, Elena Demidova, Mirian Halfeld Ferrari, Maria Teresa Gomez Lopez, Carmem S. Hara, Slavica Kordić, Ivan Luković, Andrea Mannocci, Paolo Manghi, Francesco Osborne, Christos Papatheodorou, Sonja Ristić, Dimitris Sacharidis, Oscar Romero, Angelo A. Salatino, Guilaine Talens, Maurice van Keulen, Thanasis Vergoulis, and Maja Zumer</i>	

1st Workshop on Intelligent Data - From Data to Knowledge (DOING 2020)

Extraction of a Knowledge Graph from French Cultural Heritage Documents	23
<i>Erwan Marchand, Michel Gagnon, and Amal Zouaq</i>	
Natural Language Querying System Through Entity Enrichment	36
<i>Joshua Amavi, Mirian Halfeld Ferrari, and Nicolas Hiot</i>	
Public Riots in Twitter: Domain-Based Event Filtering During Civil Unrest	49
<i>Arturo Oncevay, Marco Sobrevilla, Hugo Alatrísta-Salas, and Andrés Melgar</i>	
Classification of Relationship in Argumentation Using Graph Convolutional Network	60
<i>Dimmy Magalhães and Aurora Pozo</i>	
Recursive Expressions for SPARQL Property Paths.	72
<i>Ciro Medeiros, Umberto Costa, Semyon Grigorev, and Martin A. Musicante</i>	
Healthcare Decision-Making Over a Geographic, Socioeconomic, and Image Data Warehouse	85
<i>Guilherme M. Rocha, Piero L. Capelo, and Cristina D. A. Ciferri</i>	
OMProv: Provenance Mechanism for Objects in Deep Learning	98
<i>Jian Lin and Dongming Xie</i>	

Exploiting IoT Data Crossings for Gradual Pattern Mining Through Parallel Processing 110
Dickson Odhiambo Owuor, Anne Laurent, and Joseph Onderi Orero

Cooking Related Carbon Footprint Evaluation and Optimisation 122
Damien Alvarez de Toledo, Laurent d’Orazio, Frederic Andres, and Maria C. A. Leite

2nd Workshop on Modern Approaches in Data Engineering and Information System Design (MADEISD 2020)

CrEx-Wisdom Framework for Fusion of Crowd and Experts in Crowd Voting Environment – Machine Learning Approach 131
Ana Kovacevic, Milan Vukicevic, Sandro Radovanovic, and Boris Delibasic

Temporal Network Analytics for Fraud Detection in the Banking Sector 145
László Hajdu and Miklós Krész

Abdominal Aortic Aneurysm Segmentation from Contrast-Enhanced Computed Tomography Angiography Using Deep Convolutional Networks. 158
Tomasz Dziubich, Paweł Białas, Łukasz Znaniński, Joanna Halman, and Jakub Brzeziński

Automated Classifier Development Process for Recognizing Book Pages from Video Frames 169
Adam Brzeski, Jan Cychnerski, Karol Draszawka, Krystyna Dziubich, Tomasz Dziubich, Waldemar Kortub, and Paweł Rościszewski

1st Workshop on Scientific Knowledge Graphs (SKG 2020)

DINGO: An Ontology for Projects and Grants Linked Data 183
Diego Chialva and Alexis-Michel Mugabushaka

Open Science Graphs Must Interoperate! 195
Amir Aryani, Martin Fenner, Paolo Manghi, Andrea Mannocci, and Markus Stocker

WikiCSSH: Extracting Computer Science Subject Headings from Wikipedia. 207
Kanyao Han, Pingjing Yang, Shubhanshu Mishra, and Jana Diesner

Integrating Knowledge Graphs for Analysing Academia and Industry Dynamics	219
<i>Simone Angioni, Angelo A. Salatino, Francesco Osborne, Diego Reforgiato Recupero, and Enrico Motta</i>	
A Philological Perspective on Meta-scientific Knowledge Graphs	226
<i>Tobias Weber</i>	
2nd Workshop of BI and Big Data Applications (BBIGAP 2020)	
A Scored Semantic Cache Replacement Strategy for Mobile Cloud Database Systems	237
<i>Zachary Arani, Drake Chapman, Chenxiao Wang, Le Gruenwald, Laurent d’Orazio, and Taras Basiuk</i>	
Grid-Based Clustering of Waze Data on a Relational Database	249
<i>Mariana M. G. Duarte, Rebeca Schroeder, and Carmem S. Hara</i>	
Your Age Revealed by Facebook Picture Metadata	259
<i>Sanaz Eidizadehakhcheloo, Bizhan Alipour Pijani, Abdessamad Imine, and Michaël Rusinowitch</i>	
Enacting Data Science Pipelines for Exploring Graphs: From Libraries to Studios.	271
<i>Genoveva Vargas-Solar, José-Luis Zechinelli-Martini, and Javier A. Espinosa-Oviedo</i>	
International Symposium on Data-Driven Process Discovery and Analysis (SIMPDA 2020)	
Towards the Detection of Promising Processes by Analysing the Relational Data	283
<i>Belén Ramos-Gutiérrez, Luisa Parody, and María Teresa Gómez-López</i>	
Analysis of Language Inspired Trace Representation for Anomaly Detection	296
<i>Gabriel Marques Tavares and Sylvio Barbon Jr.</i>	
The 1st International Workshop on Assessing Impact and Merit in Science (AIMinScience 2020)	
Exploring Citation Networks with Hybrid Tree Pattern Queries.	311
<i>Xiaoying Wu, Dimitri Theodoratos, Dimitrios Skoutas, and Michael Lan</i>	
ArtSim: Improved Estimation of Current Impact for Recent Articles	323
<i>Serafeim Chatzopoulos, Thanasis Vergoulis, Ilias Kanellos, Theodore Dalamagas, and Christos Tryfonopoulos</i>	

Link Prediction in Bibliographic Networks 335
*Pantelis Chronis, Dimitrios Skoutas, Spiros Athanasiou,
and Spiros Skiadopoulos*

Open Science Observatory: Monitoring Open Science in Europe 341
*George Papastefanatos, Elli Papadopoulou, Marios Meimaris,
Antonis Lempesis, Stefania Martziou, Paolo Manghi,
and Natalia Manola*

Skyline-Based University Rankings 347
*Georgios Stoupas, Antonis Sidiropoulos, Dimitrios Katsaros,
and Yannis Manolopoulos*

Doctoral Consortium

**Supervised Machine Learning Model to Help Controllers Solving
Aircraft Conflicts** 355
Md Siddiqur Rahman

Handling Context in Data Quality Management 362
Flavia Serra

Author Index 369

**ADBIS, TPDL and EDA 2020 Common
Workshops**



Databases and Information Systems in the AI Era: Contributions from ADBIS, TPDL and EDA 2020 Workshops and Doctoral Consortium

Ladjel Bellatreche¹✉, Fadila Bentayeb², Mária Bieliková³, Omar Boussaid²,
Barbara Catania⁴, Paolo Ceravolo⁵, Elena Demidova⁶, Mirian Halfeld Ferrari⁷,
Maria Teresa Gomez Lopez⁸, Carmem S. Hara⁹, Slavica Kordić¹⁰,
Ivan Luković¹⁰, Andrea Mannocci¹¹, Paolo Manghi¹², Francesco Osborne¹³,
Christos Papatheodorou¹⁴, Sonja Ristić¹⁰, Dimitris Sacharidis¹⁵,
Oscar Romero¹⁶, Angelo A. Salatino¹³, Guilaine Talens¹⁷,
Maurice van Keulen¹⁸, Thanasis Vergoulis¹⁹, and Maja Zumer²⁰

¹ LIAS/ISAE-ENSMA, Poitiers, France
bellatreche@ensma.fr

² Université de Lyon, Lyon 2, ERIC EA 3083, Lyon, France

³ Slovak University of Technology in Bratislava, Bratislava, Slovakia

⁴ University of Genoa, Genoa, Italy

⁵ Università degli Studi di Milano, Milan, Italy

⁶ L3S Research Center, Hannover, Germany

⁷ Université d'Orléans, INSA CVL, LIFO EA, Orléans Cedex 2, France

⁸ University of Seville, Seville, Spain

⁹ Universidade Federal do Paraná, Curitiba, Brazil

¹⁰ Faculty of Technical Sciences, University of Novi Sad, Novi Sad, Serbia

¹¹ ISTI-CNR, Pisa, Italy

¹² Institute of Information Science and Technologies, CNR, Rome, Italy

¹³ The Open University, Milton Keynes, UK

¹⁴ National and Kapodistrian University of Athens, Athens, Greece

papatheodor@phs.uoa.gr

¹⁵ TU Wien, Vienna, Austria

¹⁶ Universitat Politècnica de Catalunya, Barcelona, Spain

¹⁷ Université de Lyon, Jean Moulin, iaelyon, Magellan, Lyon, France

¹⁸ University of Twente, Enschede, The Netherlands

¹⁹ IMSI, Athena Research Center, Athens, Greece

²⁰ University of Ljubljana, Ljubljana, Slovenia

Abstract. Research on database and information technologies has been rapidly evolving over the last couple of years. This evolution was led by three major forces: Big Data, AI and Connected World that open the door to innovative research directions and challenges, yet exploiting

four main areas: (i) computational and storage resource modeling and organization; (ii) new programming models, (iii) processing power and (iv) new applications that emerge related to health, environment, education, Cultural Heritage, Banking, etc. The 24th East-European Conference on Advances in Databases and Information Systems (ADBIS 2020), the 24th International Conference on Theory and Practice of Digital Libraries (TPDL 2020) and the 16th Workshop on Business Intelligence and Big Data (EDA 2020), held during August 25–27, 2020, at Lyon, France, and associated satellite events aimed at covering some emerging issues related to database and information system research in these areas. The aim of this paper is to present such events, their motivations, and topics of interest, as well as briefly outline the papers selected for presentations. The selected papers will then be included in the remainder of this volume.

1 Introduction

The East-European Conference on Advances in Databases and Information Systems (ADBIS) aims at providing a forum where researchers and practitioners in the fields of databases and information systems can interact, exchange ideas and disseminate their accomplishments and visions. Inaugurated 24 years ago, ADBIS originally included communities from Central and Eastern Europe, however, throughout its lifetime it has spread and grown to include participants from many other countries throughout the world. The ADBIS conferences provide an international platform for the presentation of research on database theory, development of advanced DBMS technologies, and their advanced applications. The ADBIS series of conferences aims at providing a forum for the presentation and dissemination of research on database theory, development of advanced DBMS technologies, and their advanced applications. ADBIS 2020 in Lyon continues after St. Petersburg (1997), Poznan (1998), Maribor (1999), Prague (2000), Vilnius (2001), Bratislava (2002), Dresden (2003), Budapest (2004), Tallinn (2005), Thessaloniki (2006), Varna (2007), Pori (2008), Riga (2009), Novi Sad (2010), Vienna (2011), Poznan (2012), Genoa (2013), Ohrid (2014), Poitiers (2015), Prague (2016), Nicosia (2017), Budapest (2018) and Bled (2019).

ADBIS 2020 is coupled with TPDL Conference and EDA Workshop. This year, ADBIS, TPDL, and EDA 2020 attract six workshop proposals and Doctoral Consortium.

- The 1st Workshop on Intelligent Data - From Data to Knowledge (DOING 2020), organized by Mirian Halfeld Ferrari (Université d’Orléans, INSA CVL, LIFO EA, France) and Carmem S. Hara (Universidade Federal do Paraná, Curitiba, Brazil).
- The 2nd Workshop on Modern Approaches in Data Engineering and Information System Design (MADEISD 2020), organized by Ivan Luković, Slavica Kordić, and Sonja Ristić (all from University of Novi Sad, Faculty of Technical Sciences, Serbia).

- The 1st Workshop on Scientific Knowledge Graphs (SKG 2020), organized by Andrea Mannocci (ISTI-CNR, Pisa, Italy), Francesco Osborne (The Open University, Milton Keynes, UK) and Angelo A. Salatino (The Open University, Milton Keynes, UK).
- The 2nd Workshop of BI & Big Data Applications (BBIGAP 2020), organized by Fadila Bentayeb and Omar Boussaid (University of Lyon 2, France).
- The Tenth IFIP 2.6 - International Symposium on Data-Driven Process Discovery and Analysis (SIMPDA 2020), organized by Paolo Ceravolo (Università degli Studi di Milano, Italy), Maurice van Keulen (University of Twente, The Netherlands) and Maria Teresa Gomez Lopez (University of Seville, Spain).
- The 1st International Workshop on Assessing Impact and Merit in Science (AIMinScience 2020), organized by Paolo Manghi (Institute of Information Science and Technologies, CNR, Italy), Dimitris Sacharidis (TU Wien, Austria), and Thanasis Vergoulis (Athena Research Center, Greece).
- The ADBIS, TPD & EDA 2020 Doctoral Consortium, organized by Barbara Catania (University of Genoa, Italy), Elena Demidova (L3S Research Center, Germany), Oscar Romero (Universitat Politècnica de Catalunya, Spain) and Maja Zumer (University of Ljubljana, Slovenia).

The main ADBIS, TPD, and EDA 2020 conferences as well as each of the satellite events had its own international program committee, whose members served as the reviewers of papers included in this volume. This volume contains papers on the contributions of all workshops and the doctoral consortium of ADBIS, TPD, and EDA 2020. In the following, for each event, we present its main motivations and topics of interest and we briefly outline the papers selected for presentations. The selected papers will then be included in the remainder of this volume. Some acknowledgments from the organizers are finally provided.

2 DOING 2020: The 1st Workshop on Intelligent Data - From Data to Knowledge

Description. The 1st Workshop on Intelligent Data - From Data to Knowledge (DOING 2020), organized by Mirian Halfeld Ferrari (Université d'Orléans, INSA CVL, LIFO EA, France) and Carmem S. Hara (Universidade Federal do Paraná, Curitiba, Brazil).

Texts are important sources of information and communication in diverse domains. The intelligent, efficient and secure use of this information requires, in most cases, the transformation of unstructured textual data into data sets with some structure, and organized according to an appropriate schema that follows the semantics of an application domain. Indeed, solving the problems of modern society requires interdisciplinary research and information cross-referencing, thus surpassing the simple provision of unstructured data. There is a need for representations that are more flexible, subtle, and context-sensitive, which can also be easily accessible via consultation tools and evolve according to these principles. In this context, consultation requires a robust and efficient processing of

queries, which may involve information analysis, quality, consistency, and privacy preservation guarantees. Knowledge bases can be built as these new generation infrastructures which support data science queries on a user-friendly framework and are capable of providing the required machinery for advised decision-making.

DOING Workshop focuses on transforming data into information and then into knowledge. It gathers researchers from Natural Language Processing (NLP), Databases (DB), and Artificial Intelligence (AI). This edition features works in two main areas: (1) information extraction from textual data and its representation on knowledge bases; (2) intelligent methods for handling and maintaining these databases: new forms of requests, including efficient, flexible, and secure analysis mechanisms, adapted to the user, and with quality and privacy preservation guarantees. Overall, the purpose of the workshop is to focus on all aspects concerning modern infrastructures to support these areas, giving particular attention, but not limited, to data on health and environmental domains. DOING received 17 submissions, out of which 8 were accepted as full papers and 1 as a short paper, resulting in an acceptance rate of 50%. Each paper received three reviews from members of the program committee. The accepted papers were allocated in two technical sessions of the workshop program: NLP for information extraction, and Intelligent Data Management. The workshop program also featured an invited keynote talk entitled “Knowledge Graph Completion and Enrichment in OntoSides using Text Mining” by Professor Marie-Christine Rousset, a member of the Laboratoire d’Informatique de Grenoble (LIG). This workshop is an event connected to the working group DOING, involved in the French networks MADICS and RTR DIAMS. The workshop is the result of the collective effort of a large community, which we gratefully acknowledge. We thank the ADBIS-TPDL-EDA joint conference chairs, who worked hard to support the workshop organization. We are also grateful to the members of the program committee, who did an outstanding job, providing timely and thoughtful reviews. Finally, we are grateful to the authors who submitted their work to DOING 2020.

Selected Papers

The *NLP for information extraction Session* includes 4 papers. The first paper, entitled “Extraction of a Knowledge Graph from French Cultural Heritage Documents” [17], describes a method for extracting entities and relations in French heritage descriptive texts, using tools that are available only for English. It presents results using as input texts on Quebec’s cultural heritage. Besides heritage-specific type identification, the paper contributes by showing how tools developed for English can be used for other languages with fewer resources available, such as French.

The second paper, “Natural Language Querying System through Entity Enrichment” [2] focuses on translating natural language queries into database queries. The proposed approach is divided into a domain-dependent pre-processing and domain-independent query generation phases. This separation allows the second step to be applied on any domain, although the paper has been motivated by texts in life sciences applications.

The third paper, “Public Riots in Twitter: Domain-Based Event Filtering during Civil Unrest” [19] considers texts from Twitter to identify violence incidents during public riots. The method is composed of 4 steps: temporal clustering, term extraction, scoring, and evaluation. The method was evaluated contrasting the results for a violent and a non-violent event in Peru.

The last paper of the session, “Classification of Relationship in Argumentation using Graph Convolutional Network” [16], is in the area of argumentation mining, which aims to identify claims and evidences from text. The authors propose to model both words and relationships as nodes in a graph and apply a method based on the graph convolutional network to classify relationships among arguments.

The *Intelligent Data Management Session* includes 5 papers. The session starts with the paper “Recursive Expressions for SPARQL Property Paths” [18]. It proposes `rCFSPARQL` (restricted-context-free SPARQL), an extension to SPARQL with a subset of context-free languages. It is based on recursive expressions, which the authors claim that are more user-friendly than context-free languages for writing queries.

The second paper of the session, titled “Healthcare decision-making over a geographic, socioeconomic, and image data warehouse” [25] tackles the problem of processing healthcare analytical queries that involve geographic, socioeconomic and image data. To this end, it proposes three storage models (jointed, split and normalized) and presents an experimental study that determines their performance on a cluster running Spark extended with similarity predicates.

The third paper of the session, titled “Provenance Mechanism for Objects in Deep Learning” [15] proposes OMProv, a mechanism to keep track of the various objects involved in deep learning workflows, as well as their relationships. Each execution is modeled as a weighted directed acyclic graph, and it helps in understanding the outcome of the process. OMProv has been implemented in OMAI, a deep learning platform for the cloud.

The fourth paper, “Exploiting IoT data crossings for gradual pattern mining through parallel processing” [20], proposes a fuzzy approach to mine patterns in time series provided from multiple sources. The algorithm, called FuzzTX, applies a triangular membership function to cross time-series data sets. To show its applicability, the algorithm has been integrated to a Docker implementation of the OGC SensorThings framework.

The paper, “Cooking related Carbon Footprint Evaluation and Optimization” [1] closes the Intelligent Data Management Session. It concerns the carbon footprint of cooking, based on the location of the cooker and the ingredients in a recipe. The authors propose the `CaRbon fOotprint reciPe oPtimizER` (CROP-`PER`), which takes as input a desired carbon footprint and a money threshold, and generates as output an updated recipe with substitutions of the origin and/or type of its ingredients.

3 MADEISD 2020: The 2nd Workshop on Modern Approaches in Data Engineering and Information System Design

Description. The 2nd Workshop on Modern Approaches in Data Engineering and Information System Design (MADEISD 2020), organized by Ivan Luković, Slavica Kordić and Sonja Ristić (all from University of Novi Sad, Faculty of Technical Sciences, Serbia).

For decades, there is an open issue how to support information management process so as to produce useful knowledge and tangible business values from data being collected. Nowadays, we have a huge selection of various technologies, tools, and methods in data engineering as a discipline that helps in a support of the whole data life cycle in organization systems, as well as in information system design that supports the software process in data engineering. Despite that, one of the hot issues in practice is still how to effectively transform large amounts of daily collected operational data into the useful knowledge from the perspective of declared company goals, and how to set up the information design process aimed at production of effective software services.

The main goal of the Modern Approaches in Data Engineering and Information System Design (MADEISD) workshop is to address open questions and real potentials for various applications of modern approaches and technologies in data engineering and information system design so as to develop and implement effective software services in a support of information management in various organization systems. Intention was to address interdisciplinary character of a set of theories, methodologies, processes, architectures, and technologies in disciplines such as Data Engineering, Information System Design, Big Data, NoSQL Systems, and Model Driven Approaches in a development of effective software services. In this issue, from 9 submissions, after a rigorous selection process, we accepted 4 papers for publication at ADBIS 2020.

Selected Papers

This edition of MADEISD workshop includes four papers.

The authors of the paper “CrEx-Wisdom Framework for fusion of crowd and experts in crowd voting environment - machine learning approach” [14] address the problem of integration of experts domain knowledge with “Wisdom of crowds” by proposing machine learning based framework that enables ranking and selection of alternatives, as well as quantification of quality of crowd votes. The framework proposed by the authors enables weighting of crowd votes with respect to expert knowledge and procedures for modeling trade-off between crowd and experts satisfaction with final decisions based on ranking or selection.

In the paper “Temporal network analytics for fraud detection in the banking sector” [12], the authors present a new methodology in temporal networks for fraud detection in the banking sector. While, standard approaches of fraudulence monitoring mainly have the focus on the individual client data, the authors’ approach concentrate on the hidden data produced by the network of a transaction database. The methodology is based on a cycle detection method with the help

of which important patterns can be identified as shown by the test on real data. Proposed solution is integrated into a financial fraud system of a bank.

One of the most common imaging methods for diagnosing an abdominal aortic aneurysm, and an endoleak detection is computed tomography angiography. In the paper “Abdominal Aortic Aneurysm segmentation from contrast-enhanced computed tomography angiography using deep convolutional networks” [10], the authors address the problem of aorta and thrombus semantic segmentation, what is a mandatory step to estimate aortic aneurysm diameter. In the presented research, the three end-to-end convolutional neural networks were trained and evaluated. Finally, the authors proposed an ensemble of deep neural networks with underlying U-Net, ResNet, and VNet frameworks, and show a possibility to outperform state-of-the-art methods by 3% on the Dice metric without any additional post-processing steps.

One of the latest developments made by publishing companies is introducing mixed and augmented reality to their printed media, e.g. to produce augmented books. An important computer vision problem they are facing with is classification of book pages from video frames. In their paper “Automated classifier development process for recognizing book pages from video frames” [6], the authors address the problem by proposing an automated classifier development process that allows training classification models that run real-time, with high usability, on low-end mobile devices and achieve average accuracy of 88.95% on an in-house developed test set consisting of over 20 000 frames from real videos of 5 books for children.

4 SKG 2020: The 1st Workshop on Scientific Knowledge Graphs

Description. The 1st Workshop on Scientific Knowledge Graphs (SKG 2020), organized by Andrea Mannoce (ISTI-CNR, Pisa, Italy), Francesco Osborne (The Open University, Milton Keynes, UK) and Angelo A. Salatino (The Open University, Milton Keynes, UK).

In the last decade, we experienced a strong need for a flexible, context-sensitive, fine-grained, and machine-actionable representation of scholarly knowledge and corresponding infrastructures for knowledge curation, publishing, and processing. These technical infrastructures are becoming increasingly popular in representing scholarly knowledge as structured, interlinked, and semantically rich scientific knowledge graphs. Knowledge graphs are large networks of entities and relationships, usually expressed in W3C standards such as OWL and RDF. Scientific knowledge graphs focus on the scholarly domain and describe the actors (e.g., authors, organizations), the documents (e.g., publications, patents), and the research knowledge (e.g., research topics, tasks, technologies) in this space as well as their reciprocal relationships. These resources provide substantial benefits to researchers, companies, and policymakers by powering several data-driven services for navigating, analyzing, and making sense of research dynamics.

Some examples include Microsoft Academic Graph (MAG), AMiner, Open Academic Graph, ScholarlyData.org, PID Graph, Open Research Knowledge Graph, OpenCitations, and the OpenAIRE research graph. Current challenges in this area include: i) the design of ontologies able to conceptualise scholarly knowledge, ii) (semi-)automatic extraction of entities and concepts, integration of information from heterogeneous sources, identification of duplicates, finding connections between entities, and iii) the development of new services using this data, that allow exploring this information, measuring research impact and accelerating science.

The 1st Workshop on Scientific Knowledge Graphs (SKG 2020) is a forum for researchers and practitioners from different fields (including, but not limited to, Digital Libraries, Information Extraction, Machine Learning, Semantic Web, Knowledge Engineering, Natural Language Processing, Scholarly Communication, and Bibliometrics) in order to explore innovative solutions and ideas for the production and consumption of scientific knowledge graphs. The scientific program of SKG consists of five papers: three full papers and two short papers, out of 10 submissions, which corresponds to an acceptance rate of 50%. The workshop received submissions from authors of 8 countries in four continents (Europe, Asia, America, Australia). In this edition, three contributions are centered around acquisition, integration and enhancement of scientific knowledge graphs. One contribution covers interoperability between science graphs and another contribution describes a new knowledge organisation system to structure the information within SKGs. Crucially, ontologies are at the core of all submission highlighting the importance of their role in this endeavour.

Selected Papers

The first paper, “Dingo: an ontology for projects and grants linked data” [8], the authors present DINGO (Data INtegration for Grants Ontology), an ontology that provides a machine-readable extensible framework to model data about projects, funding, actors, and funding policies in the research landscape. DINGO is designed to yield high modelling power and elasticity to cope with the wide variety in funding, research and policy practices, which makes it applicable also to other areas besides research where funding is a crucial aspect.

The second paper, “Open science graphs must interoperate!” [5] deals with the major drivers for interoperability of Open Science Graphs (OSGs), Scientific Knowledge Graphs whose represented information may span across entities, such as research artefacts and items of their content, research organisations, researchers, services, projects, funders, and whose intent is to improve the overall FAIRness of science and support stakeholder needs, such as discovery, reuse, reproducibility, statistics, trends, monitoring, validation, and assessment. Despite being valuable individually, OSGs would greatly benefit from information exchange across their collections and, therefore, reuse and exploit the data aggregation and added value that characterise each one of them, decentralising the effort and capitalising on synergies. This work describes the critical motivations for *i)* the definition of a classification for OSGs to compare their features, identify commonalities and differences, and added value and for *ii)* the definition

of an Interoperability Framework, consisting of an information model and APIs that enable a seamless exchange of information across graphs.

The third paper, “WikiCSSH: Extracting Computer Science Subject Headings from Wikipedia” [13] focuses on domain-specific classification schemas (or subject heading vocabularies) which are used to identify, classify, and disambiguate concepts that occur in scholarly articles. Specifically, the authors introduce the Wikipedia-based Computer Science subject headings (WikiCSSH), a large-scale, hierarchically-organised subject heading vocabulary for the domain of Computer Science. It was created by developing, applying, and evaluating a human-in-the-loop workflow that first extracts an initial category tree from crowd-sourced Wikipedia data, and then combines community detection, machine learning, and hand-crafted heuristics or rules to prune the initial tree. WikiCSSH is able to distinguish between coarse-grained and fine-grained CS concepts.

The fourth paper, “Integrating Knowledge Graphs for Analysing Academia and Industry Dynamics” [3] concentrates on knowledge flows between academia and industry. Understanding their mutual influence is a critical task for researchers, governments, funding bodies, investors, and companies. To this end, the authors introduce the Academia/Industry DynAmics (AIDA) Knowledge Graph, which characterises 14M papers and 8M patents according to the research topics drawn from the Computer Science Ontology. 4M papers and 5M patents are also classified according to the type of the author’s affiliations (academy, industry, or collaborative) and 66 industrial sectors (e.g., automotive, financial, energy, electronics) obtained from DBpedia. AIDA was automatically generated by integrating different bibliographic corpora, such as Microsoft Academic Graph, Dimensions, English DBpedia, the Computer Science Ontology, and the Global Research Identifier Database.

The last paper, “A Philological Perspective on Meta-Scientific Knowledge Graphs” [30] discusses knowledge graphs and networks on the scientific process from a philological point of view. He argues that all smallest entities or lower-level constituents of science and the scientific text shall be identifiable and contain information on their use throughout all (con)texts, at the same time linking the published version to its parent nodes (e.g. collections, full data sets), and allowing for enquiry through the metadata. Knowledge graphs would then expand the researchers’ contributions on a manuscript, as we are reaching conclusions building on the analyses, transcriptions, and interpretations of other scholars.

5 BBIGAP 2020: 2nd Workshop of BI and Big Data Applications

Description. The 2nd Workshop of BI & Big Data Applications (BBIGAP 2020), organized by Fadila Bentayeb and Omar Boussaid (University of Lyon 2, France).

BBIGAP focuses on BI and big data applications. Big Data becomes a huge opportunity for computer science research but it also revolutionizes many fields,

including business, social science, social media, medicine, public administration, and so on. In this case, big data requires a revisit of data management and data analysis techniques in fundamental ways at all stages from data acquisition and storage to data transformation, analysis, and interpretation. The types of available data fall into various categories: social data (e.g., Twitter feeds, Facebook likes), data about mobility and geospatial locations (e.g., sensor data collected through mobile phones or satellite images), data collected from government administrative sources and multilingual text datasets, only to name a few. Big data bring us into a new scientific and technological era offering architectures and infrastructure (clouds, Hadoop-like computing, NoSQL databases) that allow better data management and analytics for decision-making. BBIGAP workshop received 7 submissions, out of which 3 were accepted as full papers and 1 as a short paper, resulting in an acceptance rate of 50%. Each paper received three reviews from members of the program committee.

Selected Papers

This edition of BBIGAP 2020 workshop includes four papers.

The first paper, “A Scored Semantic Cache Replacement Strategy for Mobile Cloud Database Systems” [4] considers the problem of determining the best cache entry to replace in the case of a mobile device accessing a cloud database system. The key idea is that, instead of traditional approaches (e.g., LRU, LFU), replacement techniques must take into account metrics such as current battery life, location, and connectivity quality. For this, they proposed a cache replacement method for mobile cloud database systems that utilizes decisional semantic caching. Specifically, they proposed the Lowest Scored Replacement policy (LSR), a method that uses scored metrics that determine cache relevancy and the mobile devices constraints. The objective is to derive query execution plans (QEPs) expressed as tuples (money, time, energy) and then compute the QEP that best suits the user’s requirements.

The second paper, “Grid Based Clustering of Waze Data on a Relational Database” [9] investigates the effect of a grid clustering on the performance of spatial queries, using a relational database. The authors proposed to organize spatiotemporal data as a set of relational tables, using a clustering strategy in order to group together spatiotemporal events. The objective is to show the benefits of organizing the spatial data in these clustering structures for answering spatial queries. Given data collected from traffic events, the authors proposed an approach for partitioning a geographic area of interest. They implemented their approach by using data from Waze over a period of one year, in a specific geographic area. The approach also uses spatial index, like R-trees, to speed up the execution of the type of queries analyzed at the experimental results.

The third paper, “Your Age Revealed by Facebook Picture Metadata” [11] proposes to use machine learning algorithms to infer social media (e.g. Facebook) users’ age from metadata related to their pictures. They used logistic regression to classify users’ ages into four classes. They showed how sensitive the age information of a given target user can be predicted from his/her online pictures. They investigated the feasibility of age inference attacks on Facebook

users from the metadata of pictures they publish. They showed that commenters react differently to younger and older owner pictures. The proposal is validated with experiments on a data set of 8922 random collected pictures.

The last paper, “Enacting Data Science Pipelines for Exploring Graphs: From Libraries to Studios” [29] provides an overview of data science pipeline libraries, IDE, and studios combining classic and artificial intelligence operations to query, process, and explore graphs. Then, data science pipeline environments are introduced and compared. The paper describes these environments and the design principles that they promote for enacting data science pipelines intended to query, process, and explore data collections and particularly graphs. An example is presented to illustrate how to express graph data science pipelines, that converts data into Data Frame representation and then compute graph metrics.

6 SIMPDA 2020: Tenth IFIP 2.6 - International Symposium on Data-Driven Process Discovery and Analysis

Description. The Tenth IFIP 2.6 - International Symposium on Data-Driven Process Discovery and Analysis (SIMPDA 2020), organized by Paolo Ceravolo (Università degli Studi di Milano, Italy), Maurice van Keulen (University of Twente, The Netherlands) and Maria Teresa Gomez Lopez (University of Seville, Spain).

With the increasing automation of business processes, growing amounts of process data become available. This opens new research opportunities for business process data analysis, mining, and modeling. The aim of the IFIP 2.6 - International Symposium on Data-Driven Process Discovery and Analysis is to offer a forum where researchers from different communities and the industry can share their insight into this hot new field.

Our thanks go to the authors, to the program committee, and to who participated in the organization or the promotion of this event. We are very grateful to the Università degli Studi di Milano, the University of Seville, the University of Twente, and the IFIP, for supporting this event.

Selected Papers

We selected two papers. In the first paper, “Towards the detection of promising processes by analysing the relational data” [24], the authors cope with the problem of converting relational data into processable event logs observing that multiple views can be obtained from the same relational model fostering different business process analytics.

In the last paper, “Analysis of language inspired trace representation for anomaly detection” [28], the authors develop a comparative study about approaches using vector space modeling for trace profiling. Their comparison can guide the appropriate trace profiling choice for all methods working at the intersection of Process Mining and Machine Learning.

7 AIMinScience 2020: 1st International Workshop on Assessing Impact and Merit in Science

Description. The 1st International Workshop on Assessing Impact and Merit in Science (AIMinScience 2020), organized by Paolo Manghi (Institute of Information Science and Technologies, CNR, Italy), Dimitris Sacharidis (TU Wien, Austria) and Thanasis Vergoulis (Athena Research Center, Greece).

The first edition of the International Workshop on Assessing Impact and Merit in Science (AIMinScience 2020) was held in conjunction with the 24th International Conference on Theory and Practice of Digital Libraries (TPDL 2020). We have compiled these proceedings containing the papers selected for presentation. In the last decades, the growth rate of scientific articles and related research objects (e.g., data sets, software packages) has been increasing, a trend that is expected to continue. This is not only due to the increase in the number of researchers worldwide, but also due to the growing competition that pressures them to continuously produce publishable results, a trend widely known as “publish or perish”. This trend has also been correlated with a significant drop in the average quality of published research. In this context, reliable and comprehensive metrics and indicators of the scientific impact and merit of publications, data sets, research institutions, individual researchers, and other relevant entities are now more valuable than ever. Scientific impact refers to the attention a research work receives inside its respective and related disciplines, the social/mass media etc. Scientific merit, on the other hand, is relevant to quality aspects of the work (e.g., its novelty, reproducibility, FAIR-ness, readability). It is evident that any impact or merit metrics and indicators rely on the availability of publication-related (meta)data (e.g., abstracts, citations) which, until recently, were restricted inside the data silos of publishers and research institutions. However today, due to the growing popularity of Open Science initiatives, a large amount of useful science-related data sets have been made openly available, paving the way for more sophisticated scientific impact and merit indicators (and, consequently, more precise research assessment).

Research assessment attracts the interest of researchers and professionals from diverse disciplines. For example, librarians have been working on scientometrics for many decades, while the problems regarding the management and processing of large amounts of scholarly data for research assessment applications has attracted the attention of data scientists recently. The workshop aimed to bring together professionals in academia and industry with such diverse backgrounds being interested in the aforementioned topics and to encourage their interaction. This was facilitated by the fact that this year, TPDL was co-located and co-organized with ADBIS and EDA.

AIMinScience 2020 accepted for presentation 2 full papers and 3 short papers. The program of the workshop also included 3 invited talks and one special session that presented the results of a hackathon. We would like to thank the authors for publishing and presenting their papers, the hackathon participants for their efforts, and our keynote speakers for their talk. We would like to thank the program committee for reviewing the submitted papers and providing their pro-

fessional evaluation. We hope that these proceedings will inspire new research ideas and that you will enjoy reading them.

Keynote Presentations

Predicting the future evolution of scientific output, by Prof. Yannis Manolopoulos.

In the past decade various efforts have been made to quantify scientific impact and in particular identify the mechanisms in play that influence its future evolution. The first step in this process is the identification of what constitutes scholarly impact and how it is measured. In this direction, various approaches focus on future citation count or h-index prediction, either at author or publication level, on fitting the distribution of citation accumulation or accurately identifying award winners, upcoming hot topics in research or academic rising stars. A plethora of different features have been contemplated as possible influential factors in this process and assorted machine-learning methodologies have been adopted to ensure timely and accurate estimations. In the present work, we provide an overview of the challenges rising in the field and a taxonomy of the existing approaches to identify the open issues that are yet to be addressed.

Scientific careers: evolution, interdisciplinarity, gender, and the chaperone effect, by Prof. Roberta Sinatra.

The unprecedented availability of large scale datasets about scholarly output has advanced quantitatively our understanding of how science progresses. In this talk we present a series of findings from the analysis and modelling of large-scale datasets of publications and of scientific careers. We focus on individual scientific careers and tackle the following questions: How does impact evolve in a career? What is the role of gender and of scientific chaperones in dropout and achieving high impact? How interdisciplinary is our recognition system? We show that impact, as measured by influential publications, is distributed randomly within a scientist's sequence of publications, and formulate a stochastic model that uncouples the effects of productivity, individual ability, and luck in scientific careers. We show the role of chaperones in achieving high scientific impact and we study the relation between interdisciplinarity and scientific recognitions. Taken together, we contribute to the understanding of the principles governing the emergence of scientific impact.

Beyond the impact factor: possibilities of scientometrics to understand science and society, by Dr. Rodrigo Costa.

Scientometrics have quite often been related, if not equated, with research evaluation and academic rankings. Multiple debates have emerged about the meaning of citations, the limitations of the Journal Impact Factor or the validity of the h-index as evaluative tools of researchers and research organizations. This strong focus on evaluation may have sometimes concealed other values and uses of scientometric tools regarding research management, and more broadly to study and understand science-society relationships. The main aim of this presentation is to propose and discuss some "non-conventional" uses of scientometric approaches, such as the study of the workforce composition of research organizations, the

tracking of the mobility of researchers across national boundaries, or the interactions between non-academic actors with scholarly objects via social media and altmetrics. These examples are meant to illustrate how the analytical power of scientometric indicators can expand the traditional notions of impact and success.

Selected Papers

The first paper, “Exploring citation networks with hybrid tree pattern queries” [31], proposes to use hybrid query patterns to query citation networks. These allow for both edge-to-edge and edge-to-path mappings between the query pattern and the graph, thus being able to extract both direct and indirect relationships. To efficiently evaluate hybrid pattern queries on citation graphs, a pattern matching algorithm that exploits graph simulation to prune nodes that do not appear in the final answer is applied. The obtained results on citation networks show that the proposed method not only allows for more expressive queries but is also efficient and scalable.

The second paper, “Artsim: Improved estimation of current impact for recent articles” [7] focuses on citation-based measures that try to estimate the popularity (current impact) of a scientific article. The authors identify that the state-of-the-art methods calculate estimates of popularity based on paper citation data. However, with respect to recent publications, only limited data of this type are available, rendering these measures prone to inaccuracies. Based on this finding, the authors present ArtSim, an approach that exploits article similarity, calculated using scholarly knowledge graphs, to better estimate paper popularity for recently published papers. This approach is designed to be applied on top of existing popularity measures, to improve their accuracy. To evaluate its efficiency and effectiveness in terms of improving their popularity estimates, ArtSim is applied on top of four well-known popularity measures.

The third paper, “Link prediction in bibliographic networks” [21] deals with an important problem related to the analysis of bibliographic networks to understanding the process of scientific publications. It should be noticed that a bibliographic network can be studied using the framework of Heterogeneous Information Networks (HINs). The authors compare two different algorithms for link prediction in HINs on an instance of a bibliographic network. These two algorithms represent two distinct categories: algorithms that use path-related features of the graph and algorithms that use node embeddings. The obtained results show that the path-based algorithms achieve significantly better performance on bibliographic networks.

The fourth paper, “Open science observatory: Monitoring open science in Europe” [22] focuses on monitoring and evaluating Open Science (OS) practices and research output in a principled and continuous way. These processes are recognized as one of the necessary steps towards its wider adoption. This paper presents the Open Science Observatory, a prototype online platform that combines data gathered from OpenAIRE e-Infrastructure and other public data sources and informs users via rich visualizations on different OS indicators in Europe.

The last paper, “Skyline-based university rankings” [27] proposes a novel university ranking method based on the skyline operator, which is used on multi-dimensional objects to extract the non-dominated (i.e. “prevailing”) ones. This method is characterized by several advantages, such as it is transparent, reproducible, without any arbitrarily selected parameters, based on the research output of universities only and not on publicly not traceable or random questionnaires. The proposed method does not provide meaningless absolute rankings but rather it ranks universities categorized in equivalence classes. This method is evaluated using data extracted from Microsoft Academic.

8 DC: The ADBIS, TPDL & EDA 2020 Doctoral Consortium

Description. The ADBIS, TPDL & EDA 2020 Doctoral Consortium has been organized by Barbara Catania (University of Genoa, Italy), Elena Demidova (L3S Research Center, Germany), Oscar Romero (Universitat Politècnica de Catalunya, Spain) and Maja Zumer (University of Ljubljana, Slovenia).

The ADBIS, TPDL & EDA 2020 DC was a forum where Ph.D. students from the database and digital library communities had a chance to present their research ideas. They could gain inspiration and receive feedback from their peers and senior researchers, and to tie cooperation bounds. The DC papers aimed at describing the current status of the thesis research. The DC Committee accepted five presentations, two of which were included in the satellite events proceedings. The topics discussed at the DC included data management, data analysis, social aspects of information technologies, and digitisation of cultural heritage.

Selected Papers

The topics of the two accepted papers concern data analysis and context handling in data management, respectively. In particular, the PhD research described in [23] deals with the definition and the evaluation of a Deep Neural Network (CR-DNN) model to help air traffic controllers to detect situations in which two or more airplanes are less than a minimum distance apart on their trajectory and decide what actions pilots have to apply on the fly. The model learns the best possible action(s) to solve aircraft conflicts based on past decisions and examples.

The research described in [26] deals with Data Quality Management (DQM). Data Quality naturally depends on the application context and usage needs. Despite its recognized importance, the literature lacks of proposals for context definition, specification, and usage within major DQM tasks. Starting from this consideration, the aim of the proposed research is to model and exploit contexts at each phase of the DQM process, providing a proof of concept in the domain of Digital Government.

9 Conclusion

We hope readers will find the content of this volume interesting and will be inspired to look further into the challenges that are still ahead for the design of advanced databases and information systems, with a special reference to Big

Data, AI and Connected World. We are really sure that this volume content will stimulate new ideas for further research and developments by both the scientific and industrial communities.

ADBIS, TPDL & EDA 2020 workshops and Doctoral Consortium organizers would like to express their thanks to everyone who contributed to the volume content. We thank the authors, who submitted papers. Special thanks go to the Program Committee members as well as to the external reviewers of the main conferences and of each satellite event, for their support in evaluating the submitted papers, providing comprehensive, critical, and constructive comments, and ensuring the quality of the scientific program and of this volume.

References

1. Alvarez de Toledo, D., D’Orazio, L., Andres, F., Leite, M.: Cooking related carbon footprint evaluation and optimisation. In: Bellatreche, L., et al. (eds.) ADBIS, TPDL and EDA 2020 Common Workshops and Doctoral Consortium, CCIS 1260, pp. 122–128. Springer, Cham (2020)
2. Amavi, J., Halfeld-Ferrari, M., Hiot, N.: Natural language querying system through entity enrichment. In: Bellatreche, L., et al. (eds.) ADBIS, TPDL and EDA 2020 Common Workshops and Doctoral Consortium, CCIS 1260, pp. 36–48. Springer, Cham (2020)
3. Angioni, S., Salatino, A., Osborne, F., Recupero, D.R., Enrico Motta, E.: Integrating knowledge graphs for analysing academia and industry dynamics. In: Bellatreche, L., et al. (eds.) ADBIS, TPDL and EDA 2020 Common Workshops and Doctoral Consortium, CCIS 1260, pp. 219–225. Springer, Cham (2020)
4. Arani, Z., Chapman, D., Wang, C., Gruenwald, L., D’orazio, L., Basiuk, T.: A scored semantic cache replacement strategy for mobile cloud database systems. In: Bellatreche, L., et al. (eds.) ADBIS, TPDL and EDA 2020 Common Workshops and Doctoral Consortium, CCIS 1260, pp. 237–248. Springer, Cham (2020)
5. Aryani, A., Fenner, M., Manghi, P., Mannocci, A., Stocker, M.: Open science graphs must interoperate! In: Bellatreche, L., et al. (eds.) ADBIS, TPDL and EDA 2020 Common Workshops and Doctoral Consortium, CCIS 1260, pp. 195–206. Springer, Cham (2020)
6. Brzeski, A., et al.: Automated classifier development process for recognizing book pages from video frames. In: Bellatreche, L., et al. (eds.) ADBIS, TPDL and EDA 2020 Common Workshops and Doctoral Consortium, CCIS 1260, pp. 169–179. Springer, Cham (2020)
7. Chatzopoulos, S., Vergoulis, T., Kanellos, I., Dalamagas, T., Tryfonopoulos, C.: Artsim: improved estimation of current impact for recent articles. In: Bellatreche, L., et al. (eds.) ADBIS, TPDL and EDA 2020 Common Workshops and Doctoral Consortium, CCIS 1260, pp. 323–334. Springer, Cham (2020)
8. Chialva, D., Mugabushaka, A.M.: Dingo: an ontology for projects and grants linked data. In: Bellatreche, L., et al. (eds.) ADBIS, TPDL and EDA 2020 Common Workshops and Doctoral Consortium, CCIS 1260, pp. 183–194. Springer, Cham (2020)
9. Duarte, M.M.G., Schroeder, R., Hara, C.S.: Grid based clustering of waze data on a relational database. In: Bellatreche, L., et al. (eds.) ADBIS, TPDL and EDA 2020 Common Workshops and Doctoral Consortium, CCIS 1260, pp. 249–258. Springer, Cham (2020)

10. Dziubich, T., Białas, P., Znaniecki, L., Halman, J., Brzeziński, J.: Abdominal aortic aneurysm segmentation from contrast-enhanced computed tomography angiography using deep convolutional networks. In: Bellatreche, L., et al. (eds.) ADBIS, TPD and EDA 2020 Common Workshops and Doctoral Consortium, CCIS 1260, pp. 158–168. Springer, Cham (2020)
11. Eidizadehakhcheloo, S., Pijani, B.A., Imine, A., Rusinowitch, M.: Your age revealed by Facebook picture metadata. In: Bellatreche, L., et al. (eds.) ADBIS, TPD and EDA 2020 Common Workshops and Doctoral Consortium, CCIS 1260, pp. 259–270. Springer, Cham (2020)
12. Hajdu, L., Krész, M.: Temporal network analytics for fraud detection in the banking sector. In: Bellatreche, L., et al. (eds.) ADBIS, TPD and EDA 2020 Common Workshops and Doctoral Consortium, CCIS 1260, pp. 145–157. Springer, Cham (2020)
13. Han, K., Yang, P., Mishra, S., Diesner, J.: Wikicssh: extracting computer science subject headings from Wikipedia. In: Bellatreche, L., et al. (eds.) ADBIS, TPD and EDA 2020 Common Workshops and Doctoral Consortium, CCIS 1260, pp. 207–218. Springer, Cham (2020)
14. Kovacevic, A., Vukicevic, M., Radovanovic, S., Delibasic, B.: Crex-wisdom framework for fusion of crowd and experts in crowd voting environment - machine learning approach. In: Bellatreche, L., et al. (eds.) ADBIS, TPD and EDA 2020 Common Workshops and Doctoral Consortium, CCIS 1260, pp. 131–144. Springer, Cham (2020)
15. Lin, J., Xie, D.: OMProv: provenance mechanism for objects in deep learning. In: Bellatreche, L., et al. (eds.) ADBIS, TPD and EDA 2020 Common Workshops and Doctoral Consortium, CCIS 1260, pp. 98–109. Springer, Cham (2020)
16. Magalhaes, D., Pozo, A.: Classification of relationship in argumentation using graph convolutional network. In: Bellatreche, L., et al. (eds.) ADBIS, TPD and EDA 2020 Common Workshops and Doctoral Consortium, CCIS 1260, pp. 60–71. Springer, Cham (2020)
17. Marchand, E., Gagnon, M., Zouaq, A.: Extraction of a knowledge graph from French cultural heritage documents. In: Bellatreche, L., et al. (eds.) ADBIS, TPD and EDA 2020 Common Workshops and Doctoral Consortium, CCIS 1260, pp. 23–35. Springer, Cham (2020)
18. Medeiros, C.M., Costa, U.S., Grigorev, S.V., Musicante, M.A.: Recursive expressions for SPARQL property paths. In: Bellatreche, L., et al. (eds.) ADBIS, TPD and EDA 2020 Common Workshops and Doctoral Consortium, CCIS 1260, pp. 72–84. Springer, Cham (2020)
19. Oncevay, A., Sobrevilla, M., Alatrística-Salas, H., Melgar, A.: Public riots in Twitter: Domain-based event filtering during civil unrest. In: Bellatreche, L., et al. (eds.) ADBIS, TPD and EDA 2020 Common Workshops and Doctoral Consortium, CCIS 1260, pp. 49–59. Springer, Cham (2020)
20. Owuor, D., Laurent, A., Orero, J.: Exploiting IoT data crossings for gradual pattern mining through parallel processing. In: Bellatreche, L., et al. (eds.) ADBIS, TPD and EDA 2020 Common Workshops and Doctoral Consortium, CCIS 1260, pp. 110–121. Springer, Cham (2020)
21. Chronis, P., Dimitrios Skoutas, S.A., Skiadopoulos, S.: Link prediction in bibliographic networks. In: Bellatreche, L., et al. (eds.) ADBIS, TPD and EDA 2020 Common Workshops and Doctoral Consortium, CCIS 1260, pp. 335–340. Springer, Cham (2020)

22. Papastefanatos, G., et al.: Open science observatory: monitoring open science in Europe. In: Bellatreche, L., et al. (eds.) ADBIS, TPD and EDA 2020 Common Workshops and Doctoral Consortium, CCIS 1260, pp. 341–346. Springer, Cham (2020)
23. Rahman, M.S.: Supervised machine learning model to help controllers solving aircraft conflicts. In: Bellatreche, L., et al. (eds.) ADBIS, TPD and EDA 2020 Common Workshops and Doctoral Consortium, CCIS 1260, pp. 355–361. Springer, Cham (2020)
24. Ramos-Gutiérrez, B., Parody, L., López, M.T.G.: Towards the detection of promising processes by analysing the relational data. In: Bellatreche, L., et al. (eds.) ADBIS, TPD and EDA 2020 Common Workshops and Doctoral Consortium, CCIS 1260, pp. 283–295. Springer, Cham (2020)
25. Rocha, G.M., Capelo, P.L., Dutra De Aguiar Ciferri, C.: Healthcare decision-making over a geographic, socioeconomic, and image data warehouse. In: Bellatreche, L., et al. (eds.) ADBIS, TPD and EDA 2020 Common Workshops and Doctoral Consortium, CCIS 1260, pp. 85–97. Springer, Cham (2020)
26. Serra, F.: Handling context in data quality management. In: Bellatreche, L., et al. (eds.) ADBIS, TPD and EDA 2020 Common Workshops and Doctoral Consortium, CCIS 1260, pp. 362–367. Springer, Cham (2020)
27. Stoupas, G., Antonis Sidiropoulos, D.K., Manolopoulos, Y.: Skyline-based university rankings. In: Bellatreche, L., et al. (eds.) ADBIS, TPD and EDA 2020 Common Workshops and Doctoral Consortium, CCIS 1260, pp. 347–352. Springer, Cham (2020)
28. Tavares, G.M., Junior, S.B.: Analysis of language inspired trace representation for anomaly detection. In: Bellatreche, L., et al. (eds.) ADBIS, TPD and EDA 2020 Common Workshops and Doctoral Consortium, CCIS 1260, pp. 296–308. Springer, Cham (2020)
29. Vargas-Solar, G., Zechinelli-Martini, J., Espinosa-Oviedo, J.A.: Enacting data science pipelines for exploring graphs: from libraries to studios. In: Bellatreche, L., et al. (eds.) ADBIS, TPD and EDA 2020 Common Workshops and Doctoral Consortium, CCIS 1260, pp. 271–280. Springer, Cham (2020)
30. Weber, T.: A philological perspective on meta-scientific knowledge graphs. In: Bellatreche, L., et al. (eds.) ADBIS, TPD and EDA 2020 Common Workshops and Doctoral Consortium, CCIS 1260, pp. 226–233. Springer, Cham (2020)
31. Wu, X., Theodoratos, D., Skoutas, D., Lan, M.: Exploring citation networks with hybrid tree pattern queries. In: Bellatreche, L., et al. (eds.) ADBIS, TPD and EDA 2020 Common Workshops and Doctoral Consortium, CCIS 1260, pp. 311–322. Springer, Cham (2020)

**1st Workshop on Intelligent Data - From
Data to Knowledge (DOING 2020)**



Extraction of a Knowledge Graph from French Cultural Heritage Documents

Erwan Marchand, Michel Gagnon^(✉), and Amal Zouaq

LAMA-WeST Lab, Polytechnique Montréal, Montreal, Canada
michel.gagnon@polymtl.ca
<http://www.labowest.ca/>

Abstract. Cultural heritage in Quebec is often represented as collections of French documents that contain a lot of valuable, yet unstructured, data. One of the current aims of the Quebec Ministry of Culture and Communications (MCCQ) is to learn a knowledge graph from unstructured documents to offer an integrated semantic portal on Quebec's cultural heritage. In the context of this project, we describe a machine learning and open information extraction approach that leverages named entity extraction and open relation extraction in English to extract a knowledge graph from French documents. We also enhance the generic entities that can be recognized in texts with domain-related types.

Our results show that our method leads to a substantial enrichment of the knowledge graph based on the initial corpus provided by the MCCQ.

Keywords: Natural language processing · Open information extraction · Supervised learning · Cultural heritage

1 Introduction

Digital humanities is a field which aims to make resources in the humanities easily accessible and reusable. In this paper, we present an approach to learn a knowledge graph in the domain of cultural heritage. To our knowledge, there are very few initiatives that target cultural heritage data in French. The Quebec Ministry of Culture and Communications has a large number of cultural heritage documents in French whose content is not found in any knowledge base. One of the objectives of this work is to extract some of this information using open information extraction tools that are built primarily for the English language. The second objective is to offer a supervised learning model for the extraction of entities with domain-specific types and to improve the extraction already carried out on general types of entities. The third objective of our work is to automatically build a cultural knowledge graph using the extracted information. Our approach which, to our knowledge, has never been proposed in the domain of Quebec cultural heritage,

leverages automatically extracted relational feature maps of entities of already known types to enable the identification of new entities of the same types.

The paper is structured as follows. In Sect. 2, we present the existing approaches for the extraction of entities and relations. In Sect. 3, we describe the overall architecture of our approach. In Sect. 4, we present the knowledge graph resulting from the application of our extraction method. In Sect. 5, we evaluate the extraction carried out using our model with and without the application of a supervised learning model. Finally, Sect. 6 presents our conclusions.

2 Related Work

Many open information extraction tools exist but they generally work for the English language. Among the most recent tools are Ollie [13], ClausIE [4], the Stanford NLP toolkit [11], NESTIE [3], MinIE [6] and MinSciE [10]. These tools are, however, ill-suited for use on French texts and entity extractors are not specialized for extracting entities from the cultural heritage domain. Some tools are domain-related. For instance, the PKDE4J [14] tool offers automatic extraction of entities in English in the biomedical field. In fact, our initial architecture was inspired by this tool. There are, however, few open information extraction initiatives in French. One notable approach [7] is based on a modification of the ReVerb tool so that it allows French text processing to finally extract simple facts from Wikipedia pages. However this approach does not specify the types of entities so it cannot be directly used to populate a knowledge base. In fact, we need to know, for each entity, to which class (or type) it pertains.

To represent data in a knowledge base, it is necessary to have an adequate conceptual schema that is used to represent the entities and their relations. This structured representation is called an *ontology*. In the cultural heritage field, extensions and adaptations of the CIDOC Conceptual Reference Model [5] have been proposed in some specific domains (for example, virtual museum [1], archival descriptions [9] and archeological sciences [12]), but there is not, to our knowledge, any stable and largely adopted ontology that can be used in a task such as the one described in this paper. At a generic level, however, many organizations have taken an interest in the representation of structured data and several large models have emerged. The knowledge bases DBpedia [2] and Wikidata [15], based on ontologies, aim to represent Wikipedia data in a structured form. The Dublin Core taxonomy [16] formalizes the representation of digital data such as videos, images or web pages. The schema.org taxonomy [8], which we use in our work, is a collaborative work between Bing, Google and Yahoo with the aim of creating a structured data schema that can easily be reused. Overall, we are not aware of a work in the state of the art that leverages English tools to extract a cultural heritage knowledge graph from French resources.

3 Proposed Architecture

Figure 1 presents our architecture. The inputs consist of 17 138 cultural heritage documents, the schema.org taxonomy which allows us to identify the types of real

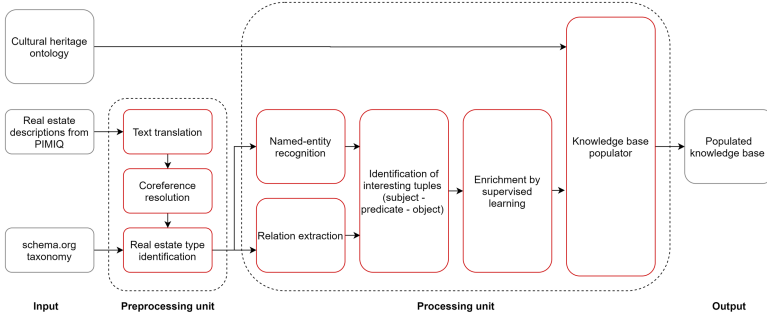


Fig. 1. Proposed architecture to populate a cultural heritage knowledge graph using real estate descriptions

estate heritage, and a heritage ontology. The cultural heritage documents come from the PIMIQ database of the Quebec Ministry of Culture and Communications (available at <http://www.patrimoine-culturel.gouv.qc.ca/rpcq/>). Here, we consider only the documents that describe real estates. Out of the 17 138 PIMIQ heritage documents, 5501 are typed (for example *Sainte-Geneviève's church* is of type *Church*) and 11,637 are addresses (for example *141 Bateau-Passeur street*). The schema.org taxonomy allows us to extract the type of some of our real estates as we discuss in Sect. 3.1. Our heritage ontology allows us to structure our output data when populating our cultural heritage knowledge graph. The output of our model consists of a populated knowledge base containing the entities and relations extracted from heritage documents. Our architecture is mainly composed of two modules: a data pre-processing unit and a processing unit.

3.1 Pre-processing Unit

The purpose of the preprocessing unit is to format the text in order to maximize the efficiency of the processing module. It is made up of three sub-modules: a text translation module, a coreference resolution module and a module for identifying types of historic real estate (e.g. Church).

Text Translation. As previously stated, there are very few named entity annotators and relation extractors aimed at the French language. We thus rely on a translation step that exploits APIs working on English documents. In this work, we use the Google API. We translate the texts into English to enable the use of the open information extraction tools from the Stanford NLP toolkit. In fact, after various comparative experiments on open information extraction tools using our data, we found out that Stanford NLP NER obtained the best recall. However, it does not handle French texts.

Co-reference Resolution. The second module of the pre-processing stage consists of a module for resolving co-references. The role of this module is to eliminate personal and possessive pronouns by replacing them with the entities they

refer to. Thus, the text “*Victor Hugo is an author. He wrote Les Misérables.*” can be replaced, using this module, by *Victor Hugo is an author. Victor Hugo wrote Les Misérables.* The main purpose here is to improve the quality of the facts obtained with open relation extractors at a later stage. Without this module, the extracted tuples would be $\langle \textit{Victor Hugo}, \textit{is}, \textit{author} \rangle$ and $\langle \textit{He}, \textit{wrote}, \textit{Les Misérables} \rangle$. The second tuple could not be added to our knowledge graph because *He* is not a named entity. In this module, we use the Stanford co-reference API on the translated text.

Real Estate Type Identification. The third and final module of the pre-processing unit is a module for identifying the types of real estate assets. This module has mainly two objectives: to allow the classification of heritage according to its category (House, Church, etc.) but also to complete the previous co-reference resolution module by allowing the identification of mentions of a heritage site within a text. Indeed, we noticed, by analyzing the texts manually, that real estate assets are often mentioned by their type. We found, for example, in the description of *Sainte-Geneviève’s church*, the sentences *The current church is built from 1782 to 1787* and *From 1844 to 1850, the church is enlarged.* In both cases, *church* refers to *Sainte-Geneviève’s church*. We also noticed, by manual analysis of the texts, that the type of a real estate can generally be extracted from its name (e.g., *Sainte-Geneviève’s church*).

The schema.org taxonomy contains many classes associated with building types, including recurrent categories of our heritage sites such as the classes *Church*, *House* and *Bridge*. We have therefore generated, using this taxonomy, a list of building types which we then compare to each real estate heritage name. When we find one of the types within the name of a heritage site, we associate the site with this type and replace the references to the heritage site in the text with its name. For example, in the document that describes *Sainte-Geneviève’s church*, every occurrence of the mention *the church* in the text is replaced by the name *Sainte-Geneviève’s church*. We were able to identify the type of 2853 among the 5501 (or 52%) typed heritage sites. Unidentified heritage sites have a type that is not found in the schema.org taxonomy.

3.2 Processing Unit

The processing unit is made up of five modules: the entity and relation extraction modules using Stanford NLP’s open information extraction tools, the identification of relevant tuples, which filters the extracted relations, the supervised learning model, which improves the extractions and finally, the knowledge graph population module, which converts the extracted tuples into data elements that can be added to the knowledge base. In this section, we describe each of these modules.

Relation Extraction. The first module of our processing unit is a relation extraction module. To extract relations from texts, we have considered and evaluated three open relation extraction tools: Stanford NLP, OllIE and MinIE. To select the tool with the best performances for our field, we carried out relation extractions on six heritage description texts from PIMIQ and evaluated, for

each tool, the number of exact relations extracted and relevant to our knowledge graph. The results of these are presented in Table 1. In each case, the Stanford NLP tool is the one displaying the highest performance.

Table 1. Number of exact relations extracted and relevant to our knowledge base based on the tool used per heritage description

Name of the heritage description text	StanfordOIE	OllIE	MinIE
Domaine Beauséjour	14	5	9
90, Rang 1 Neigette Est	5	3	4
Ancien magasin général de la mine Johnson	11	7	5
605, avenue Royale	8	6	1
Ancien couvent Notre-Dame-de-la-Merci	18	10	12
17-25, rue des Remparts	14	8	6

Using Stanford NLP’s open relation extraction tool allows us to obtain a list of tuples $\langle \text{Subject}, \text{Predicate}, \text{Object} \rangle$ representing relations between entities that could be identified. Using this tool allowed us to extract a total of 256,409 tuples.

Named-entity Recognition. The purpose of the extraction module for named entity types is to identify the type of the largest possible number of entities within the texts. This task is necessary because it identifies the class to which an instance must be linked. The tool used for the extraction of entity types is the Stanford NER tool from the Stanford NLP toolkit. We were able to identify the type of 22,116 entities among 20 types (*Person*, *Country*, *Organization*, etc.). To this list, we also add the list of real estates assets extracted by the module *Real estate type identification*, where each item is associated to the type *Real estate*. We also add to this list the types (such as *Church* and *House*) that have been extracted from the schema.org taxonomy.

Identification of Relevant Tuples. The *relation extraction* module returns a very large number of tuples. We must filter this list in order to keep only the tuples that can be added to our knowledge graph. This filtering is performed using the types of named entities extracted by the *named-entity recognition* module. For each tuple $\langle \text{Subject}, \text{Predicate}, \text{Object} \rangle$, the types of the subject and object, if they were identified, are added to the tuple to obtain a new typed tuple $\langle \text{Subject (type of subject)}, \text{Predicate}, \text{Object (type of object)} \rangle$. In order to enrich our ontology schema, we manually selected, among the most frequent tuples $\langle (\text{type of subject}), \text{Predicate}, (\text{type of object}) \rangle$, ten patterns representing relevant relations for our heritage ontology. These relations represent the nationality as well as the profession of a person but also various relations between a person and a real estate heritage such as *was built by* as well as relations about the date of construction of the real estate. Thus, for a tuple to be kept by this

module, the types of the subject and object as well as the predicate must be found in one of the patterns. An example of one of our patterns is $\langle\langle\text{REAL ESTATE}\rangle\rangle$, was built by, $\langle\langle\text{PERSON}\rangle\rangle$. After going through this list of patterns, we kept a total of 6542 tuples. Table 2 shows all the patterns defined in this module.

Table 2. List of patterns defined in the *Relevant tuples identification* module

Type of subject	Predicate(s)	Type of object	Relation meaning
PERSON	is/is of/is by/is to	NATIONALITY	Nationality of a person
PERSON	is/is of/is by/is to	TITLE	Profession of a person
PERSON	built	REAL ESTATE	Person who built the real estate
REAL ESTATE	was built by	PERSON	Person who built the real estate
REAL ESTATE	was built in	DATE	Date of construction of a real estate
REAL ESTATE	was built around	DATE	Approximate date of construction of a real estate
REAL ESTATE	was built for	PERSON	Person for who built the real estate
REAL ESTATE	was sold by	PERSON	Person who sold the real estate
PERSON	sold	REAL ESTATE	Person who sold the real estate
REAL ESTATE	was sold to	PERSON	Person who the real estate was sold to

Enrichment by Supervised Learning. This module is our main contribution together with the whole pipeline for knowledge graph population from cultural heritage documents. It improves the identification of entities of the *Person* and *Real estate* types. We have noticed that the module for recognizing named entities missed a large number of entities of the type *Person* in our cultural heritage documents, and many real estate assets were also not properly identified. In fact, the type *Real estate* does not exist among the types of the Stanford NER tool. We have, therefore, designed two supervised learning models which take as input three *feature maps* representing a given entity and return a boolean indicating, for the first model, if the entity is of type *Person* and, for the second model, if the entity is of type *Real estate*.

The three feature maps representing a given entity are:

- The typographic map: It contains a list of typographic characteristics, which are the number of words composing the entity name, the number of characters, the number of capital letters and the count of numbers;
- The sub-sequences type map: For each type that is recognized by the *Named-entity recognition* module, we check if it can be associated to some sub-sequence of words in the entity mention. For example, the entity *American inventor William Howe* contains sub-sequences of type *Nationality*, *Title* and *Person*. For each type, we assign 1 to its corresponding position in the map if we were able to find a sub-sequence of this type in the entity mention, 0 otherwise;
- The property map: It leverages statistics on the most common relations that are usually used with a given entity. To identify these relations, we use the outputs obtained by the tool *StanfordOIE*. These entries depend on the type

of the entity that one wishes to identify. In order to generate the entries for a type of entity, we extract the already identified entities of this type, we then identify, for each one, the relations in which they are used as well as the position of the entity in the relation. For example, the extracted relation $\langle \textit{The house, was built by, Octave Lapointe} \rangle$ will result in the inclusion of feature $\langle \textit{Object - was built by} \rangle$ for the entity *Octave Lapointe* of type *Person*. We then keep only the most frequent N couples $\langle \textit{position - relation} \rangle$. To facilitate the identification of negative entities (whose type does not correspond to the type sought), we also add the N most frequent pairs of entities whose type could be identified by the tool *StanfordNER* but is different from the target type. We are not aware of any other work that uses such a property map to help the identification of the type of an entity.

Table 3. Example of inputs in the supervised learning model with $N = 10$ for the classification of entities of type *Person*. Expected outputs are 1 and 0 respectively.

		“François Fortier”	“May 1979”	
Input	Typographic map (4 integers)	Number of words	2	2
		Number of characters	16	8
		Number of capital letters	2	1
		Number of numbers	0	4
	Sub-sequences type map (21 booleans)	REAL ESTATE	0	0
		DATE	0	1
		PERSON	1	0
		...		
	relation map (15 booleans)	$\langle \textit{Subject - is} \rangle$	1	0
		$\langle \textit{Subject - has} \rangle$	0	0
		$\langle \textit{Subject - is of} \rangle$	0	0
		$\langle \textit{Subject - is by} \rangle$	1	0
		$\langle \textit{Subject - is to} \rangle$	0	0
		$\langle \textit{Subject - of} \rangle$	0	0
		$\langle \textit{Subject - was} \rangle$	0	0
		$\langle \textit{Subject - built} \rangle$	1	0
$\langle \textit{Object - belongs to} \rangle$		0	0	
$\langle \textit{Subject - is 's} \rangle$		0	0	
$\langle \textit{Object - is in} \rangle$		0	0	
$\langle \textit{Subject - is in} \rangle$		0	0	
$\langle \textit{Object - was built in} \rangle$	0	1		
$\langle \textit{Subject - was built in} \rangle$	0	0		
$\langle \textit{Object - is} \rangle$	0	0		

Table 3 shows two examples of the inputs and expected outputs in the supervised learning model for the identification of entities of type *Person*, taking a value $N = 10$ (note that in our experiments we used $N = 100$). The 16 entries not indicated on the sub-sequences type map all have the value “0” for the two examples. In the property feature map, we see that there are only 15 entries, and not 20, as one may expect, because in 5 cases (the ones in green), the property is frequent for both *Person* and non-*Person* entities. We can see that 5 properties

(in blue) are frequent only for entities of type *Person* and, finally, 5 properties (in red) for entities that are not associated with this type.

Our learning method takes advantage of entities whose type has already been properly identified by the *Named-entity recognition* module or the *Real estate type identification* module. Using these entities, we generate two learning sets, $L_{positives}$ and $L_{negatives}$. $L_{positives}$ corresponds to the list of entities whose type is the same as our learning model target type (e.g. *Person*) and $L_{negatives}$ corresponds to the list of entities whose type is distinct and does not overlap with the type considered. For each entity in these sets, we generate the three maps as described above. Our learning set, which contains a total of 23,212 items, is then generated with $L_{sets} = L_{positives} \cup L_{negatives}$.

We then split the list L_{sets} into a learning set $L_{learning}$ (85% of all the set) and a test set L_{test} (the remaining 15% of all the set). In order to evaluate our models, we proceeded by cross validation on 10 folds. Finally, we tested both models on the tests sets. In order to fix the hyper parameters in our models, we tested various combinations to find the one with the highest average F1 score during cross validation. Our final classifier architecture, that has been used for both types of entities (person and real estate), is a feed-forward neural net, with two hidden layers (50 and 30 neurons for the first and second layer, respectively), and using a Sigmoid activation function.

On the test set, our model to identify entities of type *Person* reached an F1 score of 0.868 and our model to identify entities of type *Real estate* obtained an average F1 score of 0.883.

Once both our models were generated, we were able to use them to identify new entities of type *Person* or *Real estate*. For each entity subject or object of a tuple extracted by our *Relation extraction* module whose type was yet unknown, we generate the input maps for both models. We then run each model to predict if the entity is of type *Person*, *Real estate* or still unknown. With our approach, we were able to identify 4773 new entities of type *Person* while we previously identified 10,547 of such entities using only existing Open Information Tools and no supervised learning models. We were also able to identify 8471 new entities of type *Real estate* while we previously identified 9816 of such entities.

Knowledge Graph Populator. The *knowledge graph populator*. module is straightforward. For each tuple extracted and validated by one of the patterns of the *Relevant tuples identification* module, and after applying the classifier model described previously, we generate relation instances in accordance with our heritage ontology and add them to our knowledge graph.

4 Ontology and Knowledge Graph

We have built our ontology in parallel with the extraction process and its aim is to enable a structured representation of the data that was extracted. Figure 2 shows our heritage ontology. Each entity extracted by our system is represented as an instance of one of the subclasses of the class *Entity*. Each relation extracted is represented as an instance of one of the subclasses of the class *Relation*. Each

document that describes a heritage site is represented as an instance of the class *Document*. We then link each document to the entities it contains and link each entity to the relations in which it participates.

Our approach allowed us to identify the classes and properties of our heritage ontology and our extraction work allowed us to extract tuples that became instances and properties in our knowledge graph. We implemented the relations by reification to allow the addition of information about the relations, such as the original document where the relation was found.

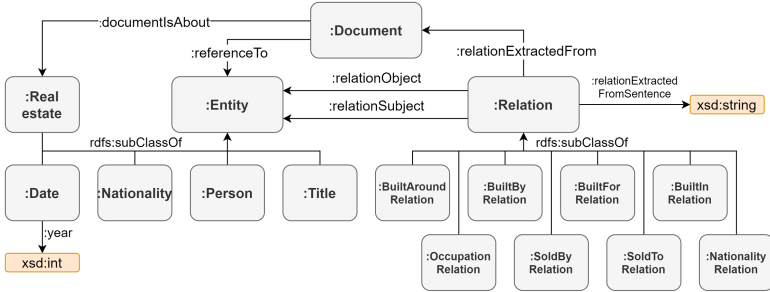


Fig. 2. Diagram representing the classes and relations of our heritage ontology

Figure 3 presents the representation within our knowledge base of a relation extracted from the text of a heritage site document. The heritage document is the one about *West Brome United Church*. The relation that we were able to extract comes from the tuple $\langle church, was\ built\ by, Simon\ Shufelt \rangle$ extracted automatically by the tool *StanfordOIE* from the sentence *This Methodist church was built around 1857 by Simon Shufelt.*

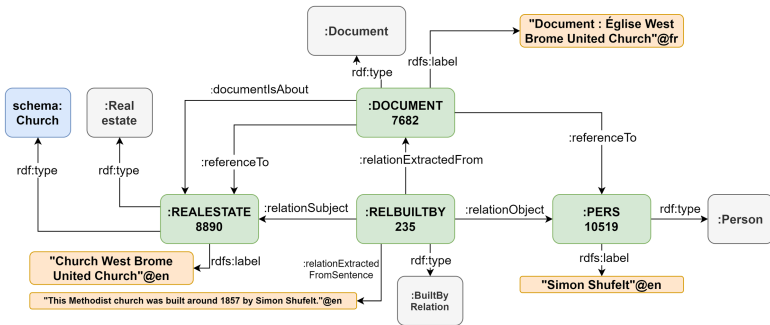


Fig. 3. Representation, within our knowledge base, of a relation extracted from the text of a heritage site document

Tables 4(a) and 4(b) show the number of instances for each *Entity* and *Relation* classes. Here we show only the main classes that were considered important in our heritage ontology.

Table 4. Number of instances per Entity Class and Relation class

Person	15 320
RealEstate	69 857
Title	1 901
Nationality	93
Date	2 186

(a) Entity class

OccupationRelation	5 715
BuiltInRelation	1 083
SoldToRelation	191
BuiltByRelation	703
BuiltForRelation	116
BuiltAroundRelation	341
SoldByRelation	355
NationalityRelation	205

(b) Relation class

Using our knowledge graph, we were able to answer several competency questions such as: *Which people have built churches and what is their profession? Which real estate assets, in relation to which heritage document, were built before 1680? What are the five most common nationalities of architects and how many architects are there for each? What are the five most common types of real estate and how many are there? What are all the relations with subjects and objects that we can find in the document : Maison Richard-Cruice?*

It is worth noting that these questions did not have any answer prior to our knowledge graph extraction and population approach.

5 Evaluation

To evaluate the impact of our supervised learning models compared to the sole use of existing open information extraction tools, we manually evaluated the extraction results (obtained in English) for various relations. In each case, we evaluated the recall, the precision and the F1 score on the set of extracted relations, before and after applying our supervised learning model. We did this for both types that we considered: *Person* and *Real estate*. We then manually counted, among the entities E_{person} and E_{undef} (entities for which no type was identified) connected to a relation (for example, objects in the relation *was built by* whose subject is of type *Real estate*), the number of true and false positives, and the number of true and false negatives. Note that a negative is an entity that was classified as E_{undef} . We applied the same approach to evaluate the learning model about the type *Real estate*. The results of these evaluations are shown in Tables 5 and 6. In each case, we can notice that the application of our supervised learning model had a positive impact on the extractions. We can see that for all relations, recall is greatly increased, with only a small loss in precision.

Table 5. Evaluation of tuples extraction according to the relation considered with optimization of the identification of entities of type *Person* with our supervised learning model

Relation considered	StanfordNER + OIE			StanfordNER + OIE + learned model		
	P	R	F1	P	R	F1
Real estate “was built by” Person	0,977	0,765	0,858	0,904	0,970	0,936
Person “built” Real estate	0,916	0,738	0,817	0,855	0,971	0,909
Real estate “was built for” Person	1	0,814	0,898	0,896	0,986	0,939
Real estate “was sold to” Person	0,929	0,591	0,722	0,809	0,864	0,835

Table 6. Evaluation of tuples extraction according to the relation considered with optimization of the identification of entities of type *Real estate* with our supervised learning model

Relation considered	StanfordNER + OIE			StanfordNER + OIE + learned model		
	P	R	F1	P	R	F1
Real estate “was built by” Person	1	0,635	0,777	0,941	0,945	0,943
Real estate “was built in” Date	1	0,327	0,493	0,940	0,939	0,940
Real estate “is in” State_or_province	1	0,076	0,141	0,676	0,316	0,431
Real estate “was built around” Date	1	0,609	0,757	0,933	0,972	0,953

6 Conclusion

In this paper, we presented our approach to extract entities and relations in French heritage descriptive texts, using mature tools that are available only for English texts. Our work offers an important contribution in the cultural heritage field where a lot of data is still only available in unstructured texts and thus not easily reusable or discoverable. Using our proposed architecture, we were able to automatically extract a knowledge base from these texts. We also presented a method to improve the recall of the open information extraction tools using supervised learning models. Our method made it possible to extract data from heritage texts and we focused on entities of type *Person* and *Real estate* but the approach could easily be reusable for other types of texts and entities. Besides heritage-specific type identification, our approach also features how tools that are efficient in English can be properly leveraged for languages with much fewer resources such as French.

In our approach, however, we did not assess the impact of the translation on the accuracy of the results. In the future, we plan to improve the level of granularity of our model to identify precisely the type of any heritage real estate (such as Church or House). This will most likely require the identification of new feature maps. We also plan to experiment with a mixed model able to identify all types at once. Finally, given that the obtained knowledge base remains in English, our future efforts will be directed towards enriching it with labels in French and linking its elements to the original French documents.

Acknowledgement. We thank the Ministry of Culture and Communication of Quebec for funding this research and for the collaboration they provided throughout its realization.

References

1. Araújo, C., Martini, R.G., Henriques, P.R., Almeida, J.J.: Annotated documents and expanded CIDOC-CRM ontology in the automatic construction of a virtual museum. In: Rocha, Á., Reis, L.P. (eds.) *Developments and Advances in Intelligent Systems and Applications*. SCI, vol. 718, pp. 91–110. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-58965-7_7
2. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data. In: Aberer, K., et al. (eds.) *ASWC/ISWC -2007*. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76298-0_52
3. Bhutani, N., Jagadish, H., Radev, D.: Nested propositions in open information extraction. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 55–64 (2016)
4. Del Corro, L., Gemulla, R.: Clausie: clause-based open information extraction. In: *Proceedings of the 22nd International Conference on World Wide Web*, pp. 355–366 (2013)
5. Doerr, M.: The CIDOC CRM, an ontological approach to schema heterogeneity. In: *Dagstuhl Seminar Proceedings* (2005)
6. Gashteovski, K., Gemulla, R., Corro, L.D.: MinIE: minimizing facts in open information extraction. (2017)
7. Gotti, F., Langlais, P.: Harnessing open information extraction for entity classification in a French corpus. In: *Canadian Conference on Artificial Intelligence*, pp. 150–161 (2016)
8. Guha, R.V., Brickley, D., Macbeth, S.: Schema.org: evolution of structured data on the web. *Commun. ACM* **59**(2), 44–51 (2016)
9. Koch, I., Freitas, N., Ribeiro, C., Lopes, C.T., da Silva, J.R.: Knowledge graph implementation of archival descriptions through CIDOC-CRM. In: Doucet, A., Isaac, A., Golub, K., Aalberg, T., Jatowt, A. (eds.) *TPDL 2019*. LNCS, vol. 11799, pp. 99–106. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30760-8_8
10. Lauscher, A., Song, Y., Gashteovski, K.: MinSciE: citation-centered open information extraction. In: *2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pp. 386–387 (2019)
11. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J.R., Bethard, S., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55–60 (2014)
12. Niccolucci, F.: Documenting archaeological science with CIDOC CRM. *Int. J. Digit. Libr.* **18**(3), 223–231 (2016). <https://doi.org/10.1007/s00799-016-0199-x>
13. Schmitz, M., Bart, R., Soderland, S., Etzioni, O.: Open language learning for information extraction. In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 523–534 (2012)
14. Song, M., Kim, W.C., Lee, D., Heo, G.E., Kang, K.Y.: PKDE4J: entity and relation extraction for public knowledge discovery. *J. Biomed. Inform.* **57**, 320–332 (2015)

15. Vrandečić, D.: Wikidata: a new platform for collaborative data collection. In: Proceedings of the 21st International Conference on World Wide Web, pp. 1063–1064 (2012)
16. Weibel, S.L., Koch, T.: The Dublin core metadata initiative. *D-lib Mag.* **6**(12), 1082–9873 (2000)



Natural Language Querying System Through Entity Enrichment

Joshua Amavi¹(✉), Mirian Halfeld Ferrari²(✉), and Nicolas Hiot^{1,2}(✉)

¹ EnnovLabs, Ennov, Paris, France
{jamavi,nhiot}@ennov.com
<https://en.ennov.com/>

² Université d'Orléans, INSA CVL, LIFO EA, Orléans, France
{mirian,nicolas.hiot}@univ-orleans.fr

Abstract. This paper focuses on a domain expert querying system over databases. It presents a solution designed for a French enterprise interested in offering a natural language interface for its clients. The approach, based on entity enrichment, aims at translating natural language queries into database queries. In this paper, the database is treated through a logical paradigm, suggesting the adaptability of our approach to different database models. The good precision of our method is shown through some preliminary experiments.

Keywords: NLI · NLP · Database query · Question answering

1 Introduction

Graph database querying systems adapted to *domain experts*, and not only to database experts, deserve great attention nowadays and become an important research topic. Query language such as SPARQL or CYPHER are powerful tools but require knowledge of the database structure in order to retrieve information. To simplify the accessibility of such databases, the research of natural language interface (NLI) to (structured) databases receives considerable attention today [15, 18]. The idea of NLI is to allow users to focus on the semantic of what they want rather than on how to retrieve it.

This paper describes a practical solution for simple natural language queries on an RDF database, developed for clients of Ennov, a French enterprise specialised in building software solutions to the life sciences industry. Our proposal focuses on the enterprise needs, *i.e.*, factoid queries concerning instances of one RDF ‘class’, but achieves good results allowing (i) to envisage its use to other domains and (ii) to extend its ideas to more complex queries. The proposal consists in translating a given natural language query (denoted as NL-query) in a database query (denoted as DB-query). In this paper, we use a logical formalism to express database and DB-queries which can be easily translated to any graph, or relational model (and thus to queries on SQL, SPARQL, etc).

In this context, the main contributions of this paper are:

- An original method, based on entities’ enrichment, for translating a NL-query into a DB-query. Indeed, entity extraction is a subtask of NLP (Natural Language Processing) and consists of identifying part of an unstructured text that represents a named entity. After identifying entities connected to a specific domain, a classification into different entity types is possible. Following this classification, some of them are merged and a set of enriched entities is obtained. DB-queries are built from this set of enriched entities.
- An approach composed by two distinct phases: a domain specific pre-processing step and a general query-generating step. The pre-processing step puts in place the general environment which guides query translation: lexicons are built (partially) from the information stored in the database, grammars and ontology mappings are set up. Query-generating algorithms classify and enrich extracted entities and then transform the obtained set of enriched entities into database queries.
- A good-precision querying system. Our approach focus on restricted and specialized domain queries which imply a relatively small vocabulary (mostly composed by people and technical terms appearing in the database instance). Our method takes advantages of this context and gives priority to the use of grammar- and lexicon- based tools. The result is an efficient and precise query translation system.
- An approach proposing a non disambiguation of the natural language queries. Indeed, instead of resolving the ambiguity problem intrinsic to natural language, we adopt a lazy approach and consider all possible interpretations, generating all possible database queries. This option avoids the expensive disambiguation process and speed up the whole performance. The same idea is used to solve ambiguity coming from the use of coordinating conjunctions.

Our querying system is available over an RDF database storing information about medical documents. The system translates a NL-query into a DB-query offering a user-friendly interface. Let us briefly introduce each of these queries together with a running example.

NL-query. Ennov’s motivation is to offer a querying system capable to allow its users to perform the so-called *facet search*, narrowing down search results by applying multiple filters. Accepted queries are those requiring information on instances of one unique RDF class (denoted here as solar-class). An allowed query selects only the solar-class instances (the nodes of the given type) via properties having the solar-class as domain or range (the in- or out-edges). For instance, supposing that *Book* is a solar-class, a query requiring book instances edited by doctor Alice on cardiology after year 2018 is an allowed query. On the contrary, a query requiring book instances edited by doctor Alice who is cardiologist, is not allowed since *is cardiologist* is not a property (and edge) of the current solar-class. If the user wants to identify doctors who are writers, then he has, firstly, to change his solar-class specification. In other terms, our query are simple queries identifying instances of one class (even if it can also renders

values concerning properties of these instances). The NL-query follows the format *Find books which...* (establishing *Book* as the solar-class), *Find doctors who...* (Doctors as the solar-class), etc.

Now, as a running example, we introduce query Q_{run} . We use it in the rest of the paper to show, step by step, how to obtain a DB-query. When talking specifically about the NL-query version we can write NLQ_{run} .

Find books with title 'Principles of Medicine' co-authored by Bob and Alice and whose price is less than 30 dollars.

DB-query. We use a logical paradigm to express RDF databases and queries. We write $Book(Anatomy)$ to express that *Anatomy* is an instance of class *Book* and $writtenBy(Anatomy, Bob)$ to express that *Anatomy* has value *Bob* for property $writtenBy$.

We briefly introduce this logic formalism (refer to [4] for some background on this aspect). An *atom* has one of the forms: (i) $P(t_1, \dots, t_n)$, where P is an n -ary predicate and t_1, \dots, t_n are terms (terms are constants or variables); (ii) \top (true) or \perp (false); (iii) $(t_1 \text{ op } \alpha_2)$, where t_1 is a term, α_2 is a term or a character string, and op is a comparison operator. A *fact* is an atom $P(u)$ where u has only constants. A *database schema* is a set of predicates \mathbb{G} and a *database instance* is a set of facts (denoted by \mathbb{D}) on \mathbb{G} .

A (*conjunctive*) query q over a given schema has the rule-form $R_0(u_0) \leftarrow R_1(u_1) \dots R_n(u_n), comp_1, \dots, comp_m$ where $n \geq 0$, R_i ($0 \leq i \leq n$) are predicate names, u_i are tuples of terms of appropriate arity and $comp_j$ ($0 \leq j \leq m$) are comparison formulas involving variables appearing in at least one tuple from u_1 to u_n . We denote $head(q)$ (respect. $body(q)$) the expression on the left hand-side (respect. right hand-side) of q . The answers for q are tuples t only with constants. For each t there exists a mapping h_t (which maps variables to constants and a constant to itself) such that $\{R_1(h_t((u_1))), \dots, R_n(h_t((u_n)))\} \subseteq \mathbb{D}$, the conjunction of all $h_t(comp_j)$ is evaluated to true (according to the usual semantic of operators op) and $h_t(u_0) = t$. In this rule-based formalism, the union is expressed by allowing more than one rule with the same head. For instance, $q(X) \leftarrow writtenBy(X, Bob)$ together with $q(X) \leftarrow editedBy(X, Bob)$ express a query looking for documents written or edited by *Bob*.

Book is the solar-class in NLQ_{run} and thus the DB-query should return the identifiers of book instances. The following conjunctive DB-query is the DBQ_{run} – it includes all the conditions imposed on the books being selected.

$$\begin{aligned} Q(x) \leftarrow & Book(x), hasTitle(x, y_1), writtenBy(x, y_2), Person(y_2), writtenBy(x, y_3), \\ & Person(y_3), hasPrice(x, y_4), (y_1 = 'Principles of Medicine'), \\ & (y_2 = :bob), (y_3 = :alice), (y_4 < 30). \end{aligned}$$

□

Paper Organization. Our approach is depicted in Sects. 2 and 3 while implementation and testing results are presented in Sect. 4. Sect. 5 concludes the paper with some related work and perspectives.

2 Entity Extraction and Enrichment

In this paper we define a *simple entity* as the tuple $E = (\mathcal{V}, \mathcal{T}, m)$ where \mathcal{V} and \mathcal{T} are lists containing values and lexical types, respectively, and m is a mapping such that $\forall v \in \mathcal{V}, \exists T \subseteq \mathcal{T}, m(v) \rightarrow T$.

Indeed, during the entity extraction phase, ambiguity, an inherent problem in many steps of natural language processing, exists: it concerns the type (*e.g.* *Paris* can refer to a city or a person) or the value (*e.g.*, several people in the database have the same name). Generally, we seek to eliminate this ambiguity by keeping only the most likely solution. Such a solution may introduce contradiction *w.r.t.* the text semantics. In our approach we explicitly reveal ambiguity (a value may be associated to different types) and we keep track of multiple interpretations during this extraction step, a convenient solution for querying, if we consider the situations where ambiguity can be represented by an *OR* connective.

Entity Extraction (EE) or Named Entity Recognition (NER) is a subtask of NLP and consists of identifying part of an unstructured text that represents a named entity. This task can be performed either by grammar-based techniques or by a statistical models such machine learning (refer to [11] for a complete introduction in the domain). Statistical approaches are widely used in the industry because they offer good results with the latest research and the work of giants like Google, Facebook or IBM. However, these approaches mainly require a lot of data to get good results, implying high costs. More conventional grammar-based methods are very useful for dealing with small data sets.

Entity Extraction. Our proposal consists in applying different grammar- or lexicon-based methods together in order to extract simple entities from a given NL-query. The combination of their results allow us to improve entity extraction. The initial parsing step is followed by two different entity extraction methods. One consists of looking up on dictionaries (lexicons) for qualifying an entity. The other is based on local grammars. Notice that the dependency tree resulting from the parsing phase may be used for guiding entity extraction with local grammars.

Parsing. In our approach, tokenization (*i.e.*, determine the words and punctuation), part-of-speech (POS) tagging (*i.e.*, determine the role of the word in a sentence), lemmatization (*i.e.*, determine word canonical form), stematization (*i.e.*, strip word suffixes and prefixes) and dependence analysis are achieved by SpaCy [1] built on a convolutional neural network (CNN) [9] learned from a generic English corpus [16]. The dependency tree produced by SpaCy [8] guides different choices in some of the following steps of our approach. Fig. 1 illustrates part of the dependency tree for NLQ_{run} .

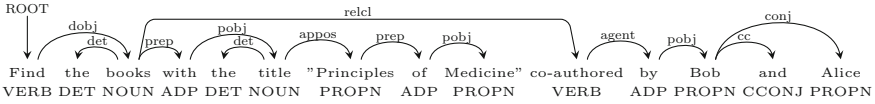


Fig. 1. Dependency tree and POS tagging

LEXICONS			INVERSE INDEX	
<i>EntityName</i>	<i>Lexemes</i>		Pointer to	Index Entry (word from text)
:alice	<i>Alice, Wonderful, Wonderful Alice</i>		:alice	<i>Alice</i>
:bob	<i>Sponge, Bob, Sponge Bob</i>		:alice	<i>Alice Wonderful</i>
...			...	
author	<i>co-authored, written by, created by</i>		:bob	<i>Bob</i>
...			author	<i>co-authored</i>
lt	<i>less than</i>		lt	<i>less than</i>
...			:alice	<i>Wonderful</i>
...			...	

Ontology-Mappings					Operator Dictionary	
<i>EntityName</i>	<i>LexType</i>	<i>DBType</i>	<i>DBType</i>	Predicate	<i>EntityName</i>	Comparator
:alice	Person	Person	Person	Person (X)	lt	<
author	Context	Author	Author	writtenBy (., X)	leq	≤
:bob	Person	Person	Book	Book (X)	gt	>
title	Context	Title	Title	hasTitle (., X)	geq	≥
:alice	Doctor	MedicalDoc	MedicalDoc	Doctor (X)

Fig. 2. Auxiliary structures built in the pre-processing phase

Lexicons. When, as in our case, we deal with entity extraction founded on a relatively small database, one could envisage to verify, for each value in the database, whether it appears in the text. However, to improve performance and ensure low coupling, our option consists in building lexicons (*i.e.*, lookup tables) from the database and then in using them as dictionaries containing a summary of the database instance.

Figure 2 illustrates an inverse index pointing to lexicons. It helps finding *EntityValue* quickly. For instance, if *Wonderful Alice* is the string found in the text, the index allows us to find how to refer to it, *i.e.*, with its *EntityValue*, :alice. With *EntityValue* we have access to information stored in auxiliary structures denoted here by **Ontology-Mappings**. In order to simplify notations, we consider the existence of three functions that render information over **Ontology-Mappings**. They are: (1) *lexT*: given *EntityValue*, the function renders its lexical type indicating entity’s lexical role, detected during the lexical analysis; (2) *dbT*: given *EntityValue*, the function renders its database type which indicates the semantic associated to the entity in our RDF database and (3) *getPred*: given the database type, the function renders the associated predicate which is the one we should use in the DB-query. Moreover, the arity of predicates is indicated together with the argument position reserved for an entity having this database type. According to the example in Fig. 2, we have: *lexT*(:alice) = *Person*, *dbT*(:alice) = *Person* and *getPred*(*Person*) = *Person*(X_{person}). Similarly, lexeme *co-authored by* refers to the entity value *author* and we obtain *lexT*(*author*) = *Context*, *dbT*(*author*) = *Author* and *getPred*(*Author*) = *writtenBy*(X, Y_{author}).

Local Grammar. There is no sense in storing possible values for attributes associated to huge domains. Dates, general numerical attributes (*e.g.*, prices, weight, etc) or even publication or section titles are not stored in our database. Entity extraction, for such cases, is based on local grammars. Currently we have designed 15 local grammars as a support for our entity extraction mechanism.

Entity	EntityValue	lexT
E_1	book	Context
E_2	title	Context
E_3	author	Context
E_4	:bob	Person
E_5	:alice	Person
E_6	price	Context
E_7	lt	Operator
E_8	'Princ. of Medicine'	Text
E_9	30	Number

Fig. 3. Simple entities extracted from NLQ_{run}

Example 1. We show our NLQ_{run} with expressions in boxes indicating the entities detected after the extraction step. Figure 3 summarizes obtained entities: E_1 – E_7 via Lexicons while E_8 – E_9 are detected by local grammars. Figure 3 does not represent set notation to avoid figure overload. Here, each entity has only one value and each value is associate to a singleton.

Find the books₁ with the title₂ "Principles of Medicine"_s
co-authored by₃ Bob₄ and Alice₅ and whose price₆ is less than₇ 30₉
dollars. □

Once we have extracted simple entities, we classify them into categories in order to decide about their fusion or not. Our goal is to build enriched entities which concentrate information initially available in the detected simple entities. An enriched entity is a first-class citizen which will guide the construction of DB-queries while simple entities are auxiliary ones considered as second-class citizens and committed to integrate the enriched entity.

Entity Enrichment. An *enriched entity* is a relation. It can thus be seen as table (as in the relational model) with schema $E_e[EntityValue, DBType, LexType, op]$. The entity E_e itself is a relation instance, *i.e.*, a set of functions (tuples) associating each attribute in the schema to a value. Thus, each tuple maps: (i) *EntityValue* to the value v of an entity as represented in a Lexicon (Fig. 3), (ii) *DBType* and *LexType* to $dbT(v)$ and $lexT(v)$, respectively (Fig. 2), and (iii) *op* to a comparison operation indicating the kind of comparison imposed on the entity value. By default, the comparison operation *op* is *equal to*.

On the other hand, we distinguish the following classes of simple entities:

- A *reference entity* is the one chosen to evolve, *i.e.*, to be transformed into an enriched entity. It corresponds to an instance value in a database (*e.g.*, people names, document titles, dates, etc).
- An *operator entity* E_{cp} represents words which indicate that another (reference) entity $E = (\mathcal{V}, \mathcal{T}, m)$ is constrained by a comparison condition. A connection between E and E_{cp} in the dependency tree determines E as the reference entity. Then, E evolves to an enriched entity E_e such that for each $v \in \mathcal{V}$ we have $(v, dbT(v), lexT(v), op)$ in E_e where *op* is defined by E_{cp} , according to an available dictionary (see example in Fig. 2). Notice also that

op corresponds to the operator used in the DB-query (Sect. 1) where *comp* atoms have the general format $(t_1 \text{ op } \alpha_2)$. In Example 1, expression *less than* is an operator entity having 30 as its reference entity.

- A *context entity* E_C gives information about the type of another (reference) entity. Once again, the dependency tree obtained during the parsing, determines the reference entity E which evolves to a new enriched entity E_e according to Algorithm 1. In Example 1, *price* is a context entity and 30 as its reference. Similarly, *title* is a context entity and *Principles of Medicine* its reference (Fig. 1).

We remark that the first For-loop of Algorithm 1 transforms a simple entity into an enriched one. Given a simple entity E , $extend(E)$ is the relation instance obtained by converting E into its extended counterpart. Notice that the entity evolution process starts with a simple entity which becomes an enriched entity, but such an enriched entity can continue evolving.

Example 2. From Fig. 3 and our auxiliary structures (partially depicted in Fig. 2) we obtain the following enriched entities:

$$\begin{aligned}
 E_{e0} &= \{(book, Book, Context, =), \}; \\
 E_{e1} &= \{('Princ. of Medicine', Text, Text, =), \\
 &\quad ('Princ. of Medicine', Title, Context, =)\} \\
 E_{e2} &= \{(:bob, Person, Person, =), (:bob, Author, Context, =)\} \\
 E_{e3} &= \{(:alice, Person, Person, =), (:alice, Author, Context, =)\} \\
 E_{e4} &= \{(30, Number, Number, <), (30, Price, Context, <)\}.
 \end{aligned}$$

Algorithm 1: contextEnrichment

Input: $E_C = (\mathcal{V}_C, \mathcal{T}_C, m_C)$ and $E = (\mathcal{V}, \mathcal{T}, m)$

Output: E_e : an instance over schema $E_e[EntityValue, DBType, LexType, op]$

- 1: **for all** $v \in \mathcal{V}$ **do**
 - 2: Insert $(v, dbT(v), lexT(v), op)$ in E_e ;
 - 3: **for all** $u \in \mathcal{V}_C$ **do**
 - 4: **for all** $v \in \mathcal{V}$ **do**
 - 5: Insert $(v, dbT(u), lexT(u), op)$ in E_e ;
-

All entities are enriched ones. E_{e1} results from the integration of context entity E_2 into E_8 , E_{e2} results from integration of E_3 into E_4 and E_{e3} results from integration of E_3 into E_5 . Notice that coordinating conjunction *and* in NLQ_{run} implies the existence of these two latter independent enriched entities. E_{e4} results from the integration of operator entity E_7 to E_9 . Entity E_{e0} is just the enriched version of E_1 . It corresponds to the solar-class. \square

Now, a NL-query may include multiple conditions (or filters) connected by coordinating conjunctions. Our current version deals only with *and* and *or*, even

if we intend to extend this initial proposal to more complex coordinating conjunctions such as *nor*, *for*, etc. Coordinating conjunctions are expressed through logical formulas which guide the construction of the DB-query, by specifying: (i) the kind of atoms *comp* it will have and (ii) whether the query is defined by one of several rules. Taking into account coordinating conjunctions implies entity enrichment. Let $E_1 = (\mathcal{T}_1, \mathcal{V}_1, m_1)$ and $E_2 = (\mathcal{T}_2, \mathcal{V}_2, m_2)$ be two simple entities having the same LexType. If, in the query text, these two entities are connected by an *or*, they are merged, forming a new enriched entity composed by $extend(E_1) \cup extend(E_2)$. The original entities do not exist any more. Otherwise, if in the text, these two entities are connected by an *and*, they are kept as independent ones.

Example 3. In Example 2, $E_{e2} = \{(:bob, Person, Person, =), (:bob, Author, Context, =)\}$ and $E_{e3} = \{(:alice, Person, Person, =), (:alice, Author, Context, =)\}$ are independent entities. When considering NLQ_{run} these entities do not merge because the coordinating conjunction is an *and*. If we change the sentence to '*written by Bob or Alice*', entities are merged resulting in:

$$E_{enew} = \{(:bob, Person, Person, =), (:bob, Author, Context, =), (:alice, Person, Person, =), (:alice, Author, Context, =)\}.$$

□

Queries may have multiple coordinating conjunctions as illustrate in sentence *written by Alice or Bob and Charlie* and, in this case, its interpretation (due to the ambiguity of natural language) can vary, resulting in the logic formula $\mathcal{F}_1 \equiv (X = :alice) \vee ((X = :bob) \wedge (X = :charlie))$ or in the formula $\mathcal{F}_2 \equiv ((X = :alice) \vee (X = :bob)) \wedge (X = :charlie)$. To avoid erroneous query answers, one may envisage to take into account all the alternative interpretations, or to give choices to the user. In our approach we do not plan interactions with the user and thus, we propose to consider a larger interpretation, *i.e.*, to overcome ambiguity by replacing a mixed *and-or* sentence by an *only-or* sentence. Thus, let \mathcal{F} be the logic formula obtained from a sentence with coordinate conjunctions. If \mathcal{F} contains both \vee and \wedge , then we replace it by the formula \mathcal{F}' composed only of \vee . The idea is based on the fact that any answer satisfying \mathcal{F} also satisfies \mathcal{F}' . In that way, when multiple coordinate conjunctions are present, the DB-query will be represented by multiple rules with the same head.

3 Building DB-Queries from Enriched Entities

Once our NL-query is analysed and all enriched entities are completed, the DB-query is generated by Algorithm 2. The algorithm starts by considering entity E_{e0} (line 3) which has a special role since it specifies the solar-class, *i.e.*, the class on which the query focuses. The query is initialized with a body composed by one unique atom over the predicate associated to the solar-class. Notice the use of function `bAt` which is responsible for building an atom for the query being constructed. The predicate symbol to be used in the construction of an atom is

found via the value of attribute $DBType$ in E_{e0} – which is then used as an input for function getPred . In Example 2, $Book$ is the value of attribute $DBType$ in E_{e0} and the name of the associated unary predicate. Atom A_0 in our case is $Book(x)$. Notice that Algorithm 2 builds only queries whose answers are books’ identifiers (*i.e.*, instantiations of x). Our initial query is thus $q(x) \leftarrow Book(x)$.

Lines 11 to 15 of Algorithm 2 consider entities E_e enriched with a context entity. If in E_e there are more than one tuple t for which the value of attribute $lexType$ is “Context”, then E_e is an entity obtained after taking into account coordinating conjunction *or*. Each tuple t having value “Context” for attribute $lexType$ has to be grouped together with the tuple t' representing its reference. On line 14, Algorithm 2 groups each t with another $t' \in E_e$ having the same value for attribute $EntityValue$. From information in t and t' we build a list l , added to set $Parts$. Each $l \in Parts$ is a list of atoms to be added to the body of the query under construction. Notice that Algorithm 2 divides E_e ’s tuples into parts (or lists). Each list in $Parts$ generate a new distinct query with the same *head*. Indeed, on line 24, in the for-loop, each list l is used to create a new query q' – continuing the construction of a query q already in \mathcal{Q} . If there are more than one list l in $Parts$, there will be more than one query q' .

From Example 2, E_{e2} has two tuples. Let t_1 be the first tuple for which $\text{getPred}(t_1(DBType)) = \text{getPred}(Person) = Person$ and t_2 be the second one for which $\text{getPred}(t_2(DBType)) = \text{getPred}(Author) = writtenBy$. The function bAt can be used to build atoms that will be added to $body(q)$. Notice that bAt also takes into account information concerning positions marked as the place for the entity value in the atom being built. Thus, the new variable y representing the entity is placed accordingly. In binary predicates x is always the other variable. Atoms $comp$ may associate a value to variable y . Thus, on line 14 list $\langle Person(y_2), writtenBy(x, y_2), (y_2 = :bob) \rangle$ is added to $Parts$ and on line 24 the query being built is $q(x) \leftarrow Book(x), Person(y_2), writtenBy(x, y_2), (y_2 = :bob)$. The result obtained with entity E_{e3} is similar. However, entities E_{e1} and E_{e4} are treated in a different way since their lexical types are *Text* and *Number*, respectively. These entities are treated on lines 17–19. For instance, E_{e1} gives rise to list $\langle hasTitle(x, y_1), (y_1 = \text{“Princ. of Medicine”}) \rangle$. After considering all entities, Algorithm 2 returns set \mathcal{Q} with the following DB-query:

$$\begin{aligned} q(x) \leftarrow & Book(x), hasTitle(x, y_1), Person(y_2), \\ & writtenBy(x, y_2), Person(y_3), writtenBy(x, y_3), asPrice(x, y_4), \\ & (y_1 = \text{“Princ. of Medicine”}), (y_2 = :bob), (y_3 = :alice), (y_4 < 30) \end{aligned}$$

However, if we consider E_{enew} of Example 3, Algorithm 2 (lines 11 to 15) produces two lists from the same entity, namely, $l_1 = \langle Person(y_2), writtenBy(x, y_2), (y_2 = :bob) \rangle$ and $l_2 = \langle Person(y_3), writtenBy(x, y_3), (y_3 = :alice) \rangle$. Then, on line 24, each list is considered separately and the query q is replaced by two new queries. At the end, \mathcal{Q} returns a DB-query composed by two rules:

Algorithm 2: EntitiesToQueries

Input: \mathcal{E} an enriched entity set $\{E_{e0}, E_{e1}, \dots\}$
Output: \mathcal{Q} a set of query rules, *i.e.*, the DB-query with one or more rules

- 1: $\mathcal{Q} := \emptyset$
- 2: **for all** enriched entity E in \mathcal{E} **do**
- 3: **if** E is E_{e0} **then**
- 4: $\{(eval, dbTval, lTval, opval)\} := E$
- 5: $A_0 := \mathbf{bAt}(dbTval, x)$ // Build the first atom for the query's body
- 6: $\mathcal{Q} := \{q(x) \leftarrow A_0\}$
- 7: **else**
- 8: $Parts := \emptyset$ // Set of list of atoms. Each $l \in Parts$ is a list of atoms whose conjunction should be added to query's body.
- 9: $E' := \emptyset$ // Set storing E 's tuples already treated
- 10: // Treatment of entities enriched with a context
- 11: **for all** tuple $t = (eval, dbTval, \text{"Context"}, opval)$ in E **do**
- 12: Let $t' \in E$, s.t $t' = (eval, dbTval', lTval', =)$ and $lTval' \neq \text{"Context"}$
- 13: $y := GetNewVar()$
- 14: $Parts := Parts \cup \{(\mathbf{bAt}(dbTval', y), \mathbf{bAt}(dbTval, y), \mathbf{bAtOp}(eval, y, opval))\}$
- 15: $E' := E' \cup \{t', t\}$
- 16: // Treatment of enriched entities without tuples where $lTval = \text{"Context"}$
- 17: **for all** tuple $t = (eval, dbTval, lTval, opval)$ in $(E \setminus E')$ **do**
- 18: $y := GetNewVar()$
- 19: $Parts := Parts \cup \{(\mathbf{bAt}(dbTval, y), \mathbf{bAtOp}(eval, y, opval))\}$
- 20:
- 21: $\mathcal{Q}' := \emptyset$
- 22: **for all** query $q \in \mathcal{Q}$ **do**
- 23: **for all** list $l \in Parts$ **do**
- 24: $q' = BuildNewQuery(q, l)$
- 25: $\mathcal{Q}' := \mathcal{Q}' \cup \{q'\}$
- 26: $\mathcal{Q} := \mathcal{Q}'$
- 27: **return** \mathcal{Q}

$$q(x) \leftarrow Book(x), hasTitle(x, y_1), Person(y_2), (y_1 = \text{"Princ. of Medicine"}),$$

$$writtenBy(x, y_2), hasPrice(x, y_4), (y_2 = :bob), (y_4 < 30)$$

$$q(x) \leftarrow Book(x), hasTitle(x, y_1), Person(y_3), (y_1 = \text{"Princ. of Medicine"}),$$

$$writtenBy(x, y_3), hasPrice(x, y_4), (y_3 = :alice), (y_4 < 30)$$

Finally, consider $E_{new2} = \{(:bob, Person, Person, =), (:bob, Author, Context, =), (:bob, Editor, Context, =)\}$. The resulting lists on line 14 are $l_1 = \langle Person(y_2), writtenBy(x, y_2), (y_2 = :bob) \rangle$ and $l_2 = \langle Person(y_3), editedBy(x, y_3), (y_3 = :bob) \rangle$ and \mathcal{Q} also returns a DB-query composed by two rules. Here we are looking for *books edited or written by Bob*.

Thus, queries can be directly generated from enriched entities. Currently we only deal with conjunctive queries – easily translated to SQL or SPARQL.

4 Implementation and Experimental Results

In order to validate our system, we implemented it in the form of a pipeline which allows us to divide the separate stages and explore various combinations. For lexicon-based entity extraction, Apache SolR [2] is used with its text tagger, an inverted index and n -gram algorithm [12]. It allows lexemes detection even with typographic errors. We also use a combination of hand-written grammars together with a Facebook project called Duckling [3] which provides powerful tools for extracting entities such as numbers or dates. Each extraction step is performed independently and simple entities are defined by taking into account all different methods. In particular, if several approaches identify entities in the same place (but not necessarily with the same bounds), we keep only the entity resulting from the union of the overlapping entities to represent the ambiguity. To implement this pipeline and link each step, we use the RASA NLU framework [6] in combination with SpaCy for the parsing phase. Notice that the pre-processing part is based on generic grammars and lexicons. Some lexicons are generated automatically by considering values in the RDF database (*e.g.*, first and last name for *Person*). Hand-written lexicons such as those for operators and contexts concerning dates (*e.g.*, application, archive, creation, expiration) or persons (*e.g.*, author, signatory) are also used. Partial matches are managed using multiple lexemes when possible (*e.g.* *create by*, *create with*, *create*, ...).

We conduct our preliminary experiments on an RDF database concerning medical publications. The database has 66 classes (possible candidates for a query solar-class) with a total of 29327 class instances. In our tests, about 10 classes have been used as solar-classes. These tests have considered entity extraction and enrichment phases. Ambiguity has not been tested and thus we only take into account one value per entity. The evaluation is done by analysing the obtained enriched entities. Figure 4 shows the results of our experiments on a set of 113 NL-queries (varying number of *and* and *or*). It summarizes the precision, recall, f1-score, and the weighted mean (weighted by support) obtained for each dbT on the NL-queries set. As our system is implemented as a pipeline, we intend to perform tests step by step, in order to identify the impact of each step.

Our precision is good, indicating that most of our detected entities are the expected ones. This is clearly a consequence of the effective use of lexicons and grammars. For recall, our results are not bad, but weaker than our precision, indicating that some entities are not detected. Lower precision on some dbT like *creation_date* or *doc_author* is partially explained because ambiguity is not taken into account in our experiment, but entities giving rise to these

<i>DBT</i> type	precision	recall	f1-score	support
solar_class	1.00	0.62	0.77	82
application_date	1.00	0.62	0.76	13
archive_date	0.50	0.67	0.57	3
creation_date	0.60	0.60	0.60	5
expiration_date	1.00	0.50	0.67	2
customers	1.00	0.60	0.75	5
department	1.00	0.11	0.20	9
sector	1.00	0.95	0.98	21
doc_author	0.77	0.63	0.70	38
doc_signatory	0.90	0.86	0.88	21
doc_status	1.00	0.50	0.67	6
doc_unit	1.00	0.27	0.43	11
...
Weighted avg.	0.86	0.59	0.67	295

Fig. 4. Results on enriched entities

types are enriched with a similar context and associated with both types. A similar issue occurs with the recall for *department* and *doc_unit*. In our test database they are semantically close. So, we have significant overlap on the two lexicons (a lot of lexemes are shared, adding ambiguity to the type). Our current work consists in improving lexicons for context detection, in particular those generated automatically.

5 Related Work and Concluding Remarks

Recently, NLI has been widely discussed in the literature. Some work focuses on augmenting the expression power of queries while others on domain-independence. For instance, in [18] authors propose the use of binary templates rather than semantic parses to better understand complex queries while [15] proposes a cross-domain NLI with a general propose question tagging strategy. Several work (such as in [5, 7, 19]) consider RDF question/answering (QA). Aggregate queries are considered in [10]: the authors propose a method to automatically identify the aggregation and transform it into a SPARQL aggregate statement. Methods used vary a lot. In [13, 17, 18] authors base their approach on NLP techniques with entity extraction and grammars, while in [14, 15] they use neural networks.

The paper presents a method where enriched entities allow us to translate NL-queries into DB-queries. The use of a logical paradigm to deal with databases shows that our method can be adapted to different data models. Our approach is divided into a domain-dependent pre-processing and domain-independent query generation phases. The first step, responsible for building lexicons, grammar-tools and ontology mappings, also sets up general propose tools which can be considered as domain-independent (*e.g.*, grammars for date recognition). The second step of our method can be applied on any domain, provided the ontology mappings are set up. This division allows us to deal with possible extensions and improvements separately. We are currently considering extensions of Algorithm 2 in order to deal with queries on more than one solar-class or aggregate queries. We also plan to extend entity extraction by including alternative approaches such as machine learning to complete grammars (*e.g.* for title identification).

References

1. <https://spacy.io>
2. <https://lucene.apache.org/solr/>
3. <https://github.com/facebook/duckling>
4. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases, vol. 8. Addison-Wesley, Reading (1995)
5. Amsterdamer, Y., Kukliansky, A., Milo, T.: A natural language interface for querying general and individual knowledge. Proc. VLDB Endow. **8**(12), 1430–1441 (2015)
6. Bocklisch, T., Faulkner, J., Pawlowski, N., Nichol, A.: Rasa: open source language understanding and dialogue management. [arXiv:1712.05181](https://arxiv.org/abs/1712.05181) [cs] (2017)

7. Fader, A., Zettlemoyer, L., Etzioni, O.: Open question answering over curated and extracted knowledge bases. In: The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2014, New York, NY, USA, 24–27 August 2014, pp. 1156–1165. ACM (2014)
8. Honnibal, M., Johnson, M.: An improved non-monotonic transition system for dependency parsing. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1373–1378. Association for Computational Linguistics, Lisbon, Portugal, September 2015
9. Honnibal, M., Montani, I.: spacy 2: natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing **7**(1) (2017, to appear)
10. Hu, X., Dang, D., Yao, Y., Ye, L.: Natural language aggregate query over RDF data. *Inf. Sci.* **454–455**, 363–381 (2018)
11. Jurafsky, D., Martin, J.H.: *Speech and Language Processing*, 3rd draft edn. Stanford Univ. (2019). <https://web.stanford.edu/~jurafsky/slp3/>
12. Kim, J.Y., Shawe-Taylor, J.: Fast string matching using an n-gram algorithm. *Software Pract. Exper.* **24**(1), 79–88 (1994)
13. Steinmetz, N., Arning, A.K., Sattler, K.U.: From natural language questions to SPARQL queries: a pattern-based approach. *BTW* **2019**, 289–308 (2019)
14. Utama, P., et al.: An end-to-end neural natural language interface for databases. arXiv preprint [arXiv:1804.00401](https://arxiv.org/abs/1804.00401) (2018)
15. Wang, W.: A cross-domain natural language interface to databases using adversarial text method. In: *Database*, vol. 1, p. 4 (2019)
16. Weischedel, R., et al.: OntoNotes: a large training corpus for enhanced processing. In: *Handbook of Natural Lang. Processing and Machine Translation*, p. 59 (2011)
17. Zafar, H., Napolitano, G., Lehmann, J.: Formal query generation for question answering over knowledge bases. In: Gangemi, A., et al. (eds.) *ESWC 2018. LNCS*, vol. 10843, pp. 714–728. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_46
18. Zheng, W., Yu, J.X., Zou, L., Cheng, H.: Question answering over knowledge graphs: question understanding via template decomposition. *Proc. VLDB Endowment* **11**, 1373–1386 (2018)
19. Zou, L., Huang, R., Wang, H., Yu, J.X., He, W., Zhao, D.: Natural language question answering over RDF: a graph data driven approach. In: *International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, June 22–27, 2014*, pp. 313–324. ACM (2014)



Public Riots in Twitter: Domain-Based Event Filtering During Civil Unrest

Arturo Oncevay¹, Marco Sobrevilla³, Hugo Alatrística-Salas^{2(✉)},
and Andrés Melgar¹

¹ Pontificia Universidad Católica del Perú, Lima, Peru
{arturo.oncevay, amelgar}@pucp.edu.pe

² Universidad del Pacífico, Lima, Peru
h.alatristas@up.edu.pe

³ Universidade de São Paulo, São Paulo, Brazil
msobrevillac@usp.br

Abstract. Civil unrest is public manifestations, where people demonstrate their position for different causes. Sometimes, violent events or riots are unleashed in this kind of events, and these can be revealed from tweets posted by involved people. This study describes a methodology to detect riots within the time of a protest to identify potential adverse developments from tweets. Using two own datasets related to a violent and non-violent protest in Peru, we applied temporal clustering to obtain events and identify a tweet headline per cluster. We then extracted relevant terms for the scoring and ranking process using a different domain and contrast corpus built from different sources. Finally, we filtered the relevant events for the violence domain by using a contrast evaluation between the two datasets. The obtained results highlight the adequacy of the proposed approach.

Keywords: Riot · Social media analysis · Event detection · Clustering

1 Introduction

Social networks have allowed people to communicate their messages through the internet. This effect has also reached public activism in many groups to organize, mobilize, and execute public manifestations to express a position in favor of against an issue (political, economic, or social). In that way, when a protest or civil unrest is summoned, it is possible to perceive two main events progressing: 1) a real-world event in the streets, and 2) a massive amount of publications in the social networks, specifically on Twitter.

This context draws the attention of social and computational researches because they are allowed to analyze a real-time manifestation through text posts from protesters in the street or people discussing the event. Recent studies have focused in different review aspects during civil unrest, such as modeling propagation effect of protests throughout Twitter [12], analyzing the level of influence

of users [16,22], detecting topics in relevant events during social upheavals [10], or predicting potential future protests through mining posts in social networks and open-source data [4,19].

Furthermore, there is a widespread issue during a protest: violent events. Despite good intentions, some manifestations end up as riots, due to fights caused by the parties involved. These manifestations are a menace to public safety, so it is essential to know when a public manifestation could develop violent confrontations. The study of violent content in social media has been widely studied from the computer science perspective, such as the prediction of negative emotions related to crime [8], the classification of verbal aggression in posts [9] from Twitter, the identification and measurement of participation in peaceful and violent political protest events from social media datasets [3], or the detection of disruptive events that threaten social safety and security through Twitter messages [2]. In this context, this study explores a method to detect riots or violent events using tweets representing a protest to identify these negative situations alongside the civil unrest. This goal could be possible by the application of techniques, such as clustering for event detection and a process of term extraction and scoring for the selection of the cluster of interests.

On one side, there are different techniques proposed in the Twitter event detection task. A comparison was performed in [1], where methods were fit in document-clustering and term-clustering categories, with different approaches such as probabilistic models (e.g. LDA [5]) or classical document-pivot topic detection (e.g. FSD [18]). In the same way, applying clustering methods for event/topic detection in the streaming corpus has had great results in the state-of-the-art. For instance, in SNOW 2014 competition [17], the best result for detecting relevant topics was obtained using hierarchical clustering [11] at defined time intervals.

On the other side, term extraction and scoring processes are relevant in the pursuit of automating keywords extraction and ontology learning from a textual corpus. The identification of keywords has a very intensive use of POS-tagging and grammatical construction of phrases (e.g., CFinder [13] or Text2Onto [7] methods). Moreover, taxonomy or ontology learning from selected keywords requires confident term filters, which are based on the domain-related corpus (e.g., using a contrast corpus in the ATCT framework [15] or applying Formal Concept Analysis [6]).

In this work, the methodology applied is described in Sect. 2. After that, Sect. 3 presents the experiments (for each phase, until the validation process) and discusses the results obtained with the acquired datasets. Finally, conclusions and future work can be seen in Sect. 4.

2 Proposed Methodology

The proposed four-phase methodology is the following: 1) corpus acquisition from tweets and domain-related publications; 2) event detection with clustering; 3) term extraction and scoring for the clusters, and 4) contrast evaluation for filtering the relevant events. Fig. 1 resumes the steps mentioned before.

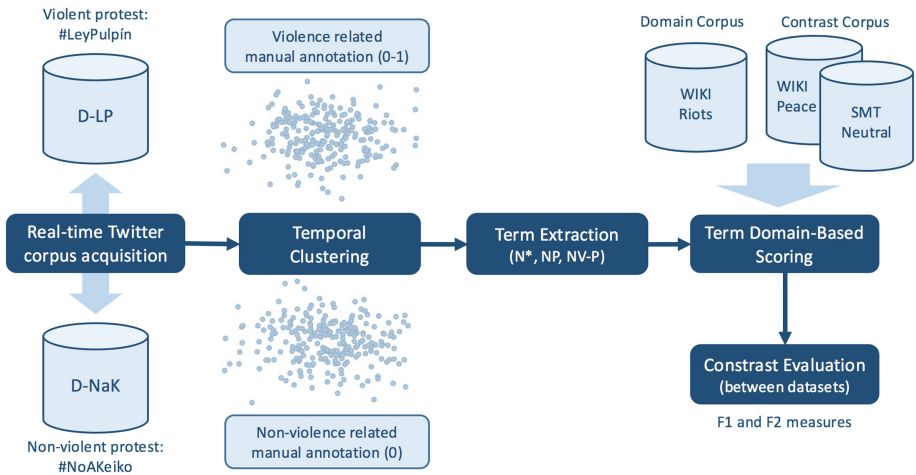


Fig. 1. Proposed methodology for the study

2.1 Corpus Acquisition and Case Studies

The primary source for the corpus acquisition is Twitter. With the Streaming API¹, it was possible to retrieve publications during real-world protest events in Perú. In this study, it was decided to contrast two civil unrest study cases. They were convened peacefully at the beginning but had different violence-related developments. The acquisition was performed during the most relevant event (when most of the publications were made). Details of the acquisition process and content of each dataset are published in a project page².

1. **#LeyPulpín dataset (D-LP)** This is a public manifestation against the right labor law with benefit cuts for young formal workers. This dataset is the violent case study acquired with a high occurrence of riots and clashes between the police and activists. 9,322 tweets; 192,765 tokens. Date: January 15th, 2015. Time: 19:30–23:30 (4 h).
2. **#NoAKeiko dataset (D-NaK)** The second case is a political manifestation in the context of the presidential general election in 2016, and despite previous upheavals, the protest was peaceful without violent incidents reported. 4,165 tweets; 72,223 tokens. Date: March 11th, 2016. Time: 21:00–24:00 (3 h).

On the other side, for the term filtering step, it was necessary to get a domain (violence-related) corpus and two contrast corpus (with opposite and neutral topics). They were extracted from Spanish Wikipedia, and we used another large Spanish corpus also:

¹ <https://dev.twitter.com/streaming/overview>.

² <http://inform.pucp.edu.pe/~grpiaa/?p=920>.

1. **Domain Corpus (WIKI-Riots)** 97 documents; 323,361 tokens
2. **Contrast Corpus (WIKI-Peace)** 47 documents; 38,458 tokens
3. **Large Neutral Corpus (SMT-Neutral)** 1 document from the Europarl Corpus in Spanish [14]; 2,123,835 sentences; 54,806,927 tokens.

This corpus may be small in comparison to other experiments in the term filtering domain. However, that is due to the specific domain (riots and violence in protests).

2.2 Clustering for Event Detection

The applied method in this step is the Aggressive Filtering and Hierarchical Tweet Clustering algorithm proposed by Ifrim et al. [11]. This algorithm is an event detection approach based on tweet-clustering, which has benefits against the classic term-clustering. The strategy is considering a tweet as a central unit for the content processing, so it is not necessary to re-create a headline topic from separate terms. Also, it is noteworthy that a time-window parameter could be set to analyze clusters grouped by different time intervals. After the basic clustering process, a tweeted headline per cluster is obtained, which is the input used for the next steps.

2.3 Term Extraction

For the term extraction step, we used the Tree-Tagger tool in Spanish [20] to retrieve potential keywords in tweets headlines according to their POS-tags. We considered three different approaches:

1. **Nouns (N*)**: we extracted nouns and complex nouns (N*) according the ATCT framework procedure [15].
2. **Noun Phrases (NP)**: because of Spanish grammar peculiarity, we adopted the proposition in the CFinder method [13] and extract NP (ADJ* + N* + ADJ*), but also considered the construction N+PREP+N as a complex N*.
3. **Noun and Verbal Phrases (NV-P)**: following the proposal of Cimiano et al. [6], we added some VP constructions (Verbs only) to the previous group due to their relevance in ontology learning processes.

2.4 Term Domain-Based Scoring

We adopted three of four-term scores defined in [21] and applied in the ACTC framework [15].

1. **Domain Pertinence (DP)**: “It is a measure that is used to acquire terms that are representative for a specific domain corpus, and not for other contrastive corpora” [15].
2. **Domain Consensus (DC)**: “It is used to determine if a term frequently appears across the domain corpus documents” [15].

3. **Domain Score (D-score):** It is obtained using the DP and DC values per term. Also, it is considered a weight for each metric (α and β , which sum to 1), and the values are normalized with the max value per measure in the corpus [15].

For each tweet headline, we sum up the D-score of the terms extracted to determine a ranking of clusters, considering the closeness of their headlines to the corpus domain, and the opposite for the contrastive corpus. In our case, we did not consider filtering out terms according to their DP or DC values for the D-score calculation, according to the ATCT framework, because the number of terms extracted from a tweet is smaller than their experiment it would not be profitable to reduce them.

2.5 Contrast Evaluation Process

Table 1. Tags for the manual annotation (violence-related) in the dataset clusters. In this study, the Low and High labels were considered as one (1)

Tag	Description	Example(s) from D-LP
(0)	The content has no relation to any topic or description of violent events. As expected, this include all the clusters from D-NaK	<i>“No queremos y no nos da la gana ser mano de obra barata y explotada!” se oye hoy en las calles #LeyPulpin #15E</i>
(1) Low	The content has a ground description of a violent event or riot, most of all informative	<i>En redes, hubo abuso policial y pulpines heridos. En medios, hay pulpines vándalos que agreden policías y periodistas. #LeyPulpin #15E</i>
(1) High	The content describes an explicit action of a violent event, such as shots, clashes, fights, arrests, repression, etc	<i>Policía ha estado disparando con balas de goma. Cientos de jóvenes heridos dispersos por av. Tacna y Campo de Marte. #15E #LeyPulpin</i>

- **Manual cluster annotation:** In this evaluation, we proceeded to manually annotate all clusters obtained for the clustering step for both datasets, using the tweet headline contents. There are two possible tag values for violence absence (tag-0) or presence (tag-1) in the text content, following the criteria described in Table 1.
- **A threshold for cluster filtering:** In this step, the difference between datasets becomes relevant, because the maximum score of the non-violent dataset (D-NaK) clusters could serve as a threshold for cluster filtering in the violent dataset (D-LP). Figure 2 presents this proposition.

- **Metrics calculation:** Later, the F-score was calculated with the filtered clusters and their respective tags (0 or 1).
- **Threshold variation for metric values optimization:** Finally, We consider a threshold variation for the cluster filtering improvement. As can be seen in Fig. 2, we considered a subset with a p percent of clusters (ordered by the score) of the non-violent dataset. Thus, p increases from 0 to 100%, and the $p - 1$ values are counted as retrieved (in the metric computation) together with the cluster of the other dataset. This effect impacts the F-score in order to get a balanced p value for the threshold selection.

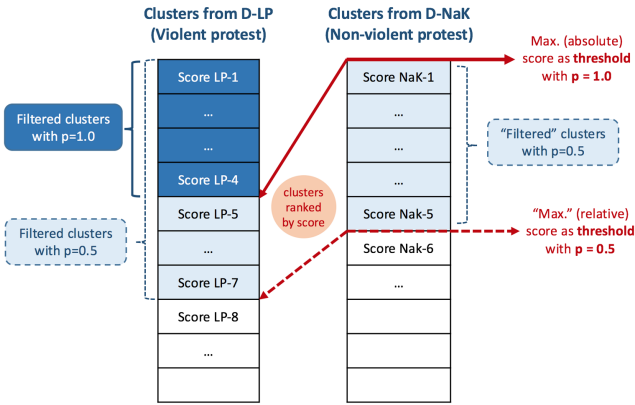


Fig. 2. Threshold selection and variation for clusters filtering in the contrast evaluation between violent and non-violent datasets

3 Experiments and Results

For the experiment, we considered different issues and parametric values along each step of the methodology:

- **Clustering:** The clusters are calculated for two time-windows values: 15 and 30 min. The reason is the importance of analyzing the filtering and selection process with different cluster granularity. In D-LP we obtained 150 and 70 clusters for the 15 and 30 min t-w, respectively. On the other hand, in D-NaK were extracted 54 and 41 clusters. Table 2 describes this brief stats of the number of clusters obtained and their tags per dataset and per time-window (t-w).
- **Term Extraction:** In this step, we replaced, in the tweet headline, user mentions with “USUARIO” (user) and hashtags with “OBJETO” (object). If that content were eliminated, the new order of the sentence would affect the POS-tag outcome. Furthermore, we filtered out a list of neutral terms composed by protest synonyms in Spanish: “marcha(s)” (march), “protesta(s)” (protest) and “manifestación(es)” (manifestation), because they are not discriminative enough due to their high presence in any protest-related dataset.

- **Term Domain-Based Scoring:** The measures calculation of every term per cluster is performed iterating different parameters values (α and β) for D-score. In this regard, we iterated their values from 0 to 1 and 1 to 0, respectively, with a 0.1 step value.
- **Evaluation - F-score:** The metric is calculated as shown in Fig. 3. In those results, the variation of α and β for both corpus tend to stabilize the scores in some fixed points. For the last step, and according to what is observed in the partial results, we set α in 0.5 (a steady middle point) for WIKI-Peace, and 0.9 for the SMT-Neutral corpus (a higher value would decrease the score abruptly).
- **Evaluation - Threshold:** With fixed α values, we proceeded to analyze the threshold variance for cluster filtering between the two datasets to identify the best p value for optimizing the computed metrics. Figure 4 shows F-score for this analysis.

Table 2. Stats of the obtained clusters (with different time-window) and their annotation values per each dataset

Dataset	t-w	#Clusters	Tag (0)	Tag (1)
D-LP	15 min	150	74	76
D-LP	30 min	70	29	41
D-NaK	15 min	54	54	0
D-NaK	30 min	41	41	0

As for the results, we observed different patterns for the two corpora in the evolution of F-score as α increases, although neither reaches to outdo the other. For the WIKI-Corpus, the fixed middle point (0.5) proves that both DP and DC values have the same relevance (so the domain and contrastive corpus get the same importance), while the SMT-Neutral shows less variance except at the end.

On the other hand, for the time-window parameter, a higher t-w (30 min.) obtained slightly better F-score values than the lower (15 min.). There were no differences in the use of the three groups of terms extracted, probably due to the reduced text content in tweets. Besides, regarding the threshold variation with α and β fixed, it is observed in Fig. 4 behavior pattern of the F-score when we reduce the maximum score (as a threshold) and increase the number of clusters filtered. As this process requires to define an optimal p , we set it at 0.9, the peak in almost all the F-score plot curves.

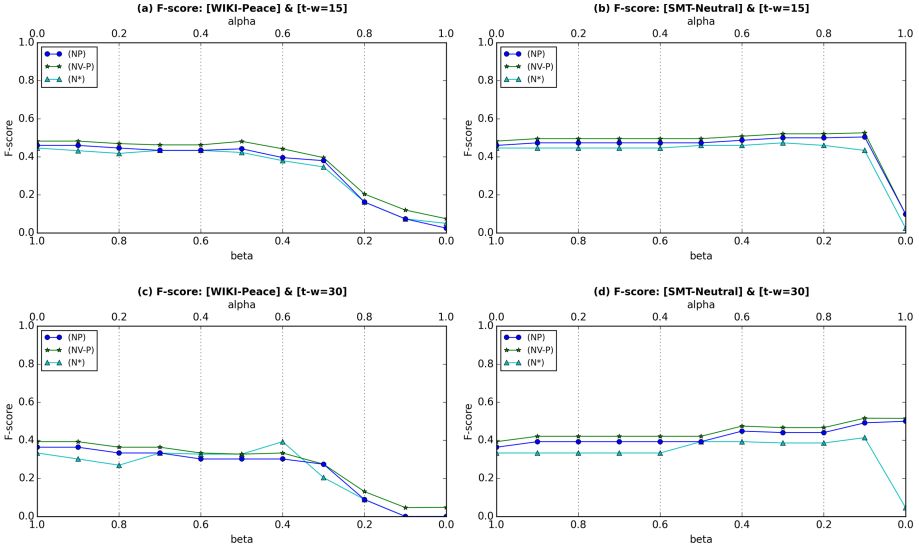


Fig. 3. Results for F-score: both contrast corpora are divided by column (WIKI-Peace at left and SMT-Neutral at right) and time-window by rows ($t-w$ in 15 and 30 min.). Each subplot shows the F-score per group of term extraction (N^* , NP , and $NV-P$) for different α and β (top and down axis in subfigures) used in the D-Score calculation.

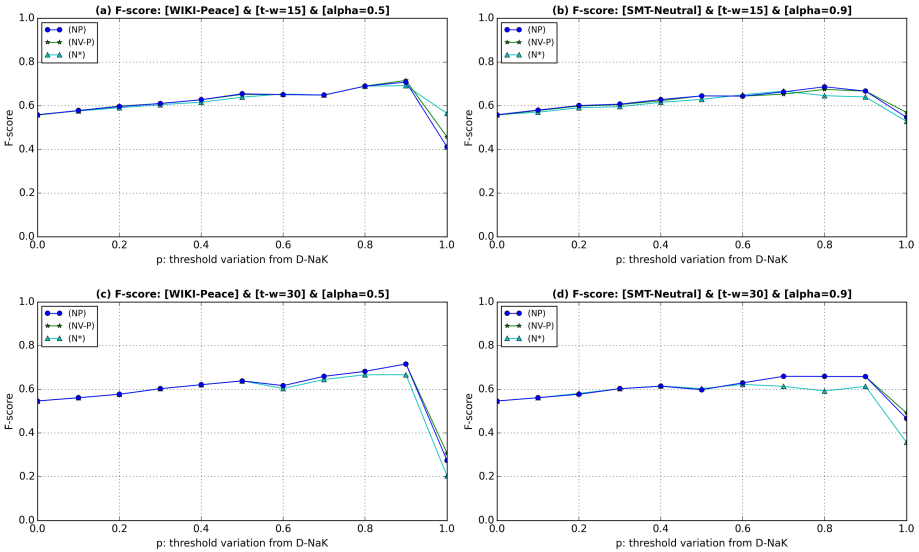


Fig. 4. Results for F-score with fixed α (0.5 for WIKI-Peace and 0.9 for SMT-Neutral) and a variant threshold (p) for the cluster filtering process between datasets. Subplots distribution is similar to Fig. 3.

Finally, Table 3 presents one of the experiments with the highest F-score. It is important to notice that, when p decreases for reducing the threshold score, the number of clusters (#c.) added from the contrast D-NaK dataset is minimal, while the F-score increases substantially with the additional cluster retrieved in D-LP.

Table 3. An experiment detail for the highest F-score obtained. Corpus: WIKI-Peace, t-w: 15 min, Terms: NP-NV, $\alpha = 0.5$ and $p = 0.9$

F-score $p = 0.9-p = 1.0$	Max. D-score with $p = 0.9-p = 1.0$	#c. filtered in D-LP $p = 0.9-p = 1.0$	#c. added ($p = 0.9$) from D-NaK
71.60% -45.71%	0.35 -0.75	86 -81	5

4 Conclusions and Future Work

This study focused on identifying incidents related to the violence domain during real-world civil unrest from tweets. A clustering method allowed us to segregate events using two different time-windows. Then, a term extraction, scoring, and ranking process were executed to calculating the domain closeness per cluster. The validation was performed, evaluating a contrast between two main social upheavals events in Perú (a violent and non-violent study case). In this way, the non-violent max score cluster served as a threshold for retrieving all suspected violent events in the other dataset, but this threshold was also not constrained to find an optimal value for the computed metrics. The obtained results confirmed our proposition, and the highest F-score value (71.60%) was achieved using a 15-min t-w for clustering, the WIKI-Peace as a contrast corpus, NP-NV as relevant terms, and an α value fixed to 0.5 for the D-score. Also, the p parameter played a critical role set at 0.9, increasing the F-score significantly while avoiding adding too many noisy clusters. Although the results are not outstanding, they are auspicious and have allowed us to demonstrate that the proposed methodology for the sub-event identification, and specifically the contrast evaluation.

As future work, the keyword extraction for the domain could be improved, considering a more complex VP extraction (e.g., analyzing subordinated terms). Besides, a robust domain and contrast corpus could be acquired with explicit relevant and neutral content, respectively, considering an automatic keyword-based extraction from the web. Furthermore, an ontology for the violence domain could be built, in order to increase the accuracy of the clusters filtered and to help in the description of more aspects (e.g., action, actors, items) from the violent event detected.

Finally, this procedure could be abstracted to a non-attached domain framework, and the contrast evaluation proposed must be subject to new experiments with different datasets and domain corpus to confirm its effectiveness further.

References

1. Aiello, L.M., et al.: Sensing trending topics in Twitter. *IEEE Trans. Multimedia* **15**(6), 1268–1282 (2013)
2. Alsaedi, N., Burnap, P., Rana, O.: Can we predict a riot? disruptive event detection using twitter. *ACM Trans. Internet Technol.* **17**(2) (2017). <https://doi.org/10.1145/2996183>
3. Anastasopoulos, L.J., Williams, J.R.: A scalable machine learning approach for measuring violent and peaceful forms of political protest participation with social media data. *PLoS ONE* **14**(3), e0212834 (2019)
4. Benkhelifa, E., Rowe, E., Kinmond, R., Adedugbe, O., Welsh, T., et al.: Exploiting social networks for the prediction of social and civil unrest: a cloud based framework. In: 2014 International Conference on Future Internet of Things and Cloud (FiCloud), pp. 565–572. *IEEE* (2014)
5. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. *J. Mach. Learn. Res.* **3**, 993–1022 (2003)
6. Cimiano, P., Hotho, A., Staab, S.: Learning concept hierarchies from text corpora using formal concept analysis. *J. Artif. Intell. Res. (JAIR)* **24**, 305–339 (2005)
7. Cimiano, P., Völker, J.: Text2Onto. In: Montoyo, A., Muñoz, R., Métais, E. (eds.) *NLDB 2005. LNCS*, vol. 3513, pp. 227–238. Springer, Heidelberg (2005). https://doi.org/10.1007/11428817_21
8. Eisenstein, J.M.D.V.J., De Choudhury, M.: Psychological effects of urban crime gleaned from social media (2015)
9. Guberman, J., Schmitz, C., Hemphill, L.: Quantifying toxicity and verbal violence on Twitter. In: Proceedings of the 19th ACM Conference on Computer Supported Cooperative Work and Social Computing Companion, pp. 277–280. *ACM* (2016)
10. Hua, T., et al.: Analyzing civil unrest through social media. *Computer* **12**, 80–84 (2013)
11. Ifrim, G., Shi, B., Brigadir, I.: Event detection in twitter using aggressive filtering and hierarchical tweet clustering. In: *SNOW-DC@ WWW*, pp. 33–40 (2014)
12. Jin, F., et al.: Modeling mass protest adoption in social network communities using geometric Brownian motion. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1660–1669. *ACM* (2014)
13. Kang, Y.B., Haghighi, P.D., Burstein, F.: CFinder: an intelligent key concept finder from text for ontology development. *Expert Syst. Appl.* **41**(9), 4494–4504 (2014)
14. Koehn, P.: Europarl: a parallel corpus for statistical machine translation. *MT Summit.* **5**, 79–86 (2005)
15. Meijer, K., Frasinca, F., Hogenboom, F.: A semantic approach for extracting domain taxonomies from text. *Dec. Supp. Syst.* **62**, 78–93 (2014)
16. Myers, S.A., Zhu, C., Leskovec, J.: Information diffusion and external influence in networks. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 33–41. *ACM* (2012)
17. Papadopoulos, S., Corney, D., Aiello, L.M.: Snow 2014 data challenge: assessing the performance of news topic detection methods in social media. In: *SNOW-DC@ WWW*, pp. 1–8 (2014)
18. Petrović, S., Osborne, M., Lavrenko, V.: Streaming first story detection with application to Twitter. In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 181–189. Association for Computational Linguistics (2010)

19. Ramakrishnan, N., et al.: ‘Beating the news’ with embers: forecasting civil unrest using open source indicators. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1799–1808. ACM (2014)
20. Schmid, H.: Treetagger—a language independent part-of-speech tagger. *Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart* **43**, 28 (1995)
21. Sclano, F., Velardi, P.: Termextractor: a web application to learn the shared terminology of emergent web communities. In: Gonçalves, R.J., Müller, J.P., Mertins, K., Zelm, M. (eds.) *Enterprise Interoperability II*, pp. 287–290. Springer, London (2007). https://doi.org/10.1007/978-1-84628-858-6_32
22. Varol, O., Ferrara, E., Ogan, C.L., Menczer, F., Flammini, A.: Evolution of online user behavior during a social upheaval. In: Proceedings of the 2014 ACM Conference on Web Science, pp. 81–90. ACM (2014)



Classification of Relationship in Argumentation Using Graph Convolutional Network

Dimmy Magalhães^(✉) and Aurora Pozo

Departamento de Ciência da Computação, Universidade Federal do Paraná,
Curitiba, PR, Brazil
{dksmagalhaes,aurora}@inf.ufpr.br

Abstract. The Argument Relationship Prediction is one of the tasks of Argumentation Mining that aim to find connections between arguments (or parts thereof). This task is considered as one of the most complex stages of argumentation. Concomitant to that, the Graph Convolutional Network (GCN) has been successfully applied to graph-based applications. In this study, we join the relationship prediction challenge with the ability of GCN to classification. We propose ArgGCN, a framework based in GCN method applied to the classification of relationships between arguments. The arguments are considered as short texts, and we abstracted the recognition of unitary elements from them (such as claims and evidence). In this study, we achieved promising results on the UKP Aspect, AFS, and Microtext corpus.

Keywords: Argumentation Mining · Graph convolutional networks · Machine Learning in NLP

1 Introduction

Argumentation is a branch of philosophy that studies the act or the process of forming reasons and drawing conclusions in the context of a discussion, dialogue, or conversation [22]. According to [24], an argument is made of six components: a *datum* which forms the basis for making a *claim*, the *warrant* (the rule of inference), the *qualifiers* (elements that show how certain we are of the claim), the *rebuttals* (set of conditions for the claim to be hold) and finally the *backing* (a justification to the warrant).

The application of artificial intelligence in the field of argumentation has been expanding rapidly in recent years due to its ability to build representations, cognitive models, and computational models for automated reasoning. In particular, *Argumentation Mining (AM)* is an area of research that focuses on the identification and the extracting of the claims and evidence, inference their structures and relationship from generic textual corpora, in order to provide structured data for computational models of argument and reasoning engines [12].

In AM context, an argument is a set of statements consisting on three components: a set of premises, a conclusion, and an inference from the premises to the conclusion [26]. In the literature, conclusions are sometimes referred to us as *claims*, *premises* are often called evidence or *reasons*, and the link between the two, that is, the inference is sometimes called the argument itself [12]. In general, AM consists of three macro tasks:

1. Argumentative sentence detection: The first task in the argumentation usually addresses the task of extracting those sentences in the input document that contains an argument (or part of it), and that can, therefore, be defined as argumentative [12];
2. Argument component detection: The second task is to detect the exact boundaries of each argument component, such as claim or evidence [23];
3. Argument relationship prediction: The main objective of this task is to find connections between the elements extracted in the previous activities. In general, the output of this stage is a graph connecting the retrieved arguments (or parts thereof).

Figure 1 shows a generic pipeline architecture of the AM system. Figure 2 shows an example of text and its decomposition on claims, evidences and relationships.

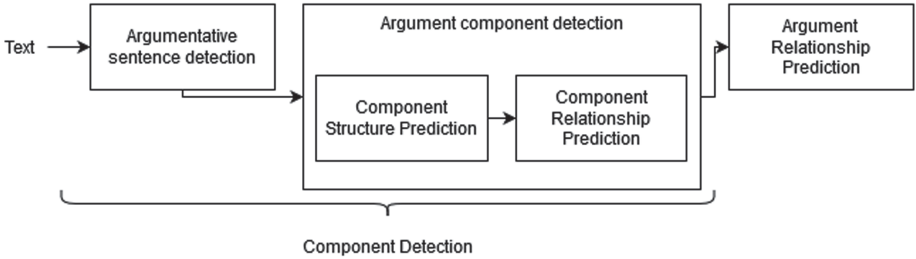


Fig. 1. Generic pipeline architecture of AM systems

In this context, the argument relationship prediction is the most complex stage from pipeline. It requires to understand connections and relationships between the detected arguments, thus involving high-level knowledge representation and reasoning issues [12]. Knowledge representation (specifically, text representation) has been learned using convolutional models, such as convolutional neural networks (CNN) [11] and long short-term memory (LSTM) [5]. In [27], the authors proposed a graph text representation that captures context information of the terms using the graph convolutional network (GCN) model [7].

Motivated by this perspective, in this study, we propose a new graph-based representation for relationship argument classification, **ArgGCN**. We build an unique graph from all the arguments, like **TextGCN** [27]. However, the graph contains words and relationship arguments nodes. The edge between two-word

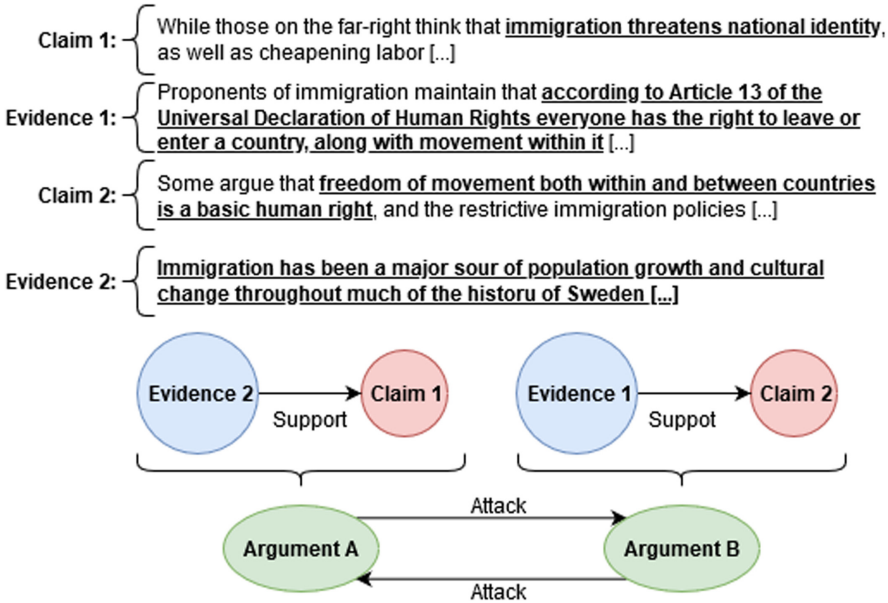


Fig. 2. Example of a text and its decomposition on claims, evidence and relationships. Adapted from [12]

nodes is built by word co-occurrence information. The edge between a word node and relationship argument node is built using word frequency. Finally, we model the graph with a Graph Convolutional Network [7]. We hypothesize that this representation can capture contextual features inherent in the relationships between arguments. The details of the proposed approach are explained in Subsect. 4.1.

We evaluated our approach by comparing it with TF-IDF-based text classification, with a BERT-based grouping approach, and with human performance. We use three widely used benchmark corpora: UKP ASPECT Corpus [21], AFS Corpus [14] and Microtext [18].

The remainder of the paper is organized as follows. Related work is discussed in Sect. 2. Section 3 presents an overview of Graph Convolutional Network used to build the model graph. In Sect. 4, the framework ArgGCN and methodology of the experiments are presented. Section 5 describes the results and discussions. Finally, advantages, limitations, and future research directions are discussed in Sect. 6.

2 Related Work

In recent years, identifying relationships between arguments, or between elements of an argument (claim and evidence) has gained focus in the AM research area. Approaches to identify these relations are based on the prior classification of individual clauses or in the direct extraction of relations [10].

Firstly, we highlight the studies that relate elements of an argument. Palau et al. [16] created a classification model for each argument sentence as either premise or conclusion using a context-free grammar. Peldszus et al. [17, 19] used a microtext corpus where they first identify the roles of argument segments and then conflict relations by examining the texts for occurrences of counter-considerations. Cartens et al. [3] represent customer reviews as a tree of arguments finding a relationship between components of arguments, where a child-parent relationship between two sentences is determined if they refer to the same concepts, with the child being the sentence that has been posted later.

Cabrio and Villata [2] propose a textual entailment to analyze online dialogues to extract the abstract arguments and their relationships. Lawrence and Reed [8] used a LDA topic model to determine the topic similarity of consecutive propositions in a piece of text. In [9], they improved the approach replacing LDA by WordNet¹. Wachsmuth et al. [25] presented a model to determine the best counterargument to any argument without prior knowledge of the topic of the argument, using a five pre-trained word embedding models for representing arguments and evaluate four inverse vector-based distance measures: Cosine, Euclidean, Manhattan, and, Jaccard similarity.

In the analyzed studies, we highlight the need for the general argument structure or how claim and evidence are related. In all models presented, except in the approach of Wachsmuth, a set of features must be proposed to build the relationships of the arguments. That is, the models reduce the arguments to their elementary units and then build such relationships. On the other hand, we use a convolutive approach to extract features from the arguments, and there is no prior knowledge about the units that compound the argument.

3 Graph Convolutional Network

In general, the goal in graph convolutional model is learn a function of features on a graph $G = (V, E)$, where V is set of node and E is set of edges, which takes as input:

- A feature description x_i for every node i ; summarized in a $N \times D$ feature matrix X (N : number of nodes, D : number of input features)
- A representative description of the graph structure in matrix form.

and produces a node-level output Z (a $N \times F$ feature matrix, where F is the number of output features per node). Furthermore, every neural network layer can then be written as a non-linear Eq. 1:

$$H^{(l+1)} = f(H^{(l)}, A) \tag{1}$$

with $H^{(0)} = X$ and $H^{(L)} = Z$, L is the number of layers. The specific models then differ only in how $f(.,.)$ is chosen and parameterized [7].

¹ <https://wordnet.princeton.edu/>.

The GCN proposed by Kipf and Welling [7] is a multilayer neural network that operates directly on a graph and induces embedding vectors of nodes based on properties of their neighborhoods. They introduced a simple layer-wise propagation rule (Eq. 2) for neural network models based on a first-order approximation of spectral convolutions on graphs [4].

$$H^{(l+1)} = \sigma(\bar{D}^{-\frac{1}{2}} \bar{A} \bar{D}^{-\frac{1}{2}} H^{(l)} W^l) \quad (2)$$

where $\bar{A} = A + I_N$ is the adjacency matrix of the undirected graph G with added self-connections. I_N is the identity matrix, $\bar{D}_{ii} = \sum_j \bar{A}_{ij}$, and W^l is a layer-specific trainable weight matrix. $\sigma(\cdot)$ denotes an activation function, such as the ReLU [15]. H^l in $\mathbb{R}^{N \times D}$ is the matrix of activation in the l^{th} layer. In particular, for a one-layer GCN, the new D -dimensional node feature matrix $H^{(1)} \in \mathbb{R}^{N \times D}$ is computed as:

$$H^1 = \sigma(\bar{D}^{-\frac{1}{2}} \bar{A} \bar{D}^{-\frac{1}{2}} X W^0) \quad (3)$$

It is important to note that convolution in a GCN is closely linked to the adjacency matrix (constructed from the graph). Therefore, the edge values present in the adjacency matrix can direct the convolution to distinct niches in the vector space, thus justifying an in-depth analysis of this aspect of graph construction.

The authors present some limitations of their model, in particular, we highlight that the GCN does not naturally support edge features and is limited to undirected graphs (weighted or not). This is an important limitation, since in many cases the arguments are connected by directed relationships.

4 Methodology

In this section, we describe how to construct a meaningful graph-representation to describe the relationship between arguments, ArgGCN. Furthermore, we describe how we will evaluate ArgGCN. In particular, we want to determine if our model can achieve satisfactory results considering the classification of relationships in argumentation.

4.1 ArgGCN

A straightforward manner to create a graph from arguments is to treat words and relationships as nodes in the graph. Therefore, we need to build edges for word-relationship and word-word nodes. It is essential to highlight that, in general, the arguments in the analyzed datasets are composed of few relevant words, which makes us treat them as short texts.

In this context, ArgGCN is composed of three components to create a graph-based representation for relationship between arguments. Firstly, we have a pre-processing and quantitative analysis of texts. The datasets are preprocessed by cleaning them, tokenizing, and removing stop words defined in NLTK². We did

² <http://www.nltk.org/>.

not use the stemming process proposed by [13]. In the quantitative analysis stage, we compute the frequency of the words and remove words with very high and very low frequencies, the main reason for this is that we consider that such words are not discriminating and can reduce the ability of the model to associate arguments or can cause wrong contextual relationships. This task produces a dictionary of words that are used in the subsequent steps of the framework.

Secondly, ArgGCN computes the contextual metrics between words and relationships:

- **Contextual windows:** ArgGCN creates context windows between all words in all arguments (in pairs) and checks how often those words appear together;
- **Compute PMI:** ArgGCN employs point-wise mutual information (PMI) as Eq. 4 [27];
- **Compute TF-IDF:** ArgGCN computes a word-relationship values using TF-IDF algorithm for pairs of arguments according to [20].

$$PMI(i, j) = \log \frac{p(i, j)}{p(i)p(j)} \quad (4)$$

$$p(i, j) = \frac{W(i, j)}{W} \quad (5)$$

$$p(i) = \frac{W(i)}{W} \quad (6)$$

where $p(i, j)$ represents the ratio of words i and j on the total of windows and $p(i)$ represents the ratio of word i on the total of windows, such that $W(i, j)$ is the number of sliding windows that contain both word i and j and $W(i)$ is the number of sliding windows in a corpus that contain word i , and W is the total number of sliding windows in the corpus.

ArgGCN has a stage of processing the labels. In general, the label of an argument pair specifies the relationship between the two sentences, for example, if the two arguments agree on a topic. For ArgGCN, when we use this simpler labeling, we call them **ArgGCN_{simple}**. However, in order to expand our framework, we have created a more specific labeling model. We modify the label of the argument pairs, turning them into a tuple $L = (t, l)$, where t is the topic, and l is the label of the argument. Thus, we expanded the number of labels, making each pair of arguments more specific to the topic. This model configuration, we call **ArgGCN_{complex}**.

Finally, ArgGCN builds a complex graph with different kind of nodes represented by an adjacent matrix. The first lines of the matrix represent the relationships or arguments followed by N lines representing the vocabulary; the same is true for the columns of the matrix. The adjacency matrix values come from the PMI and TF-IDF computation steps. After building the graph, we run GCN with two layers. The softmax activation function, defined as $softmax(x_i) = \frac{1}{z} exp(x_i)$ with $z = \sum_i exp(x_i)$, is applied row-wise, like in [7] and [27]. Figure 3 illustrates the overall ArgGCN model.

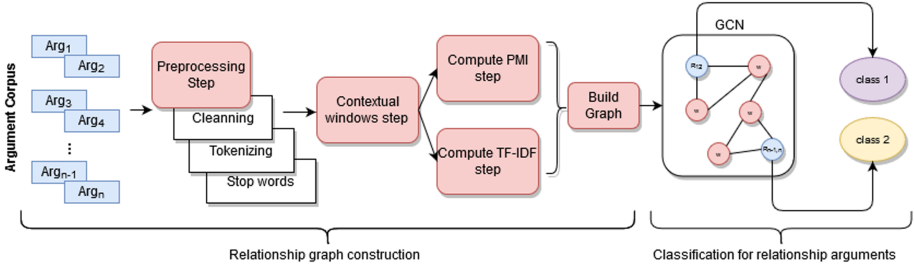


Fig. 3. The ArgGCN framework for relationship arguments classification

4.2 Evaluation

To evaluate our hypothesis, we compare ArgGCN with multiple text classification and embedding methods as follows:

- **TF-IDF + LR**: bag-of-words model with weighting term frequency- inverse document frequency. Logistic Regression is used as the classifier
- **BERT Similarity**: clustering using BERT models based on [21]. We use the approach with base and large BERT models.
- **Human Performance**. We use the results reported in [21] as a comparative basis between our approach to a human assessment approach.

The evaluation of the approach was conducted using three widely used benchmark corpora, including UKP ASPECT Corpus, AFS Corpus, and Microtext.

- **UKP ASPECT Corpus** was proposed by [21] with annotations on similar and dissimilar sentence-level arguments. The dataset contains 28 topics related to currently discussed issues from technology and society and 3,595 arguments pairs, 130 pairs for each topic.
- **AFS Corpus** created by Misra et al. [14] contains 6,000 sentential argument pairs for three topics and annotated them on a scale from 0 (“different topic”) to 5 (“completely equivalent”).
- **Microtext (MT)** was proposed by [17] is an import corpus of 112 arguments manually created, short texts with explicit argumentation.

Table 1 summarizes the values found for each dataset. The number of words is potentially different from the number of nodes due to two main reasons: 1) in the study, each argument pair is represented by a node. 2) Not all words present in the dataset will become a node in the representation, mainly due to the low-frequency reason. In the MT dataset, we have argument components, i. e. in the same argument we can have more than one pair of relationships.

In general, the arguments are defined as a tuple $T = (t, s_1, s_2, l)$, where t is the topic of arguments, s_1 and s_2 represent the sentences of each argumentative text and l is the label that relates the two sentences (for example, *high similar*). However, in some cases, datasets are represented using a directed graph

Table 1. Descriptive statistics of each dataset

Dataset	Arguments	Pairs	Words	Nodes	Edges	Classes
UKP	7,166	3,583	5,865	9,448	465,645	4
AFS	3,914	1,957	2,014	3,971	151,817	5
MT	836	418	1,979	2,397	78,141	3

$G = (V, A)$ where V is set of all arguments and A is a set of ordered relationships, in which v_{ij} and v_{ji} represent different relationships. In this study, for each v_{ij} and v_{ji} relationship was created a tuple $T_i = (t, s_1, s_2, l_i)$ and $T_j = (t, s_2, s_1, l_j)$, where l_i and l_j represent opposing relationships.

4.3 Settings

We model the graph with GCN. The GCN parameters follow [27]. We set the embedding size of the first convolution layer as 300 and set the window size as 10, learning rate as 0.02, dropout rate as 0.5, L_2 loss weight as 0. As in [6], the training and test vocabulary are processed together, so that the entire model knows all the words.

In order to further analyze our model, we performed some additional experiments using ArgGCN, varying some of its parameters. The main objective of this task is to identify critical points that impact the performance of the model. We analyzed the behavior of ArgGCN from the perspective of the context window size and the and word frequencies.

The following classification evaluation measures were collected: accuracy and F1-Score. We perform 30 independent rounds on training/test sets generated by the 4-fold cross-validation process reporting the average of the selected metrics. For implementation, we use TensorFlow [1]. The code for the framework and complete results is publicly available³.

5 Results and Discussion

In the following, we present and analyze the results. Table 2 shows the performance on all datasets. The results indicate that models applied to the relationship of argumentation based on bag-of-words to capture context using the TF-IDF show the worst performance. Such models do not have a clear contextual association between the arguments, and this can be explained by the high Euclidean distance value between the representations of consonant arguments with the same topic. Table 3 shows an example of the average Euclidean distance values between similar and high similar arguments using TF-IDF model for the UKP dataset (the value in parentheses represents the standard deviation). The values indicate that the vector representation of two similar arguments,

³ <https://github.com/dimmykarson/argGCN>.

Table 2. Average of F₁ scores on the datasets

Model	Dataset		
	UKP	AFS	MicroText
Human Performance	.7834	-	-
Supervised methods			
TF-IDF + LR	.6118	.6035	.5821
BERT _{base}	.7401	.7475	.7089
BERT _{large}	.7256	.7200	.6911
ArgGCN_{simple}	.7478	.7355	.6312
ArgGCN_{complex}	.7374	.7112	.6141

Table 3. Euclidean distances between similar (SS) and high similar(HS) arguments for UKP dataset.

Topic	Labels	
	SS	HS
Cryptocurrency	1.357749 (0.047534)	1.101095 (0.550826)
Gene editing	1.394102 (0.022195)	1.397306 (0.019416)
Net neutrality	1.265709 (0.354727)	1.370873 (0.038313)
Solar energy	1.350535 (0.058759)	1.382380 (0.027838)

which should be contextually close, are considerably separated in the vector space.

The models based on BERT are superior or statistically equivalent to the results presented by ArgGCN. Although the experiments with ArgGCN are preliminary, it is essential to emphasize that the representation in graphs of argument relationships reaches very close values. Text representations based on BERT are obtained through a previously trained model with a massive amount of data, which indicates that they are contextually more cohesive. The ArgGCN has no previous training, or embedding model, it can justify the superiority of BERT-based models.

Specifically, we found the best results in the approach ArgGCN_{simple}. In particular, for the UKP dataset, this experiment reached F1 score of 0.7478 (with standard deviation equals 0.0866). Despite being a simpler model, the construction of the graph was able to relate words of different arguments and different topics, thus creating significant areas of context.

About the context window size and word frequencies, the results indicate that these parameters are decisive for the success of the model. Figure 4 shows that the larger the context window, the worse the model. ArgGCN begins to build contextual relationships that have little impact on each other.

The frequency of words refers to how many times a word must appear in the entire dataset to be considered in the model. Very low values indicate that the

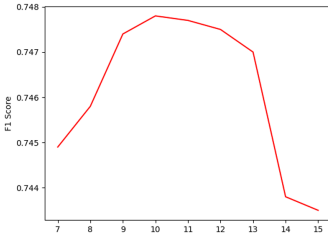


Fig. 4. Size of context window in ArgGCN.

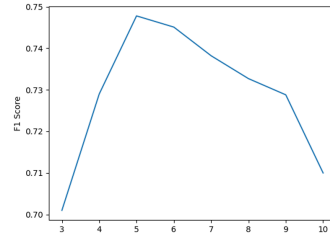


Fig. 5. Frequency of words in ArgGCN.

model should consider any words, even if they are not very relevant. Very high values generalize the model too much, leading it to consider only common and little discriminating words. These characteristics can be seen in ArgGCN, which is one of the main parameters of the model (see Figure 5). The model showed a slight sensitivity to the word frequency values, to the point of decreasing the accuracy and F1-score values when only words that appeared more frequently in the dataset are considered. This prior analysis of these parameters indicates that the model is very sensitive to them, which requires special attention.

6 Conclusion and Future Works

In this study, a framework for the classification of relationships in argumentation using GCN was proposed. The goal was to increase the set of contextual possibilities between arguments, using a representation based on heterogeneous graphs. A graph was built by using words and relationships as nodes. Our hypothesis was that to induce convolutions about the graph, capable of creating contextual areas between pairs of arguments would be sufficient to build a model for classification of relationships.

The results validate the hypothesis that the graph-based representation can capture context for the relationship between arguments. The presence of a statistical difference between the models for the same dataset indicates that there is still a gap between the consolidated models and the proposed approach. However, the refinement of the representation with more appropriate metrics of association between the arguments of the same pair of arguments can direct future work to improve the effectiveness of the model.

Acknowledgments. This work was funded by CAPES and Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq - Brazil, Tribunal de Justiça do Estado do Piauí - TJPI and supported by Intel® AI DevCloud.

References

1. Abadi, M., et al.: Tensorflow: large-scale machine learning on heterogeneous distributed systems. arXiv preprint [arXiv:1603.04467](https://arxiv.org/abs/1603.04467) (2016)

2. Cabrio, E., Villata, S.: Generating abstract arguments: a natural language approach. In: COMMA, pp. 454–461 (2012)
3. Carstens, L., Toni, F., Evripidou, V.: Argument mining and social debates. *Argument* **2**, 3 (2014)
4. Hammond, D.K., Vandergheynst, P., Gribonval, R.: Wavelets on graphs via spectral graph theory. *Appl. Comput. Harmonic Anal.* **30**(2), 129–150 (2011)
5. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
6. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, 25–29 October 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL., pp. 1746–1751 (2014). <http://aclweb.org/anthology/D/D14/D14-1181.pdf>
7. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. *CoRR abs/1609.02907* (2016). <http://arxiv.org/abs/1609.02907>
8. Lawrence, J., Reed, C.: Aifdb corpora. In: Parsons, S., Oren, N., Reed, C., Cerutti, F. (eds.) *Computational Models of Argument - Proceedings of COMMA 2014*, Atholl Palace Hotel, Scottish Highlands, UK, September 9–12, 2014. *Frontiers in Artificial Intelligence and Applications*, vol. 266, pp. 465–466. IOS Press (2014). <https://doi.org/10.3233/978-1-61499-436-7-465>
9. Lawrence, J., Reed, C.: Combining argument mining techniques. In: Proceedings of the 2nd Workshop on Argumentation Mining, ArgMining@HLT-NAACL 2015, June 4, 2015, Denver, Colorado, USA, pp. 127–136. The Association for Computational Linguistics (2015). <https://doi.org/10.3115/v1/w15-0516>
10. Lawrence, J., Reed, C.: Argument mining: a survey. *Comput. Linguist.* **45**(4), 765–818 (2020). https://doi.org/10.1162/coli_a.00364
11. Lawrence, S., Giles, C.L., Tsoi, A.C., Back, A.D.: Face recognition: a convolutional neural-network approach. *IEEE Trans. Neural Networks* **8**(1), 98–113 (1997)
12. Lippi, M., Torroni, P.: Argumentation mining: state of the art and emerging trends. *ACM Trans. Internet Technol.* **16**(2) (2016). <https://doi.org/10.1145/2850417>
13. Lovins, J.B.: Development of a stemming algorithm. *Mech. Translat. Comput. Linguistics* **11**(1–2), 22–31 (1968). <http://www.mt-archive.info/MT-1968-Lovins.pdf>
14. Misra, A., Ecker, B., Walker, M.A.: Measuring the similarity of sentential arguments in dialog. arXiv preprint [arXiv:1709.01887](https://arxiv.org/abs/1709.01887) (2017)
15. Nair, V., Hinton, G.E.: Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10), 21–24 June 2010, Haifa, Israel, pp. 807–814 (2010), <https://icml.cc/Conferences/2010/papers/432.pdf>
16. Palau, R.M., Moens, M.F.: Argumentation mining: the detection, classification and structure of arguments in text. In: Proceedings of the 12th International Conference on Artificial Intelligence and Law, pp. 98–107 (2009)
17. Peldszus, A.: Towards segment-based recognition of argumentation structure in short texts. In: Proceedings of the First Workshop on Argumentation Mining, pp. 88–97 (2014)
18. Peldszus, A., Stede, M.: An annotated corpus of argumentative microtexts. In: Proceedings of the First Conference on Argumentation, Lisbon, Portugal, June (2015, to appear)
19. Peldszus, A., Stede, M.: Towards detecting counter-considerations in text. In: Proceedings of the 2nd Workshop on Argumentation Mining, pp. 104–109 (2015)

20. Ramos, J., et al.: Using TF-IDF to determine word relevance in document queries. In: Proceedings of the first Instructional Conference on Machine Learning, Piscataway, NJ, vol. 242, pp. 133–142 (2003)
21. Reimers, N., Schiller, B., Beck, T., Daxenberger, J., Stab, C., Gurevych, I.: Classification and clustering of arguments with contextualized word embeddings. arXiv preprint [arXiv:1906.09821](https://arxiv.org/abs/1906.09821) (2019)
22. Sardianos, C., Katakis, I.M., Petasis, G., Karkaletsis, V.: Argument extraction from news. In: Proceedings of the 2nd Workshop on Argumentation Mining, pp. 56–66 (2015)
23. Stab, C., Gurevych, I.: Annotating argument components and relations in persuasive essays. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. pp. 1501–1510 (2014)
24. Toulmin, S.E.: The Uses of Argument. Cambridge University Press, Cambridge (2003)
25. Wachsmuth, H., Syed, S., Stein, B.: Retrieval of the best counterargument without prior topic knowledge. In: Gurevych, I., Miyao, Y. (eds.) Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15–20, 2018, Volume 1: Long Papers, pp. 241–251. Association for Computational Linguistics (2018). <https://doi.org/10.18653/v1/P18-1023>, <https://www.aclweb.org/anthology/P18-1023/>
26. Walton, D.: Argumentation theory: a very short introduction. In: Simari, G., Rahwan, I. (eds.) Argumentation in artificial intelligence, pp. 1–22. Springer, Heidelberg (2009). https://doi.org/10.1007/978-0-387-98197-0_1
27. Yao, L., Mao, C., Luo, Y.: Graph convolutional networks for text classification, vol. abs/1809.05679 (2018). <http://arxiv.org/abs/1809.05679>



Recursive Expressions for SPARQL Property Paths

Ciro Medeiros^{1,2(✉)}, Umberto Costa¹, Semyon Grigorev^{3,4},
and Martin A. Musicante¹

¹ Universidade Federal do Rio Grande do Norte, Natal, Brazil
`cirommed@ppgsc.ufrn.br`, `{umberto,mam}@dimap.ufrn.br`

² Laboratoire d'Informatique Fondamentale d'Orléans, Orléans, France
`ciro.morais-medeiros@etu.univ-orleans.fr`

³ Saint Petersburg State University, St. Petersburg, Russia
`s.v.grigoriev@spbu.ru`

⁴ JetBrains Research, St. Petersburg, Russia
`semyon.grigorev@jetbrains.com`

Abstract. Regular expressions are used in SPARQL property paths to query RDF graphs. However, regular expressions can only define the most limited class of languages, called regular languages. Context-free languages are a wider class containing all regular languages. There are no context-free expressions to define them, so it is necessary to write grammars. We propose an extension of regular expressions, called recursive expressions, to support the definition of a subset of context-free languages. The goal of our work is therefore to provide simple operators allowing the definition of languages as close as possible to context-free languages.

Keywords: Graphs · Context-Free Path Queries · Recursive expressions

1 Introduction

Several database models have been defined over the years, being *relational databases* the most well studied ones, due to their robustness concerning scalability and data consistency. Relational databases represent data in the form of n -ary relations (or tables), where each element is given as a tuple (or record). Regardless the adopted database model, data are retrieved from such sources by using queries specified over a domain-specific language. SQL [3] is extensively used as a query language in the context of relational database systems.

In recent years, *graph databases* have become popular. This database model represents data as subject-predicate-object triples (s, p, o) , where s and o are nodes of the graph and p is the label of the edge linking them. The *Resource Description Framework* (RDF) is W3C's recommendation for implementing

Linked Data [14]. An RDF database is a set of those triples. SPARQL is a query language similar to SQL and is the standard query language for RDF databases.

A SPARQL query is defined by specifying paths inside the graph and using relational operators to construct the answer to the query. SPARQL defines *Property Paths*, which are patterns formed by edge labels of the graph (*i.e.*, predicates, in RDF terminology). Those patterns are used to link nodes that participate in the query. SPARQL property paths are defined by means of regular expressions, thus defining regular paths inside the graph.

Regular expressions have proved to be useful for querying paths in RDF graph databases [13]. However, some papers address the problem of increasing the expressiveness in path queries to support context-free languages. The idea of this line of research is to be able to directly support queries such as the *Same Generation Queries* [1]. Queries of this kind are called *Context-Free Path Queries* (CFPQs) [5]. In general, these studies focus on the development of algorithms for the evaluation of such queries [2, 4–6, 8, 12].

Most of the research efforts in the area are devoted to the definition of algorithms to implement CFPQs. However, a few of them propose syntactic constructors for query languages to support context-free path queries. This is an important aspect of the query language *pragmatics* that still needs to be fully addressed. We can mention three initiatives devoted to define non-regular query languages for graph databases. The first one is an extension of the Cypher query language¹ that proposes the definition of a query to contain a context-free grammar, whose non-terminal symbols are used inside the property path. The proposal in [8] extends SPARQL in a similar way. In [10], SPARQL is extended with *Nested Regular Expressions*, an extension of Regular Expressions.

In this context, writing a context-free path query requires the knowledge of grammars and the use of a notation to represent them in a query format. Whilst regular expressions as used in SPARQL are concise and widely known by the database community, context-free grammars are not as simple nor well known.

The main goal of our work is to investigate the use of an extension of regular expressions to define a meaningful subset of context-free languages. We define *recursive expressions* as an extension of regular expressions, in order to specify a subset of context-free languages. We define the syntax and operational semantics of rcfSPARQL (restricted-context-free SPARQL), a query language that includes property paths built upon recursive expressions. Although rcfSPARQL is not as expressive as languages for specifying general CFPQs, we argue that it presents a reasonable trade-off by combining the ease of specification provided by regular expressions and part of the expressiveness of context-free grammars.

This paper is organized as follows. Section 2 presents notions about context-free languages and path queries in the context of graph databases. In sect. 3 we propose the syntax of recursive expressions and formalize their meaning in terms of context-free grammars. In Sect. 4 we show how recursive expressions can be used to specify rcfSPARQL queries. The operational semantics of rcfSPARQL is

¹ <https://github.com/thobe/openCypher/blob/rpq/cip/1.accepted/CIP2017-02-06-Path-Patterns.adoc#153-compared-to-context-free-languages>.

given in Sect. 5. Finally, we conclude the paper in Sect. 6 by adding important remarks and directions for future work.

2 Preliminaries

An *alphabet* Σ is a finite set of symbols. A finite sequence of symbols from an alphabet Σ is a *string*. A set of strings forms a *language*.

Context-Free Languages can be generated by *Context-Free Grammars*, which are quadruples $G = (N, \Sigma, P, S)$, where Σ is an alphabet of terminal symbols (the alphabet of a language); N is a finite set of non-terminal symbols; P is a set of production rules of the form $A \rightarrow \alpha$, where $A \in N$ and $\alpha \in (N \cup \Sigma)^*$, and $S \in N$ is the start symbol.

A *Dyck Language* is a context-free language whose strings are formed by balanced parentheses. Dyck languages are a proper subclass of context-free languages. A Dyck language over n balanced pairs may be specified by the following grammar: $S \rightarrow a_0 S b_0 S \mid \dots \mid a_{n-1} S b_{n-1} S \mid \epsilon$, where (a_i, b_i) are terminals that specify i -th balanced pair. This grammar is called a Dyck Grammar.

A *Graph Database* (also referred to as a *Data Graph*) is a set of triples (s, p, o) , where s is the subject, p is the predicate and o is called the object of the tuple. The elements s and o are seen as vertices of a graph and p is a label to a directed edge from s to o . In the context of RDF, s , p and o are URIs.

A *Path* π between nodes x and y over a graph database is defined as a sequence of triples (t_0, \dots, t_k) such that: (i) the subject of t_0 is x ; (ii) the object of t_k is y ; and (iii) for all $1 \leq i \leq k$, the object of t_{i-1} is the subject of t_i . The *Trace* of a path π is the sequence of edge labels (predicates) of t_0, \dots, t_k preserving the order of the triples.

The goal of a path query over a graph database is to identify paths inside the graph. There exist a number of proposals to define path queries in query languages. For instance, SPARQL defines them by using regular expressions. Other proposals include Nested Regular Expressions [10] or Context-Free Grammars [2, 4–6, 8, 12].

3 Recursive Expressions

In this section we define *recursive expressions*, which are an extension of regular expressions, supporting the definition of a subset of context-free languages.

Most modern programming languages extend regular expressions by adding support for recursion. They allow, for example, naming subexpressions and back-referencing to the values matching those. However, the use of such advanced features is rather complex and is not standardized among programming languages.

The purpose of our recursive expressions is to keep simplicity while still delivering desired levels of expressiveness. The syntax of recursive expressions is given by:

$$\begin{aligned} \text{exp} \rightarrow () \mid t \mid (\text{exp}) \mid \text{exp exp} \mid \text{exp|exp} \mid \text{exp} * \\ \mid \langle \text{exp}_1 \dots \text{exp}_n \rangle \text{exp} \langle \text{exp}'_n \dots \text{exp}'_1 \rangle \end{aligned}$$

Notice that recursive expressions include the ternary *recursion operator*: $\langle _ \rangle _ \langle _ \rangle$. Intuitively, the expressions exp_i and exp'_i define pairs of matching parenthesis.

The language defined by a recursive expression can be inductively defined as it is usual for regular expressions:

$$\begin{aligned}
 \mathcal{L}(\langle \rangle) &= \emptyset & \mathcal{L}(t) &= \{t\} \\
 \mathcal{L}(\langle e \rangle) &= \mathcal{L}(e) & \mathcal{L}(e_1 e_2) &= \mathcal{L}(e_1) \circ \mathcal{L}(e_2) \\
 \mathcal{L}(e_1 | e_2) &= \mathcal{L}(e_1) \cup \mathcal{L}(e_2) & \mathcal{L}(e^*) &= \cup_{i \in \mathbb{N}} \mathcal{L}(e)^i \\
 \mathcal{L}(\langle e_1 \cdots e_n \rangle e \langle e'_n \cdots e'_1 \rangle) &= \bigcup_{i=1}^n \{ \alpha^k \beta \alpha'^k \mid \alpha \in \mathcal{L}(e_i), \beta \in \mathcal{L}(e), \alpha' \in \mathcal{L}(e'_i), k \geq 0 \}
 \end{aligned}$$

Table 1. Examples of recursive expressions.

#	Recursive expression	Grammar	Language
(1)	$\langle a \rangle \langle b \rangle$	$S \rightarrow a S b$ $S \rightarrow \epsilon$	$a^n b^n$
(2)	$\langle ab \rangle \langle ab \rangle^*$	$S \rightarrow \epsilon$ $S \rightarrow S S_1$ $S_1 \rightarrow a S_1 b$ $S_1 \rightarrow b S_1 a$ $S_1 \rightarrow \epsilon$	Equal number of a 's and b 's
(3)	$\langle b \rangle a \langle bb \rangle$	$S \rightarrow b S b b$ $S \rightarrow a$	$b^n a b^{2n}$
(4)	$\langle sc t \rangle \langle sc \bar{sc} \rangle \mid \langle t \bar{t} \rangle \langle \bar{t} \bar{sc} \rangle$	$S \rightarrow sc S \bar{sc}$ $S \rightarrow t S \bar{t}$ $S \rightarrow sc \bar{sc}$ $S \rightarrow t \bar{t}$	Balanced pairs of sc (RDF <i>subClassOf</i>) and t (RDF <i>type</i>) edges [8]
(5)	$\langle sc \rangle \langle \bar{sc} \rangle \bar{sc}$	$S \rightarrow A \bar{sc}$ $A \rightarrow sc A \bar{sc}$ $A \rightarrow \epsilon$	Balanced pairs of sc (subClassOf) edges with extra \bar{sc} [8]
(6)	$a \langle a \rangle \langle b \rangle \langle c \rangle \langle d \rangle d$	$S \rightarrow a A d$ $A \rightarrow a A d$ $A \rightarrow B$ $B \rightarrow b B c$ $B \rightarrow \epsilon$	$a a^n b^m c^m d^n d$ (see [7])
(7)	$\langle \langle a \rangle b \langle c \rangle \rangle d \langle \langle e \rangle \langle f \rangle \rangle$	$S \rightarrow A S B$ $S \rightarrow d$ $A \rightarrow a A c$ $A \rightarrow b$ $B \rightarrow e B f$ $B \rightarrow \epsilon$	$(a^n b c^n)^k d (e^m f^m)^k$

Recursive expressions allow us to define a number of relevant context-free languages, including the ones found in [7,8]. In Table 1 we show some examples of recursive expressions, equivalent context-free grammars and their language. We represent a grammar by its set of production rules. In order to simplify the notation, we consider that the first non-terminal symbol appearing in a set of rules is the start symbol. The notation \bar{p} inverts the orientation of an edge, that is, if the tuple (s, p, o) is in the database, so is (o, \bar{p}, s) . It is normally implemented by explicitly adding those edges to the database.

The examples put in evidence two characteristics of recursive expressions: conciseness and legibility. These characteristics may improve the pragmatics of a query language.

Notice that most of these examples are subsets of Dyck languages (in the sense that they define a language of balanced parentheses). In this way, we can see that recursive expressions describe a subset of the class of Dyck languages. Notice that the languages described by the recursive expressions (2) and (7) are not Dyck languages. On the other hand, there are context-free languages that cannot be expressed by recursive expressions.

3.1 Obtaining a Grammar from a Recursive Expression

Let us now define how a grammar can be obtained from a given recursive expression. In the following rules, we inductively define the function φ , taking a recursive expression and producing a context-free grammar G , generating the same language.

The grammars corresponding to the recursive expressions $()$ and a generate, respectively, the empty language and the language $\{a\}$:

$$\frac{S \text{ is new}}{\varphi(\epsilon) = \{S \rightarrow \epsilon\}}$$

$$\frac{a \text{ is a terminal symbol, } S \text{ is new}}{\varphi(a) = \{S \rightarrow a\}}$$

The recursive expression (E) can be associated with a grammar that generates the same language as E :

$$\frac{\varphi(E_1) = \{S_1 \rightarrow \alpha, \dots\}, S \text{ is new}}{\varphi((E_1)) = \{S \rightarrow S_1\} \cup \varphi(E_1)}$$

The grammars associated to sequential, alternative and Kleene star operators are computationally defined by adding a new start symbol and rules for concatenation, choice and repetition of strings:

$$\frac{\varphi(E_1) = \{S_1 \rightarrow \alpha, \dots\}, \quad \varphi(E_2) = \{S_2 \rightarrow \beta, \dots\}, \quad S \text{ is new}}{\varphi(E_1 E_2) = \{S \rightarrow S_1 S_2\} \cup \varphi(E_1) \cup \varphi(E_2)}$$

$$\frac{\varphi(E_1) = \{S_1 \rightarrow \alpha, \dots\}, \quad \varphi(E_2) = \{S_2 \rightarrow \beta, \dots\}, \quad S \text{ is new}}{\varphi(E_1 | E_2) = \{S \rightarrow S_1, S \rightarrow S_2\} \cup \varphi(E_1) \cup \varphi(E_2)}$$

$$\frac{\varphi(E_1) = \{S_1 \rightarrow \alpha, \dots\}, \quad S \text{ is new}}{\varphi(E_1^*) = \{S \rightarrow S_1 \ S, S \rightarrow \epsilon\} \cup \varphi(E_1)}$$

The rule for obtaining a grammar that generates the language for the expression $\langle E_1 \dots E_n \rangle E \langle E'_n \dots E'_1 \rangle$ proceeds by: (i) generating context-free grammars for all the sub-expressions; (ii) building the set R , formed by the union of all these grammars and the set of rules $\{S \rightarrow S_i \ S'' \ S'_i \mid i \in \{1, \dots, n\}\}$. The rules in this set define the matching of strings generated by each pair of expressions E_i and E'_i :

$$\begin{aligned} \varphi(E_i) &= \{S_i \rightarrow \alpha_i, \dots\} && \text{for all } i \in \{1, \dots, n\}, \\ \varphi(E'_i) &= \{S'_i \rightarrow \alpha'_i, \dots\} && \text{for all } i \in \{1, \dots, n\}, \\ \varphi(E) &= \{S'' \rightarrow \beta, \dots\}, && S \text{ is new} \\ R &= \varphi(E_1) \cup \dots \cup \varphi(E_n) \cup \varphi(E'_1) \cup \dots \cup \varphi(E'_n) \\ \hline \varphi(\langle E_1 \dots E_n \rangle E \langle E'_n \dots E'_1 \rangle) &= \{S \rightarrow S''\} \cup \{S \rightarrow S_i \ S \ S'_i \mid i \in \{1, \dots, n\}\} \cup R \end{aligned}$$

The grammar defined by a recursive expression will be used in the next sections to define the semantics of rcfSPARQL, a query language inspired by SPARQL.

4 A Query Language Containing Recursive Expressions

In this section we present rcfSPARQL, a query language that uses recursive expressions to build non-regular property paths.

The syntax of rcfSPARQL adapts cfSPARQL by using recursive expressions to define property paths (instead of non-terminal symbols used in cfSPARQL). This eliminates the need to define a context-free grammar in the query. First, we define the syntax of triple and graph patterns and then we will use them to build queries in rcfSPARQL.

Definition 1 (Restricted Context-Free Triple Pattern). *A restricted context-free triple pattern \mathcal{T} is a tuple of the form (qs, qp, qo) , where qs, qo are literals or variables and qp is a literal, variable or recursive expression over literals. \square*

The next definition presents Restricted Context-Free Graph Patterns, used to combine restricted context-free triple patterns. Restricted graph patterns correspond to SPARQL operators, being the basis for building queries, by handling and combining sets of tuples.

Definition 2 (Restricted Context-Free Graph Pattern). *A restricted context-free graph pattern \mathcal{P} is a graph pattern built from restricted context-free triple patterns and SPARQL operations, in accordance with the following grammar:*

$$\begin{aligned} \mathcal{P} &\rightarrow \mathcal{T} \mid \mathcal{P} . \mathcal{P} \mid \mathcal{P} \text{ OPT } \mathcal{P} \mid \mathcal{P} \text{ UNION } \mathcal{P} \mid \mathcal{P} \text{ FILTER } \mathcal{E} \\ \mathcal{E} &\rightarrow \text{bound}(?x) \mid ?x \text{ op } l \mid ?x \text{ op } ?y \mid \neg \mathcal{E} \mid \mathcal{E} \wedge \mathcal{E} \mid \mathcal{E} \vee \mathcal{E} \end{aligned}$$

where $?x, ?y$ are variables, l is a literal, $\text{op} \in \{<, \leq, >, \geq, =, \neq\}$, and \mathcal{T} is a restricted context-free triple pattern. \square

We define a *Restricted Context-Free SPARQL* (rcfSPARQL) query as follows:

Definition 3 (Restricted Context-Free Graph Query). *Given an RDF data graph D and a list of variable identifiers $?x_1, \dots, ?x_n$, a restricted context-free graph query Q is defined as:*

$$Q \rightarrow \text{SELECT } ?x_1, \dots, ?x_n \text{ FROM } D \text{ WHERE } \mathcal{P}$$

where \mathcal{P} is a restricted context-free graph pattern. □

The following example (adapted from [8]) illustrates the use of recursive expressions in an rcfSPARQL query.

Example 1. Let D be the database in Fig. 1. It contains information about a company’s employees. We rewrite the query in [8] that selects the employees having have the same job, but different salaries. The query is written in rcfSPARQL as:

```

1 SELECT ?job, ?emp1, ?sal1, ?emp2, ?sal2
2 FROM D
3 WHERE {
4   ?emp1 <boss><boss> ?emp2 .
5   ?emp1 job ?job .
6   ?emp2 job ?job .
7   ?emp1 salary ?sal1 .
8   ?emp2 salary ?sal2 .
9   FILTER (?sal1 > ?sal2)
10 }
```

This query defines a relation formed by 5-tuples. The variables at line (1) define the attributes of this relation. The property path at line (4) defines a path between pairs of employees ($?emp1$ and $?emp2$). Notice that this path uses a recursive expression to look for paths between employees at the same level of the hierarchy. These paths will be formed by nested `boss` and `boss` edges.

Lines (5–8) of the query looks, respectively, for the jobs and salaries of each pair of employees identified in line (4). Notice that there is just one variable $?job$, denoting that both employees have the same position in the company. Line (9) filters the pairs of employees that have the same job but different salaries. □

5 Answering rcfSPARQL Queries

This section presents a formal semantics for rcfSPARQL. We adapt the operational semantics given in [8] to consider restricted context-free graph patterns.

Similarly to other declarative query languages such as SPARQL [11] and SQL [3], queries in rcfSPARQL rely upon the notion of *relations*. We use the same RDF data representation as in [8]. This representation defines a three-level tree for each relational table. In this tree, the root node represents the whole table. On the second level there is a node for each line of the table (these nodes

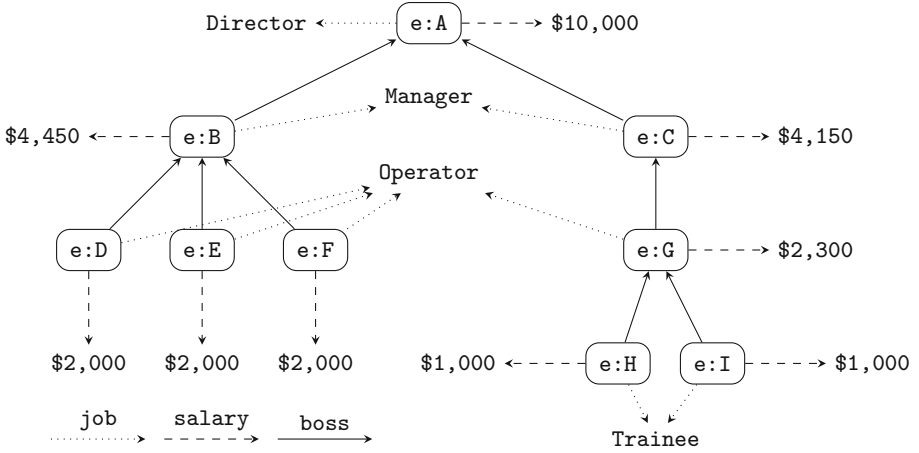


Fig. 1. Example of hierarchy database D [8].

are linked to the root by **rec**-labeled edges). The information stored in the table appears on the leaves. Each value is linked to its corresponding record node by using edges labeled according to a column of the table.

Let us now define the function $\varphi(\mathcal{P})$ as an extension of the function φ for recursive expressions. The function $\varphi(\mathcal{P})$ takes a restricted context-free graph pattern and returns a new context-free graph pattern, similar to \mathcal{P} , but replacing each recursive expression for one (literal) non-terminal symbol of G , generating the same language as the recursive expression. The function φ over patterns is defined as follows:

$$\frac{gp \text{ is a terminal symbol or query variable}}{\varphi((gs, gp, go)) = (\emptyset, (gs, gp, go))}$$

$$\frac{E \text{ is a recursive expression, } \{S \rightarrow \alpha, \dots\} = \varphi(E)}{\varphi((gs, E, go)) = (\varphi(E), (gs, S, go))}$$

$$\frac{\oplus \in \{\text{AND, OPT, UNION}\}, (G_1, \mathcal{P}'_1) = \varphi(\mathcal{P}_1), (G_2, \mathcal{P}'_2) = \varphi(\mathcal{P}_2)}{\varphi(\mathcal{P}_1 \oplus \mathcal{P}_2) = (G_1 \cup G_2, \mathcal{P}'_1 \oplus \mathcal{P}'_2)}$$

$$\frac{(G, \mathcal{P}') = \varphi(\mathcal{P})}{\varphi(\mathcal{P} \text{ FILTER } \mathcal{E}) = (G, \mathcal{P}' \text{ FILTER } \mathcal{E})}$$

The operational semantics for rcfSPARQL is similar to cfSPARQL defined in [8] and is not reproduced here due to lack of space. The semantic rules in [8] build an RDF graph tree containing one record for each solution of the query.

The only difference is on query evaluation, whose semantics for rcfSPARQL we present below. The query to be evaluated is given in double brackets. In order to answer a restricted context-free path query, our method proceeds in a similar

manner as described in [8]. We have included a step to obtain a grammar G from the restricted context-free pattern \mathcal{P} . Also, we have removed steps specific to the algorithm presented in [8], such as the construction of the predictive parsing table T_G . These steps are described as follows:

$$\begin{array}{l} (G, \mathcal{P}') = \varphi(\mathcal{P}) \quad H = \{(a, N) \mid a \in \text{Nodes}(D), N \text{ is a nonterminal of } G\} \\ D_1 = \text{eval}(G, D, H) \quad D_2 = D \cup D_1 \quad (r, D_3) = \llbracket \mathcal{P}' \rrbracket_{D_2} \\ \text{Answer} = \{(a_1, \dots, a_n) \mid (r, \text{rec}, o), (o, ?x_1, a_1), \dots, (o, ?x_n, a_n) \in D_3\} \\ \hline \llbracket \text{SELECT } ?x_1, \dots, ?x_n \text{ FROM } D \text{ WHERE } \mathcal{P} \rrbracket = \text{Answer} \end{array}$$

The semantics presented in the rule above is explained next. We will use Example 1 to show the application of each step in obtaining the answer to the query. These steps are as follows:

1. *Obtaining a context-free grammar:* A grammar G is built from the restricted context-free graph pattern \mathcal{P} by using the algorithm provided in Sect. 3.1. A new pattern \mathcal{P}' is also built, to replace the recursive expressions in \mathcal{P} for non-terminal symbols of the grammar G describing the same language.

For the query of Example 1, the recursive expression $\langle \text{boss} \rangle \langle \overline{\text{boss}} \rangle$ (line 4) is used to define the pair formed by the grammar $\{S \rightarrow \text{boss } \overline{\text{boss}}, S \rightarrow \epsilon\}$ and the pattern $(?emp1, S, ?emp2)$.

2. *Building a set of pairs:* The set H is built to contain all pairs formed by nodes of the RDF data graph and non-terminal symbols of G ;

In Example 1, $H = \{(e:A,S), (e:B,S), (e:C,S), (e:D,S), (e:E,S), (e:F,S), (e:G,S), (e:H,S), (e:I,S)\}$.

3. *Decoration of the data graph:* In this step we use a context-free path query engine to process the set of pairs H . The function $eval$ may be replaced by any CFPQ evaluation engine used to create new edges in D (proposals may be found in [2, 5, 6, 8]). Each new edge (x, S, y) indicates that there is an S -generated path between x and y .

The decorated graph from Example 1 can be seen in Fig. 2, where double ended edges mean two edges in the forward and backward senses.

4. *Query answering over the decorated graph:* In this step, the pattern \mathcal{P}' is used to build the answer to the query over the decorated data graph D_2 (built in the previous step). The RDF graph D_3 is built using the semantics of the pattern \mathcal{P}' , as defined in [8].

The graph D_3 for our Example 1 is depicted in Fig. 3a.

5. *Selection of the graph nodes that answer the query:* The set *Answer* is formed by the leaves of the RDF graph D_3 , organized as tuples that correspond to each record, and ordered as required by the query.

In Example 1, $Answer = \{(Manager, e:B, \$4450, e:C, \$4150), (Operator, e:G, \$2300, e:D, \$2000), (Operator, e:G, \$2300, e:E, \$2000), (Operator, e:G, \$2300, e:F, \$2000)\}$. See Fig. 3b.

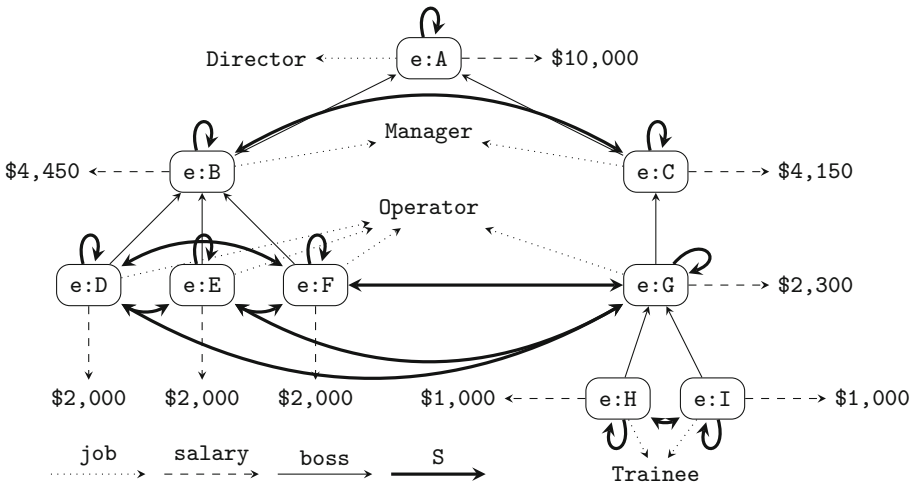
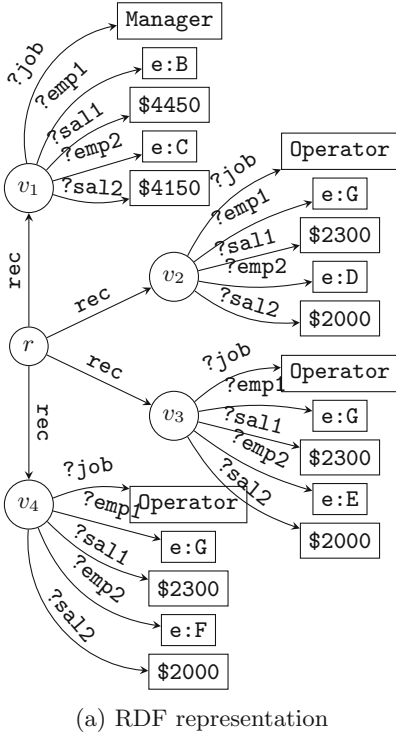


Fig. 2. Decorated graph.



?job	?emp1	?sal1	?emp2	?sal2
Manager	e:B	\$4450	e:C	\$4150
Operator	e:D	\$2000	e:E	\$2000
Operator	e:F	\$2000		

(b) Tabular representation

Fig. 3. RDF and tabular representations [8] of the answer for the query from Example 1.

6 Conclusions and Future Work

In this paper we presented rcfSPARQL, a language for the specification of a subset of CFPQs over graph databases in RDF. The proposed language is built around recursive expressions, which extends regular expressions in order to encompass a meaningful subset of context-free languages while keeping the specification of queries easy and concise. Both the syntax and semantics of rcfSPARQL are provided, as well as a running example. Next we present some topics for future work.

To implement rcfSPARQL, it suffices to extend the syntax of SPARQL to include the new operators and to make use of some CFPQ evaluation algorithm. Many algorithms for the evaluation of CFPQ are available in the literature. Open-source SPARQL engines may serve as a base for such an implementation. This will allow us to evaluate rcfSPARQL in real scenarios, like data science lifecycle provenance [9].

For simplicity, we have not included convenient operators like ‘+’ or ‘?’ present in regular expressions nor have we defined similar versions of them for

our recursive operators. Although they can be considered syntactic sugar, these operators make expressions more concise, and therefore improve readability.

Recursive expressions are user-friendlier than context-free grammars for writing queries. However, we know that context-free grammars are more expressive than recursive expressions. Thus, a relevant future direction is to formalize how recursive expressions and context-free grammars compare in terms of expressiveness. Currently, we are investigating the expressiveness of our query specification language by taking into account the families of context-free languages defined in [15].

Acknowledgements. This work is partly supported by INES grant CNPq/465614/2014-0 (Brazil) and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brazil (CAPES) - Finance Code 001.

References

1. Abiteboul, S., Hull, R., Vianu, V.: *Foundations of Databases*. Addison-Wesley, Boston (1995)
2. Azimov, R., Grigorev, S.: Context-free path querying by matrix multiplication. In: *Proceedings of the 1st ACM SIGMOD Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA) GRADES-NDA 2018*. Association for Computing Machinery, New York (2018). <https://doi.org/10.1145/3210259.3210264>
3. Coronel, C., Morris, S.: *Database Systems: Design, Implementation, & Management*. Cengage Learning, Boston (2016)
4. Grigorev, S., Ragoza, A.: Context-free path querying with structural representation of result. In: *Proceedings of the 13th Central & Eastern European Software Engineering Conference in Russia CEE-SECR 2017*. Association for Computing Machinery, New York (2017). <https://doi.org/10.1145/3166094.3166104>
5. Hellings, J.: Conjunctive context-free path queries. In: Schweikardt, N., Christophides, V., Leroy, V. (eds.) *Proceedings of 17th International Conference on Database Theory (ICDT)*, Athens, Greece, 24–28 March 2014, pp. 119–130. OpenProceedings.org (2014). <https://doi.org/10.5441/002/icdt.2014.15>
6. Hellings, J.: Path results for context-free grammar queries on graphs. *CoRR abs/1502.02242* (2015)
7. Kuijpers, J., Fletcher, G., Yakovets, N., Lindaaker, T.: An experimental study of context-free path query evaluation methods. In: *Proceedings of the 31st International Conference on Scientific and Statistical Database Management*, pp. 121–132. ACM (2019)
8. Medeiros, C.M., Musicante, M.A., Costa, U.S.: LL-based query answering over RDF databases. *J. Comput. Lang.* **51**, 75–87 (2019). <https://doi.org/10.1016/j.cola.2019.02.002>. <http://www.sciencedirect.com/science/article/pii/S1045926X18301915>
9. Miao, H., Deshpande, A.: Understanding data science lifecycle provenance via graph segmentation and summarization. In: *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pp. 1710–1713. IEEE (2019)

10. Pérez, J., Arenas, M., Gutierrez, C.: nSPARQL: a navigational language for RDF. *Web Semant.: Sci. Serv. Agents WWW* **8**(4), 255–270 (2010). <https://doi.org/10.1016/j.websem.2010.01.002>. <http://www.sciencedirect.com/science/article/pii/S157082681000003X>. Semantic Web Challenge 2009 User Interaction in Semantic Web research
11. Prud'hommeaux, E., Seaborne, A.: SPARQL query language for RDF, January 2008. <http://www.w3.org/TR/rdf-sparql-query/>. <http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/>
12. Santos, F.C., Costa, U.S., Musicante, M.A.: A bottom-up algorithm for answering context-free path queries in graph databases. In: Mikkonen, T., Klamma, R., Hernández, J. (eds.) ICWE 2018. LNCS, vol. 10845, pp. 225–233. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91662-0_17
13. W3C: SPARQL 1.1 query language (2012). <https://www.w3.org/TR/2012/PR-sparql11-query-20121108/>
14. W3C: RDF - semantics web standards (2014). <https://www.w3.org/RDF/>
15. Yntema, M.: Inclusion relations among families of context-free languages. *Inf. Control* **10**(6), 572–597 (1967). [https://doi.org/10.1016/S0019-9958\(67\)91032-7](https://doi.org/10.1016/S0019-9958(67)91032-7). <http://www.sciencedirect.com/science/article/pii/S0019995867910327>



Healthcare Decision-Making Over a Geographic, Socioeconomic, and Image Data Warehouse

Guilherme M. Rocha, Piero L. Capelo, and Cristina D. A. Ciferri^(✉)

Universidade de São Paulo, São Carlos, Brazil
{guilherme.muzzi.rocha,piero.capelo}@usp.br, cdac@icmc.usp.br

Abstract. Geographic, socioeconomic, and image data enrich the range of analysis that can be achieved in the healthcare decision-making. In this paper, we focus on these complex data with the support of a data warehouse. We propose three designs of star schema to store them: jointed, split, and normalized. We consider healthcare applications that require data sharing and manage huge volumes of data, where the use of frameworks like Spark is needed. To this end, we propose SimSparkOLAP, a Spark strategy to efficiently process analytical queries extended with geographic, socioeconomic, and image similarity predicates. Performance tests showed that the normalized schema provided the best performance results, followed closely by the jointed schema, which in turn outperformed the split schema. We also carried out examples of semantic queries and discuss their importance to the healthcare decision-making.

1 Introduction

There is a huge volume of healthcare data generated by different sources, such as hospital information systems, sensors, medical devices, websites, and medical applications [16]. The analysis of data from these heterogeneous sources can provide several benefits for the healthcare decision-making. For instance, it is possible to generate knowledge that can be used to identify trends in healthcare, combat social and health inequalities, and provide new ideas about science [9].

Data warehousing can provide support for the healthcare decision-making. It stores data from autonomous, distributed, and heterogeneous sources in the data warehouse (DW). Besides being integrated, data in the DW are subject-oriented, non-volatile, historical, and multidimensional [8]. Analytical queries, called OLAP (on-line analytical processing), are issued against the DW to discover useful trends. In relational implementations, data are usually modeled

Supported by the São Paulo Research Foundation (FAPESP), the Brazilian Federal Research Agency CNPq, and the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, Brasil (CAPES), Finance Code 001. G.M.R. and C.D.A.C. acknowledge support from FAPESP grants #2018/10607-3 and #2018/22277-8, respectively.

through a star schema, where a central fact table is linked to satellite dimension tables. This requires the processing of star join operations in OLAP queries.

The organization of the DW benefits the investigation of subjects of interest considering different analysis factors. Regarding the healthcare sources, they provide several types of data, including conventional data (e.g., numeric, alphanumeric, and date types), socioeconomic data (e.g., population by age range and household by type), geographic data (e.g., a city represented by a point object that is defined by its latitude and longitude), and image data (e.g., feature vectors that describe the intrinsic characteristics of images). In this paper, we consider these heterogeneous types of data in the healthcare decision-making. Example 1 illustrates a case study that motivates the development of our work.

Example 1. Consider a DW that integrates healthcare data related to exams. Conventional and image data from these exams were collected over several years, and refer to patients treated in different hospitals. The hospitals are located in several cities, each one described by conventional data, geographic data representing its latitude and longitude, and socioeconomic data.

The following queries can be issued against the healthcare DW. (Q_1) “How many exams from 1970 to 2010 have similar images to a given image of breast cancer?”; (Q_2) “How many exams have similar images to a given image of breast cancer from female patients that were captured in private hospitals located in cities within a 15 km radius of New York?” (Q_3) “How many exams have similar images to a particular image of breast cancer from patients of different range ages and states, and that were captured in hospitals located in cities within a 1,500 km radius of New York and whose population has a age range similar to the age range of New York?”

Query Q_1 contains conventional attributes from exams and their dates. It also requires the management of a similarity search over the image data. Query Q_2 has conventional attributes from exams, patients, and hospitals, as well as includes a similarity search over the geographic data. Query Q_3 extends Q_2 by defining a similarity search over the socioeconomic data. Figure 1 depicts the considered similarity factors. Comparing image, geographic, and socioeconomic data in OLAP queries enriches the range of analysis that can be performed. □

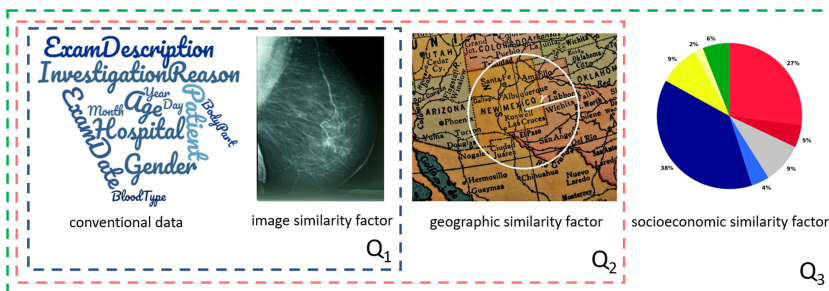


Fig. 1. Image, geographic, and socioeconomic data as similarity factors

Processing OLAP queries over a geographic, socioeconomic, and image DW demands a high computational cost. This DW stores several types of complex data and may be fed from applications in big data, indicating that its volume may be orders of magnitude larger than megabytes [9]. Further, in addition to the expensive star join operations, the OLAP query processing should be extended with similarity search predicates, which also require the processing of costly operations to calculate the distance between images [18] and the similarity between points and between socioeconomic data [19]. Applications that exceed the megabyte order and involve costly operations usually face performance issues, thus requiring storage and processing capacity to guarantee scalability [3]. This demands the use of parallel and distributed data processing frameworks, such as the Apache Hadoop¹ and the Apache Spark². These frameworks enable data sharing and robust decision-making in healthcare [16].

A core issue in the healthcare decision-making is to support the execution of analytical queries over socioeconomic, geographic, and image DWs. In this paper, we investigate this issue by introducing the following contributions. We propose three designs of star schema that contain image, geographic, and socioeconomic data as similarity factors. We also introduce SimSparkOLAP, a Spark strategy to efficiently process OLAP queries extended with these similarity search predicates. Furthermore, using SimSparkOLAP, we investigate how the proposed star schemas affect the extended query processing. We also execute semantic queries and discuss their importance to the healthcare decision-making.

This paper is organized as follows. Section 2 reviews related work, Sect. 3 describes the theoretical foundation, Sect. 4 proposes the star schemas, Sect. 5 introduces the SimSparkOLAP strategy, Sect. 6 discusses the experimental evaluation, and Sect. 7 concludes the paper.

2 Related Work

Our work integrates several research areas: image DW, geographic DW, processing of star joins in parallel and distributed data processing frameworks, and processing of similarity search over complex data. Although there are approaches proposed in the literature that investigate individually each one of these research areas, none of them consider all aspects in the same setting, which we do.

A visual data cube and a multidimensional OLAP tool of images collections is described in [7]. The imageDWE, introduced in [17], is a data warehousing environment that enables the storage of intrinsic characteristics of medical images in a image DW and supports analytical queries extended with similarity search over this image DW. In our work, we use the star schema of the image DW as a basis for our proposal; thus, we describe it in Sect. 3. However, [7, 17] do not investigate geographic and socioeconomic data or the use of Hadoop and Spark.

There are solutions for the storage of geographic data in DWs and the processing of SOLAP (spatial OLAP) queries (e.g., [4, 20]). Here, we borrow

¹ <https://hadoop.apache.org/>.

² <https://spark.apache.org/>.

from [20] the use of pictograms to indicate geographic dimension tables. Differently from our work, [4,20] do not include socioeconomic and image similarity factors or the use of parallel and distributed data processing frameworks. Also, they process different SOLAP queries, which consider topological relationships. Here, we are interested in processing similarity search over the geographic data.

Another research area of great interest is the processing of star joins in Hadoop and Spark (e.g., [1,2]). But, these approaches only improve star joins over conventional data. In a previous work [15], we introduce BrOmnImg, which process, in Spark, analytical queries extended with a similarity search predicate over the imageDWE. Because our proposed strategy extends BrOmnImg, we describe this previous method in Sect. 3. Differentials of our work are the proposal of different star schemas and the fact that SimSparkOLAP also processes similarity queries over geographic and socioeconomic data.

The processing of similarity search over images in Hadoop and Spark is investigated in [10,12], respectively. But, they do not consider the support of a DW or the processing of geographic and socioeconomic data. The SimilarQL framework [19] provides a complete and flexible set of similarity query operators to explore datasets of complex data in traditional database systems. Here, we borrow from [19] the idea that complex data seldom benefits from exact comparisons, since they are mostly evaluated in exploratory data analysis tasks. We also borrow the idea that these tasks seldom require the reuse of previous indices.

3 Theoretical Foundation

Similarity Search. It is usually inadequate to use ordinary relational operators (i.e. $<$, \leq , $>$, \geq) to compare complex data [19]. Instead, image, geographic, and socioeconomic data considered in this work should be compared through similarity search. Similarity searches are performed using a distance function, which calculates the dissimilarity between two complex elements based on their feature vectors. A distance function becomes smaller as the elements become more similar. A feature vector describes a given characteristic of interest. For instance, feature vectors generated by image extractors, such as Color Histograms [5] and Haralick descriptors [6], contain the numeric representations of these images according to the attributes of color and texture, respectively.

Consider a set S of feature vectors describing a characteristic and a distance function d . The range query calculates which elements of the dataset are similar to a given query element s_q considering a given query radius r_q . That is, the range query retrieves every element $s_i \in S$ that satisfies the condition $d(s_i, s_q) \leq r_q$. Because the range query calculates the similarity between s_q and each s_i , it has high computational cost. The Omni technique [18] drastically decreases this cost. It stores the distances between each s_i and strategically positioned elements called *foci* that are used to improve the prunability during query processing. In the filtering step, it is defined a region that identifies candidate elements to answer the query. Then, in the refinement step, distance calculations are performed only over these candidate elements to eliminate false positives.

Image DW. The star schema of the image DW [17] is composed of a fact table and conventional dimension tables. It also includes image dimension tables, i.e., one *Img Feature Vector* table and several *Perceptual Layer* tables. The Feature Vector table stores all feature vectors of all images, independent of the image extractor. For instance, if a image was processed using a *Color Histogram* extractor and a *Haralick* descriptor, then two feature vectors are stored for this image. Each Perceptual Layer table refers to a specific extractor, and stores the distances between each image and each one of the *foci* elements determined by the Omni technique for this extractor. These tables are illustrated in Example 2.

The BrOmnImg Method. The Hadoop Distributed File System (HDFS) handles large data sets running on commodity hardware. It divides the data file into blocks, distributing and replicating these blocks on nodes of a cluster. Spark was designed to read and write data from and to HDFS. This parallel and distributed data processing framework is based on in-memory computation and on the Resilient Distributed Dataset (RDD) abstraction [21]. The BrOmnImg method [15] provides an efficient processing of analytical queries extended with a similarity search predicate over images in Spark. To this end, BrOmnImg uses the broadcast join technique [1] to process star joins, i.e., it assumes that all conventional dimension tables are small enough to fit in the main memory; then, these tables are sent to all nodes of the cluster to compute the joins. BrOmnImg also integrates the Omni technique to reduce the number of distance calculations.

4 The Proposed Star Schemas

In this section, we propose three designs of star schema that contain image, geographic, and socioeconomic data as similarity factors (Fig. 2). To this end, we extend the image DW, described in Sect. 3 and detailed in Example 2, to also consider the remaining similarity factors.

Example 2. The proposed star schemas consider the case study described in Example 1. We borrow the following tables from the image DW: (i) the fact table *Exam*; (ii) the conventional dimension tables *ExamDate*, *ExamDescription*, *Patient*, and *Age*; (iii) the feature vector table *ImgFeatureVector*; and (iv) the perceptual layer tables *Color Histogram* and *Haralick*. Except for *Exam*, the attributes of the remaining tables are omitted due to space limitations. □

To consider geographic and socioeconomic data, we also store data related to the cities where the hospitals that performed the exams are located. The geographic similarity factor, which is defined by a point and depicted by the pictogram ●, is represented by feature vectors composed of the attributes *Latitude* and *Longitude*. The socioeconomic similarity factor, which contains several cities characteristics like age range, ethnicity, and household, is represented by feature vectors composed of the attributes *SocioFeat*₁, ..., *SocioFeat*_{*n*}. The proposed schemas differ on how they store these similarity factors, as described as follows.

The Jointed Schema (Fig. 2a). In this schema, the geographic and socioeconomic data are stored in the similarity dimension table *Hospital*. In addition to

the conventional attributes, this table contains the geographic and socioeconomic feature vectors of cities. Only *Hospital* is related to the fact table *Exam*.

The Split Schema (Fig. 2b). In this schema, the geographic and socioeconomic data are stored in the dimension tables *City* and *Socioeconomic*, respectively. Thus, the conventional dimension table *Hospital* only stores its conventional data, the similarity dimension table *City* stores the geographic feature vectors of cities, and the similarity dimension table *Socioeconomic* stores the socioeconomic feature vectors of cities. Each table, i.e., *Hospital*, *City*, and *Socioeconomic*, is related to the fact table *Exam*.

The Normalized Schema (Fig. 2c). In this schema, the geographic and socioeconomic data are stored in the dimension tables *City* and *Socioeconomic*, respectively. Thus, each table only stores its corresponding attributes. Here, the similarity dimension table *Socioeconomic* is related to the similarity dimension table *City*, which is related to the conventional dimension table *Hospital*, which in turn is related to the fact table *Exam*. That is, one table is linked to another respecting the granularity of the attributes.

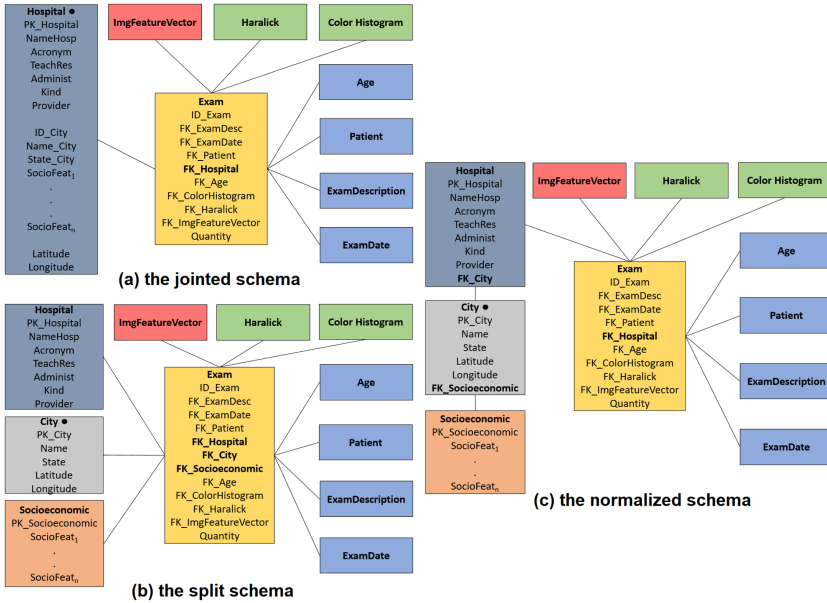


Fig. 2. The proposed designs of star schema: jointed, split, and normalized

5 The Proposed SimSparkOLAP Strategy

In this section, we propose SimSparkOLAP, a strategy to efficiently process OLAP queries extended with geographic, socioeconomic, and image similarity search predicates in Spark.

Before describing SimSparkOLAP, we give the notions behind our target OLAP queries. They are defined according to the following predicates: (i) conventional, based on selection conditions; (ii) image similarity factor, composed of a similarity operation and one or more perceptual layers; (iii) geographic similarity factor, specified in terms of a similarity operation; (iv) socioeconomic similarity factor, composed of a similarity operation and one or more socioeconomic features. For instance, for query Q_3 defined in Example 1: (i) *breast cancer* represent the conventional predicate (ii) range query and the color perceptual layer represent the image predicate; (iii) range query represents the geographic predicate; (iv) range query and age range represent the socioeconomic predicate.

SimSparkOLAP performs several tasks to process the target OLAP queries, as depicted in Fig. 3. Each conventional dimension table i ($1 \leq i \leq m$) involved in the conventional predicate is accessed to process the selection conditions that apply over its data. SimSparkOLAP stores each generated set i of results in a specific structure *HashMapConventional_i* (Fig. 3a).

For the image similarity factor, the similarity operation is applied over all the required perceptual layer tables. In the filtering step, the data of each table j ($1 \leq j \leq n$) are filtered using the Omni technique, generating a set j of candidate results that is stored in a specific structure *HashMapFiltering_j* (Fig. 3b). Then, the candidate results of these sets are analyzed in the refinement step to eliminate false positives; here the *ImgFeatureVector* table is accessed and the distances are calculated to determine those results that comply with the search. The final set of results is stored in the structure *HashMapRefinement* (Fig. 3c).

SimSparkOLAP processes the geographic and socioeconomic similarity factors as described as follows, depending on the proposed star schema (Fig. 2).

The Jointed Schema (Fig. 2a). The geographic and socioeconomic similarity factors are processed against the similarity dimension table *Hospital* together with the selection conditions that apply over this table. The results are stored in the structure *HashMapHosGeoSocio* (Fig. 3h). Finally, all the structures are broadcasted to all nodes of the cluster, where the extended star join operation is performed (Fig. 3i).

The Split Schema (Fig. 2b). The geographic and socioeconomic similarity factors are processed against the similarity dimension tables *City* and *Socioeconomic*, respectively. The sets of results are stored in their related structures *HashMapGeographic* (Fig. 3e) and *HashMapSocioeconomic* (Fig. 3d). Also, the conventional conditions defined over *Hospital* are processed and the results are stored in the structure *HashMapHosGeoSocio* (Fig. 3h). Finally, all the structures are broadcasted to all nodes of the cluster, where the extended star join operation is performed (Fig. 3i).

The Normalized Schema (Fig. 2c). The socioeconomic similarity factor is processed against the similarity dimension table *Socioeconomic* and the results are stored in the structure *HashMapSocioeconomic*. This structure is then associated to the similarity dimensional table *City* (Fig. 3f) where the geographic similarity factor is also considered, generating new results that are stored in the

structure *HashMapGeographic*. Next, this structure is associated to the conventional table *Hospital* (Fig. 3g) where the conventional conditions over this table are also carried out. The results are stored in the structure *HashMapHosGeoSocio* (Fig. 3h). Finally, all the structures are broadcasted to all nodes of the cluster, where the extended star join operation is performed (Fig. 3i).

According to Fig. 3, SimSparkOLAP performs the following tasks to process OLAP queries extended with geographic, socioeconomic, and image similarity factors. For the jointed schema: (a), (b), (c), (h), (i); for the split schema, (a), (b), (c), (e), (d), (h), (i); and for the normalized schema: (a), (b), (c), (f), (g), (h), (i).

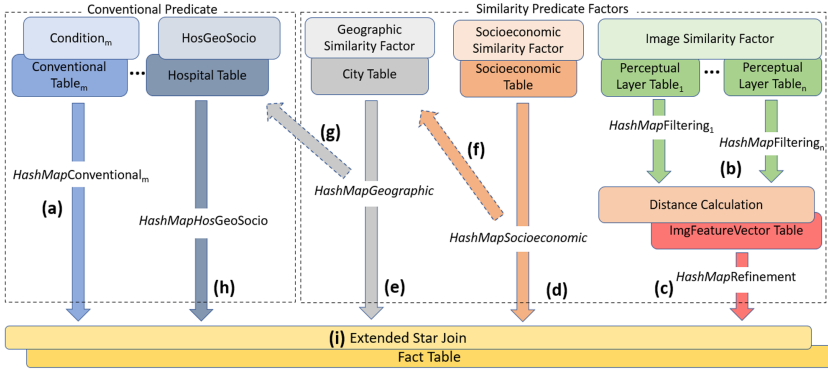


Fig. 3. General view of the SimSparkOLAP method

6 Experimental Evaluation

We conducted an experimental study based on a healthcare DW to assess the advantages of the proposed star schemas and method. The goal of our experiments was twofold: (a) determine the effect of the different schemas on the processing of OLAP queries extended with similarity search predicates; and (b) investigate the importance of these queries in the healthcare decision-making.

We used the *ImgDW* Generator tool [14] to populate the fact table, the conventional dimension tables and the tables that store the image similarity factor. The tool generates real data from medical images and synthetic data for the remaining tables. The number of tuples stored in each table was: Exam: 30 millions; ExamDate: 18,268; ExamDescription: 3 millions; Patient: 300,000; Age: 121; Color Histogram: 3 millions; Haralick: 3 millions; and *ImgFeatureVector*: 3 millions. We also used real data from 25,000 US cities from the Census dataset from year 2000³ to populate the tables related to the geographic and socioeconomic similarity factors. These data consider a very large country, with a large

³ <https://www.census.gov/programs-surveys/decennial-census/decade.2000.html>.

number of cities and an average of four hospitals per city. In the medical context, this data volume is a good representation of reality. Considering the normalized schema, the number of tuples stored in the tables *Hospital*, *City*, and *Socioeconomic* was, respectively, 100,000, 25,000 pairs of latitude and longitude, and 25,000 sets of 95 features.

The experiments were performed in a cluster with 5 nodes. Each node had, at least, 3 GB of RAM. We collected the elapsed time in seconds, which was recorded issuing each query 10 times, removing outliers, and calculating the average time. All cache and buffers were flushed after finishing each query.

Investigating the Proposed Star Schemas. We use SimSparkOLAP to investigate the effect of storing the geographic and socioeconomic similarity factors according to the schemas split, jointed, and normalized. We do not investigate the image similarity factor because we borrow the design of this factor from the image DW and describe preliminary results considering only this factor in our previous work [15].

The OLAP queries extended with geographic and socioeconomic similarity factors were defined considering combinations of the following conditions: (i) conventional condition: *Provider* = ‘Private’ for hospitals; (ii) geographic similarity condition: *cities within a given radius*; (iii) socioeconomic similarity condition: *age range distribution*. The range query was used as the similarity operation. We applied the Euclidean distance function for the image and socioeconomic factors and the GCDist function for the geographic factor. The combinations generated three configurations, depending on the conditions: (i) HosGeo, with conventional and geographic conditions; (ii) GeoSocio, with geographic and socioeconomic conditions; and (iii) HosGeoSocio, with conventional, geographic, and socioeconomic conditions. For each configuration, we also investigated three values of selectivity: 1%, 25% e 50%. The lower the selectivity, the smaller the number of results. To obtain these values of selectivity, we set the radius to 35 km, 900 km, and 1,500 km, respectively, from New York. For the other factors, we controlled the selectivity by limiting the number of returned tuples.

Figure 4 shows the results obtained and the standard deviation. For the majority of the configurations, the normalized schema provided the best performance results, followed closely by the jointed schema, which in turn outperformed the split schema. Compared to the split schema, the normalized schema provided performance gains of up to 15.89%. This is due to the fact that the normalized schema first requires joins between the significantly smaller tables *Socioeconomic*, *City*, and *Hospital*, whose results are then used in the join with the huge fact table *Exam*. Compared to the split schema, the jointed schema provided performance gains of up to 13.68%. Because the jointed schema stores all attributes in the similarity dimension table *Hospital*, it requires only one join with *Exam* to process the query. On the other hand, the split schema requires more joins.

Investigating Examples of Semantic Queries. We used SimSparkOLAP to investigate examples of semantic queries and discuss how they are useful in the healthcare decision-making. To this end, we consider queries Q_1 , Q_2 , and Q_3

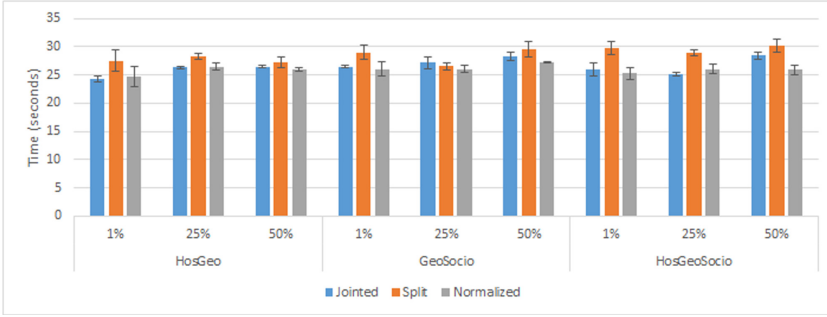


Fig. 4. Performance tests using SimSparkOLAP to investigate the effect of storing the geographic and socioeconomic similarity factors according to the proposed star schemas

described in Example 1. In the performance tests, we used the perceptual layer of *Color Histogram* for the image similarity factor. Based on the aforementioned findings, we only consider here the normalized schema. Figure 5 shows the results.

To provide a better visualization of how many exams from 1970 to 2010 have similar images to a given image of breast cancer, we show the results of query Q_1 by decade. In the decision-making, this type of query is important to analyze the evolution curve of a given disease over time, considering conventional and image data. Similar analyses can be performed using other aspects, such as the patients’ blood type. For instance, there are studies on COVID-19 that point out that the most susceptible blood type is ‘A’ and the more resistant is ‘O’ [22].

The results of query Q_2 show, for each city, how many exams have similar images to a given image of breast cancer from female patients that were captured in private hospitals located in cities within a 15 km radius of New York. In the healthcare decision-making, investigating geographic areas around a point of interest (e.g., city, tourist spot) may reveal the evolution of a given disease or identify an epicenter. It is also possible to explore the distribution of diseases that influence other diseases, such as *comorbidities* and its effects in COVID-19 [13].

For each age range and state of the patient, query Q_3 returns the quantity of exams that has similar images to a particular image of breast cancer, were captured in hospitals located in cities within a 1,500 km radius of New York, and refer to cities whose population has an age range similar to the age range of New York. Analyzing socioeconomic data is very important in the healthcare decision-making of several diseases. It is possible to identify points of interest with higher/lower number of cases and how a given disease affect people from different age ranges, salary ranges, and education levels. For instance, it is well-known that there is a higher COVID-19’s lethality for the elderly [11].

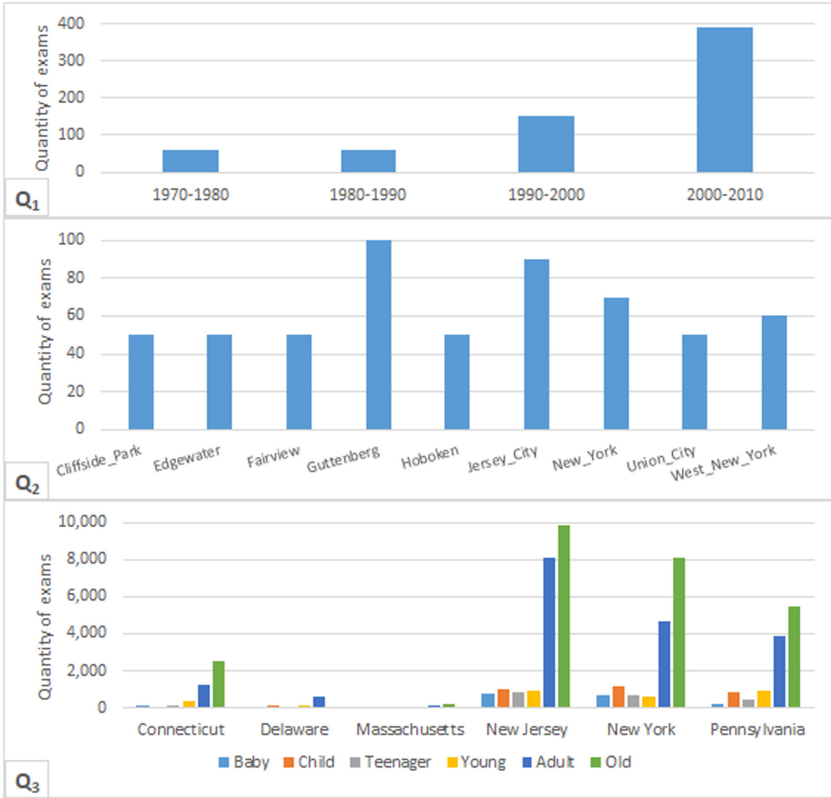


Fig. 5. Performance tests using SimSparkOLAP to investigate semantic queries

7 Conclusions and Future Work

In this paper, we focus on the management of geographic, socioeconomic, and image similarity factors in data warehouses to support healthcare decision-making. The contribution of our approach is threefold. First, regarding the storage of these factors, we propose three designs of star schema that investigate if geographic and socioeconomic data should be stored together or separately. Second, considering the processing of analytical queries extended with similarity search predicates in Spark, we introduce SimSparkOLAP and used it to investigate the effect of the different schemas on query performance. The results showed that storing the similarity search factors separately and linking one table to another respecting the granularity of the attributes provided better performance in the majority of the cases. Third, we used SimSparkOLAP to process examples of semantic queries and highlight that image, geographic, and socioeconomic data are of paramount importance to the healthcare decision-making. We are currently carrying out additional performance tests considering different

data volumes and healthcare datasets. Future work also include the analysis of other types of extended OLAP queries.



References

1. Brito, J.J., Mosqueiro, T., Ciferri, R.R., Ciferri, C.D.A.: Faster cloud star joins with reduced disk spill and network communication. In: Proceedings of the International Conference on Computational Science (2016). *Proc. Comput. Sci.* **80**, 74–85
2. Burdakov, A., et al.: Bloom filter cascade application to SQL query implementation on Spark. In: Proceedings of the 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, pp. 187–192 (2019)
3. Cuzzocrea, A.: Warehousing and protecting big data: state-of-the-art-analysis, methodologies, future challenges. In: Proceedings of the International Conference on Internet of Things and Cloud Computing. Article No.: 14, pp. 1–7 (2016)
4. Ferrahi, I., Bimonte, S., Boukhalfa, K.: Logical and physical design of spatial non-strict hierarchies in relational spatial data warehouse. *IJDWM* **15**(1), 1–18 (2019)
5. Gonzalez, R., Woods, R.: *Digital Image Processing*, 3rd edn. Prentice-Hall, Upper Saddle River (2006)
6. Haralick, R.: Statistical and structural approaches to texture. *Proc. IEEE* **67**(5), 786–804 (1979)
7. Jin, X., Han, J., Cao, L., Luo, J., Ding, B., Lin, C.X.: Visual cube and on-line analytical processing of images. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, pp. 849–858 (2010)
8. Kimball, R., Ross, M.: *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*, 2nd edn. Wiley, Hoboken (2002)
9. Kuo, M.H., Sahama, T., Kushniruk, A., Borycki, E., Grunwell, D.: Health big data analytics: current perspectives, challenges and potential solutions. *Int. J. Big Data Intell.* **1**, 114–126 (2014)
10. Li, D., Zhang, W., Shen, S., Zhang, Y.: SES-LSH: shuffle-efficient locality sensitive hashing for distributed similarity search. In: Proceedings of the IEEE International Conference on Web Services, pp. 822–827 (2017)
11. Mahase, E.: Covid-19: death rate is 0.66% and increases with age, study estimates. *BMJ* **369** (2020)
12. Nguyen, T.D.T., Huh, E.N.: An efficient similar image search framework for large-scale data on cloud. In: Proceedings of the ACM International Conference on Ubiquitous Information Management and Communication, pp. 65:1–65:8 (2017)
13. Richardson, S., et al.: Presenting characteristics, comorbidities, and outcomes among 5700 patients hospitalized with COVID-19 in the New York City area. *JAMA* **323**, 2052–2059 (2020)
14. Rocha, G.M., Ciferri, C.D.A.: ImgDW generator: a tool for generating data for medical image data warehouses. In: SBBB 2018 Proceedings Companion, pp. 23–28 (2018)
15. Rocha, G.M., Ciferri, C.D.A.: Processamento eficiente de consultas analíticas estendidas com predicado de similaridade em Spark. In: Proceedings of the 34th Brazilian Symposium on Databases, pp. 1–6 (2019, in Portuguese)
16. Sebaa, A., Chikh, F., Nouicer, A., Tari, A.: Medical big data warehouse: architecture and system design, a case study: improving healthcare resources distribution. *J. Med. Syst.* **42**, 59 (2018). <https://doi.org/10.1007/s10916-018-0894-9>

17. Teixeira, J.W., Annibal, L.P., Felipe, J.C., Ciferri, R.R., Ciferri, C.D.A.: A similarity-based data warehousing environment for medical images. *Comput. Biol. Med.* **66**, 190–208 (2015)
18. Traina, C., Filho, R.F.S., Traina, A.J.M., Vieira, M.R., Faloutsos, C.: The omnifamily of all-purpose access methods: a simple and effective way to make similarity search more efficient. *VLDB J.* **16**(4), 483–505 (2007)
19. Traina, C., Moriyama, A., Rocha, G.M., Cordeiro, R., Ciferri, C.D.A., Traina, A.J.M.: The SimilarQL framework: similarity queries in plain SQL. In: *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pp. 1–4 (2019)
20. Vaisman, A.A., Zimányi, E.: Spatial data warehouses. *Data Warehouse Systems. DCSA*, pp. 427–473. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54655-6_11
21. Zaharia, M., Chowdhury, M., Franklin, M.J., Shenker, S., Stoica, I.: Spark: cluster computing with working sets. In: *Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing*, pp. 10–10 (2010)
22. Zhao, J., et al.: Relationship between the ABO blood group and the COVID-19 susceptibility. *medRxiv* (2020)



OMProv: Provenance Mechanism for Objects in Deep Learning

Jian Lin^(✉)  and Dongming Xie 

Oriental Mind (Wuhan) Computing Technology Co., Ltd.,
Wuhan 430200, China
{linjian,xiedongming}@orientalmind.com

Abstract. Deep learning technology is widely used in industry and academia nowadays. Several kinds of objects are involved in deep learning workflows, including algorithms, models, and labeled datasets. The effectiveness of organizing and understanding the relationship among these objects determines the efficiency of development and production. This paper proposes OMProv, which is a provenance mechanism for recording the lineage within each kind of object, and the relationship among different kinds of objects in the same execution. A weighted directed acyclic graph-based version graph abstraction and a version inference algorithm are proposed. They are consciously designed to fit the characteristics of deep learning scenarios. OMProv has been implemented in OMAI, an all-in-one deep learning platform for the cloud. OMProv helps users organize objects effectively and intuitively, and understand the root causes of the changed job results like performance or accuracy in an efficient way. The management of deep learning lifecycles and related data assets can also be simplified by using OMProv.

1 Introduction

Deep learning technology has been widely used in the Internet, industry, education, and many other areas. It has become an infrastructure for artificial intelligence (AI)-enabled business innovation. The workflow of deep learning includes several phases, and each phase involves different objects. As shown in Fig. 1, typical phases are algorithm development, model training, training visualization, and data inference. Typical objects involved in these phases are algorithms, models, hyperparameters, original data, labels, labeled datasets, summary files, and visual graphs. In the execution of each phase, zero or more kinds of objects work as inputs, and some others play the roles of outputs. Users usually use different software tools (such as training engines, inference engines) for different phases, or leverage an all-in-one platform (such as cloud-based services) to manage the whole lifecycle and related objects of deep learning.

In research and engineering, users need to update these objects continually. For example, users may improve their algorithms, train models with different combinations of hyperparameters, or upgrade the models of online services.

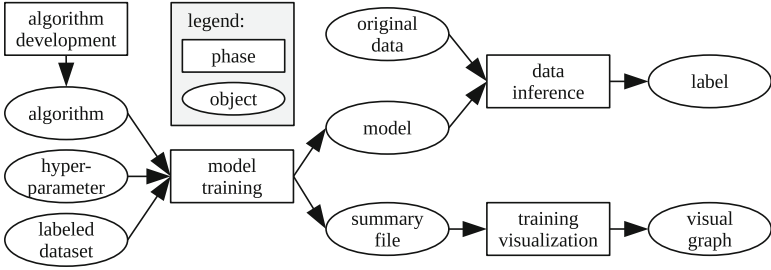


Fig. 1. The typical workflow of deep learning

The users care about the causality between updated inputs and changed outputs so that they can improve the functionality or performance of their workloads by responding to feedback. However, plenty of updating operations and intricate relationships among the operations and objects bring complexity to the management of deep learning lifecycles, which also annoy the users. In this scenario, a provenance mechanism is expected. This mechanism usually records the lineage within each kind of object, and the relationship among different kinds of objects in the same execution. In particular, it should maintain the version relationship (e.g. inheritance and derivation) of algorithms, models, etc., as well as the combination relationship of inputs and outputs within each execution of all deep learning phases.

Although many provenance mechanisms have been proposed for various data processing systems, they are not designed and optimized for the complicated scenario of deep learning. This paper aims at analyzing and solving the provenance issues in the deep learning scenario, especially for the case of cloud-based deep learning services. The contributions of this paper include:

- (1) A weighted directed acyclic graph (wDAG)-based version graph abstraction is proposed to fit the needs of deep learning workloads.
- (2) A version inference algorithm is proposed that infers the implicit lineage of outputs by referring to the explicit lineage of inputs.
- (3) A provenance mechanism that implements the abstraction and algorithm is designed. It is integrated with the OMAI [6] deep learning platform, and verifies the effectiveness of these designs.

2 Related Work

The provenance issue has been investigated in many data-related studies [10, 15]. Graph-based methods are popular in this area. The Open Provenance Model [9] is a systematic model for expressing and exchanging provenance information. Acar, et al. [1] propose a group of provenance graph semantics and corresponding query language. These two projects are based on directed acyclic graphs (DAG) and focusing on formal theories. In recent years, several version or provenance

management systems are designed deliberately for the machine learning or deep learning (ML/DL) scenario. DataLab [17] and MLdp [2] focus on the provenance of datasets. ModelDB [14] and ModelHub [8] focus on the provenance of models. ProvDB [7] works a unified provenance management system for the whole lifecycle of ML/DL. Most of the above systems utilize the information from users' inputs or actions to determine the version relationships of directly manipulated objects, but ignore those of indirectly generated objects. The latter are expected to be specified explicitly by users, rather than automatically by systems. In addition to these academic studies, some utilities for productive or experimental applications have also been developed. MEX [4] is a web service for the provenance management of machine learning experiments, which is based on semantic web technology. Runway [13] is a similar tool designed for online environments, which provides visualization and comparison features based on provenance information.

3 Problem Analysis

By analyzing the related studies and practices, deficiencies in existing version or provenance management mechanisms are identified. The main problems include:

- (1) Existing mechanisms usually record version relationships qualitatively rather than quantitatively. Traditional methods like [7] use edges of graphs to express inheritance or derivation relationships only. Numeric attributes of these relationships, such as the percentage of modification between versions, are not native parts of the graphs, but maintained by some additional data structures like commit logs. This makes it hard to do quantitative analysis or other advanced operations on the graphs, especially for generating informative version graphs for output objects.
- (2) Most mechanisms cannot automatically maintain version relationships for output objects. Traditional methods like [17] focus on the version management of input objects, because they are operated explicitly by users, and enough information can be obtained for parsing version relationships and generating corresponding data structures. The versions of output objects are also important, because they are derived data, which are less intuitive for the users but more useful for the provenance management. Existing methods like [8] often depend on user-specified information to generate version relationships for output objects, rather than maintain them automatically.
- (3) Provenance mechanisms are not well integrated with cloud-based deep learning services. In practice, the concepts and usage of most provenance mechanisms follow the convention of classic version control systems (VCS) like git. It is not friendly for deep learning cloud services. On the one hand, the semantics of VCS does not exactly match the concepts of deep learning. On the other hand, the phase-independent usage habits of VCS are different from those of cloud-based all-in-one platforms. Besides, the ability to visualize version information with deep learning semantics is also a shortcoming of existing systems.

In summary, existing version or provenance management mechanisms cannot fully meet the needs of deep learning, which is reflected in data structures, algorithms, and engineering practices.

4 Design of OMProv

In response to the above problems, and to better serve deep learning platforms, new abstractions and methods are expected. OMProv, a novel provenance mechanism is proposed. This section introduces its designs, involving data structures, algorithms, as well as the algorithm analysis and optimizations.

4.1 Version Graph Abstraction

In practice, users care not only about qualitative relationships such as inheritance or derivation, but also about quantitative relationships such as the amount of difference between two versions. This “amount” is not intended to replace the whole data structure maintaining the difference (e.g. commit log), but an intuitive value that can be used in some meaningful operations, such as quantitative analysis, version inference, and visualization. Using a single value is also a trade-off that balances simplicity and practicality.

To support this qualitative feature in version graphs, the weighted directed acyclic graph is adopted instead of the common DAG. Similar to using DAG, a node in a wDAG represents a version of a certain object, and a directed edge represents an inheritance relationship, where the source node inherits the destination node. In all the inheritance relationships, the source nodes are called the descendants, and the destination nodes are called the ancestors. The descendant and ancestor relationships are transitive in successive inheritances.

The weight of a directed edge represents the amount of difference between versions. It can be defined by using any reasonable variable or formula. The simplest way is to use 1 as the edge weight for all version graphs of input objects. It works like the case of using the common DAG, and it still benefits the version inference of output objects as been discussed in the next sub-section. A meaningful variable as the weight is the percentage of changed bytes between two versions. It can be used to measure and compare the versions of text-based objects like algorithms. A more complicated case is to use a value calculated by a formula as the weight. It is suitable for objects with multi-dimensional information like hyperparameters.

Figure 2(a) and 2(b) present the version graphs of two objects I^1 and I^2 as examples. Figure 2(a) shows the simple cases of linear inheritance and tree inheritance, while Fig. 2(b) shows the cases of multiple inheritance and variable weights. Each node is labeled with a name, which uses ni^p to denote “the version node of object I^p ”, and a subscript is marked to denote the version ID. Note that the version ID can be any format. For simplicity, integers are used in the examples, and we assume that a bigger integer represents a newer version. Each edge is labeled with its weight.

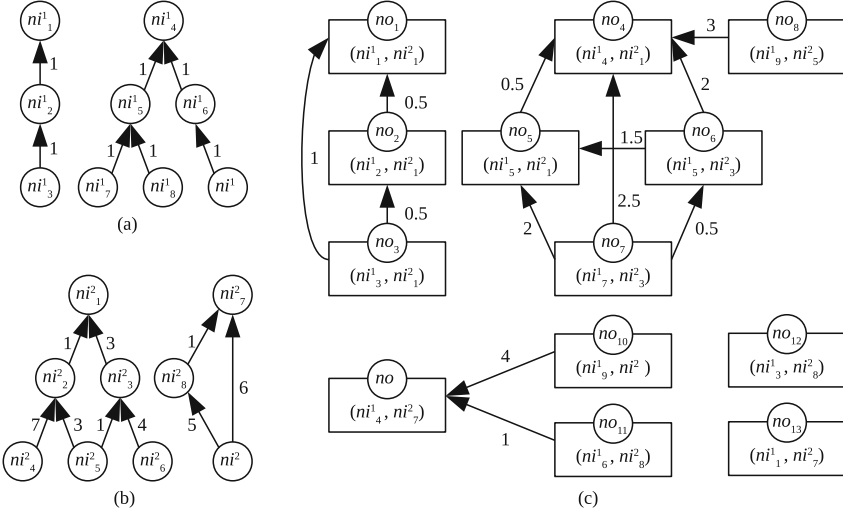


Fig. 2. The version graphs of input objects $\{I^1, I^2\}$ and output object O

4.2 Version Inference Algorithm

The version relationships of output objects are derived from those of input objects indirectly. It is not a natural choice to determine them by users' direct actions or explicit instructions, because it increases the burden of understanding and maintenance. The ideal situation is to specify the version relationships of output objects by the provenance mechanism automatically. To this end, a version inference algorithm is proposed. This algorithm works for each execution of a certain phase in deep learning workflows. It takes the version graphs of n input objects as inputs, and generates a version graph for one output object. The weights on the directed edges are calculated by the algorithm, so the qualitative relationships between the versions of the output object can be presented.

The symbols, concepts, and preconditions of the algorithm are defined as follows.

- (1) For the set S_I of n input objects $\{I^1, I^2, \dots, I^n\}$, the version graph of I^p is GI^p , which has the node set $\{ni^p_1, ni^p_2, \dots, ni^p_w\}$ and the edge set $\{ei^p_{from_1, to_1}, ei^p_{from_2, to_2}, \dots, ei^p_{from_x, to_x}\}$. For the output object O , the corresponding version graph is GO , which has the node set $\{no_1, no_2, \dots, no_y\}$ and the edge set $\{eo_{from_1, to_1}, eo_{from_2, to_2}, \dots, eo_{from_z, to_z}\}$. A graph is empty before the first version of the corresponding object is created. The node set of a graph is noted as $nodes(graph)$, while the edge set is noted as $edges(graph)$.
- (2) An additional data structure called input version tuple is attached to each node no_j in GO for recording which versions of input objects create the corresponding version of O . These input objects are called the **provenance objects** of O , and the corresponding versions are called the **provenance**

versions of no_j . The function of getting input version tuple is defined as $ivtuple(node)$. For example, Node no_j with tuple $ivtuple(no_j) = (ni_a^1, ni_b^2)$ means the version j of O is created with the version a of I^1 and the version b of I^2 . The operator $[p]$ is introduced for getting the element with the index p from the tuple, where p is the same as that in \mathbf{S}_I .

- (3) If a set of consecutive edges with the same direction existing between nodes ni_a^p and ni_b^p in GI^p , where the direction starts from ni_a^p and end to ni_b^p , then we define that it is reachable through an inheritance route $ri_{a,b}^p$ from ni_a^p to ni_b^p . Reachable is a reflexive relation, so a node is reachable to itself through a route with an empty set of edges. The boolean function of reachable is defined as $reach(graph, from, to)$. We also define $routes(GI^p, ni_a^p, ni_b^p)$ as a function of getting the set of all routes from ni_a^p to ni_b^p . These definitions also apply to the version graphs of output objects.
- (4) If ni_a^p and ni_b^p in GI^p are the provenance versions of no_i and no_j in GO respectively, then $routes(GI^p, ni_a^p, ni_b^p)$ are called the **provenance routes** from no_i and no_j . Logically, a provenance route includes all the nodes and edges in it. All the properties of the nodes and edges are required when executing the algorithm. The maintenance of provenance routes can be optimized in implementation, such as keeping the pointers to nodes in GI^p only.
- (5) The weight of a directed edge $ei_{a,b}^p$ between two nodes ni_a^p and ni_b^p in GI^p is noted as $W(ei_{a,b}^p)$. Similarly, the weight of a route $ri_{a,b}^p$ is defined as $W(ri_{a,b}^p)$, which equals to the sum of weights of all edges in the route. These symbols also apply to the version graphs of output objects.
- (6) The weight of a directed edge $eo_{i,j}$ between two nodes no_i and no_j in GO is determined by a function that takes the properties of a set $\mathbf{S}_{RI_{i,j}}$ of route sets as inputs. Each element in $\mathbf{S}_{RI_{i,j}}$ is a provenance route set $RI_{i,j}^p$ related to the version graph GI^p . Each route set consists of the provenance routes between the provenance version nodes in GI^p with which no_i and no_j are created. The weight function is noted as $weight(set)$.

A basic version of the algorithm is introduced at first. Algorithm 1 lists the steps of updating the version graph GO when a new version node no_i is created with $ivtuple(no_i)$. Later we will discuss some advanced and variant situations.

This algorithm comes from the core idea: for an output object, we consider that a new version i inherits an old version j , if and only if each provenance version that creates version i inherits the corresponding provenance version that creates version j respectively. Thus, the procedure of the algorithm is to scan the version graphs of all input objects (Line 5), and check whether an inheritance relationship (i.e. provenance version) exists for each provenance object (Line 6). If all provenance objects satisfy this (Line 13), then the inheritance relationship between the new and old versions of the output object exists. The algorithm also implies that one new version may inherit many old versions (Line 2), which is the so-called multiple inheritance.

The weight function in this algorithm is also worth analyzing. For the traditional case of using common DAG, the weight function is a constant function. For the case of wDAG, many kinds of weight functions can be proposed. Typical functions take the edge weights in the provenance routes as parameters.

Algorithm 1: Version inference algorithm

```

1 add a node  $no_i$  to  $nodes(GO)$ 
2 for each  $no_j$  in  $nodes(GO) \setminus \{no_i\}$  do
3    $flag_{reach} \leftarrow true$ 
4    $\mathbf{S}_{RI_{i,j}} \leftarrow \emptyset$ 
5   for each  $ni_k^p$  in  $ivtuple(no_i)$  do
6     if  $reach(GI^p, ni_k^p, ivtuple(no_j)[p]) = false$  then
7        $flag_{reach} \leftarrow false$ 
8       break
9     else
10      add a set  $routes(GI^p, ni_k^p, ivtuple(no_j)[p])$  to  $\mathbf{S}_{RI_{i,j}}$ 
11    end
12  end
13  if  $flag_{reach} = true$  then
14    add a directed edge  $eo_{i,j}$  (from  $no_i$  to  $no_j$ ) to  $edges(GO)$ 
15     $W(eo_{i,j}) \leftarrow weight(\mathbf{S}_{RI_{i,j}})$ 
16  end
17 end

```

A meaningful instance is introduced as follows. When adding a new edge $eo_{i,j}$ to the version graph GO , the lightest route (with the minimum weight) in each route set $RI_{i,j}^p$ is noted as $ril_{i,j}^p$. On this basis, the weight function is defined as $weight(\mathbf{S}_{RI_{i,j}}) = \frac{1}{n} \sum_{p=1}^n W(ril_{i,j}^p)$. The intuitive meaning of this function is to accumulate and average the amounts of difference between the source and destination versions in all provenance objects. The idea of the lightest route is to simplify the handling of multiple inheritance, which aims at avoiding the expansion of the number of edges. The introduction of the factor $\frac{1}{n}$ is for normalizing the range of weights, which aims at preventing the expansion of weights over cascaded phases.

As an example, Fig. 2(c) presents the inferred version graph of output object O , which is created with I^1 and I^2 in Fig. 2(a) and 2(b). The input version tuple is annotated below each node. All the nodes and edges are appended in the integer order of version ID by using Algorithm 1.

4.3 Algorithm Analysis

The time complexity of the version inference algorithm determines the performance of the provenance mechanism. We analyze it using the symbols defined above: $n = |\mathbf{S}_I|$, and $y = |nodes(GO)|$. For a certain phase in deep learning workflows, n is constant, while y grows with the version nodes of the output object. According to the steps of Algorithm 1, the total execution time of the core logic is $t_{total} = y \cdot (n \cdot t_{check} + t_{update})$, where t_{check} is the time for checking the reachability between nodes in GI^p , and t_{update} is the time for updating the data structures of GO .

To minimize t_{total} , obviously t_{check} and t_{update} should be reduced. For GI^p , a graph with efficient indexes is necessary. The classic transitive closure algorithm

[11] can query the reachability in $O(1)$ time at the expense of the index construction cost. Many new reachability algorithms [16] are available that balance the querying cost and the index construction cost, which are suitable for graphs with large scales. For GO , a simple graph data structure without extra indexes is good enough, which takes $O(1)$ time for updating a single node or edge. If a graph is used for both input and output, an indexing mechanism supporting efficient incremental updating [5] is preferred.

Note that the time cost of Algorithm 1 can be further reduced by optimizing the whole logic if proper indexing mechanisms exist. For example, it is possible to reduce the iterations of the outer for loop by referring to the transitive closures of GO . Since this kind of optimization is implementation-specified, no detailed analysis will be performed here.

4.4 Optimizations

Algorithm 1 is easy to understand and implement, but some situations in practice require special considerations. This sub-section discusses these situations and provides directions for improving the algorithm.

Redundant Edge Avoidance. Consider an inheritance route in a version graph of some output object. When a new version node is added and inheriting the most descending node in the route, a set of directed edges will be added from the new node to all nodes in the route according to the above algorithm. For example, when no_3 is added to GO in Fig. 2(c), the algorithm adds not only $eo_{3,2}$ but also $eo_{3,1}$. This is correct in semantics, but redundant in practice, especially when visualizing the version graph. A better choice is to keep the edge to the most descending node only, and omit those to its ancestors.

This design is named redundant edge avoidance. Two preconditions are proposed for this design. Firstly, a property called the descendant number is attached to each version node, which is noted as $n_desc(node)$. It records the total number of all direct and indirect descendants of a node. Secondly, an updating procedure of $n_desc(node)$ is employed. This procedure is used for updating the descendant numbers of all ancestors when a new version node is added. It can be implemented with any DAG traversal algorithm like the depth-first search. Note that the descendant number of an ancestor should be increased only once when multiple routes caused by the new node are linked to the ancestor. The descendant number of a new node is set to 0.

On this basis, Algorithm 1 can be improved as follows so that the redundant edges will not be added when a new node is added.

- (1) Before iterating through $nodes(GO) \setminus \{no_i\}$ in Line 2, sort the nodes in $nodes(GO)$ by $n_desc(node)$ in ascending order. Thus, the version nodes with smaller descendant numbers will be processed earlier in the outer for loop.
- (2) Before adding a directed edge in Line 14, check $reach(GO, no_i, no_j)$. If it is *true*, then skip the step of adding this edge. This means that a direct route will be omitted if an indirect route has already existed.

In this way, a new version node will be linked only to the most direct ancestor that is the one with the smallest descendant number in its inheritance route. Note that we should not break the outer for loop after adding this edge, because there may be other potential edges if multiple inheritance relationships exist in different inheritance routes.

Go back to the example in Fig. 2(c). By applying this design, $eo_{3,1}$, $eo_{6,4}$, $eo_{7,4}$ and $eo_{7,5}$ will not be added. The version graph becomes more clear, as it reflects the version evolution of the output object caused by the changes of input objects step by step.

Reverse Version Inference. An implicit assumption of Algorithm 1 is that creating newer versions of output objects tends to use the newer versions of input objects. This is in line with usual work practices. However, it is possible to create a new version of an output object with old versions of input objects. Even if the new version is inherited by a certain old version due to the rules of provenance versions, the algorithm will not add a directed edge from the old node to the new one. For example, when no_{11} is added to GO in Fig. 2(c), although its provenance versions are both the ancestors of the provenance versions of no_{10} , the inheritance relationship from no_{10} to no_{11} will not be found.

This behavior makes sense in most cases, because the inferred version graphs of output objects reflect the time sequence of users' explicit actions. A new version being inherited by an old one may be confusing. However, allowing this kind of reversely inferred inheritance is reasonable in some cases. For example, in a multi-user collaboration environment, one user working with old versions of input objects should be notified that some others have already had outputs created with new inputs. Another scenario is to do a retrospective analysis over a set of historical deep learning experiments. The version relationship is more worthy of attention compared to the time sequence in this scenario.

To support reverse version inference, the concepts of reverse routes and reverse reachable are introduced. They are similar to the concepts of inheritance routes and reachable defined above, but in a reverse direction. If $reach(GI^p, ni_a^p, ni_b^p) = true$, then we define that it is reverse reachable through a reverse route $ri_{b,a}^p$ from ni_b^p to ni_a^p , and its boolean function is $revreach(graph, from, to)$. The corresponding function of getting reverse routes ($revroutes(graph, from, to)$) is insignificant, because it returns the same set as $routes(graph, from, to)$ does. The only difference is that the algorithm needs to scan the route from the destination to the source.

Based on these definitions, the algorithm can be improved in a simple way. We can repeat the logic of Line 3–16 within the outer for loop, but use a different $flag_{revreach}$ and the function $revreach(graph, from, to)$ to identify the reverse reachable relationship. If an inheritance relationship between two version nodes is found by reverse version inference, then a weighted directed edge $eo_{j,i}$ will be added to the version graph. This logic can be optimized in implementation, such as merging the inner for loop with the original one.

In the example of Fig. 2(c), this design will append $eo_{10,11}$ ($W(eo_{10,11}) = 3$) and $eo_{12,13}$ ($W(eo_{12,13}) = 1.5$) to the version graph. The facts that no_{11} and no_{13} are created with relatively old versions of input objects are detected.

5 Implementation

OMProv has been implemented in the OMAI [6] deep learning platform. This section introduces the key points of implementation, and illustrates the practical value of OMProv.

5.1 OMAI Introduction

OMAI is an all-in-one and high-performance deep learning platform designed for the cloud. It covers the whole lifecycle of deep learning including algorithm development, model training, training visualization, and data inference. The objects involved in these phases are abstracted as data assets. They are saved in an object storage system, and managed by the platform uniformly. OMProv works in the layer of data asset management as the provenance mechanism for all kinds of objects. OMAI provides a group of web consoles and REST APIs for both end-users and service operators. It is designed and implemented with container technology, so the container-based resource management components such as the job scheduler and the cluster manager are indispensable.

5.2 OMProv in OMAI

OMProv provides the provenance capabilities for data assets managed by OMAI. It runs as a cloud-native micro-service, and is loosely coupled with other components. OMProv consists of four logical parts being responsible for different execution processes: version tracking, version inference, graph storage, and graph visualization. The version graph abstraction and version inference algorithm are implemented in these logical parts. The workflow of OMProv with related data flows is shown in Fig. 3.

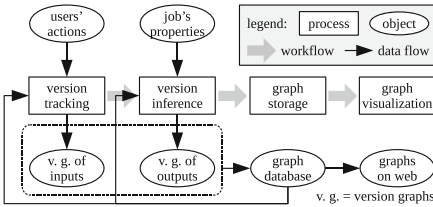


Fig. 3. The workflow of OMProv

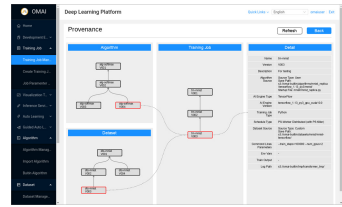


Fig. 4. The UI of OMProv in OMAI

The version tracking process is triggered by the users' actions of creating or modifying data assets. To track the version relationships of input objects, both explicit and implicit methods are provided. For a newly created object, OMProv will create an independent version node by default, but the users can still specify inheritance routes explicitly to record their opinions on the version. For a

modified object managed by the data asset management components, OMProv will create a version node inheriting the original version implicitly. The determination of edge weights has several options including custom, constant, and the modification percentage. The calculation of modification percentage also has different options. Two basic implementations are “wdiff -s”-based [12] statistics for text files, and “diff -r”-based comparison for any binary datasets.

The version inference process runs automatically when a job performing a deep learning phase is executed. The combination relationship between input and output data assets is extracted from the job’s properties. To infer the version graphs of output objects, OMProv leverages the algorithm introduced in Sect. 4 with the redundant edge avoidance mechanism. The weight function is customizable. The main application scenario of this process is to track the versions of models during iterative training jobs, as well as the provenance versions of corresponding algorithms, hyperparameters, and datasets.

The graph storage process uses ArangoDB [3], a multi-model database. The version graphs are stored as “named graphs” to benefit from the transaction and associated operation features. The graphs will be used as the inputs of other execution processes. The graph visualization process uses graphics libraries based on HTML 5. It renders the version graphs on the web consoles according to the users’ queries. Figure 4 shows the user interface of the provenance feature based on OMProv in OMAI.

In summary, the four processes are integrated naturally with the existing workflow of deep learning. The concepts of version control are hidden, and few additional operations are required. By introducing OMProv in OMAI, the users can manage the versions of input data assets like algorithms in an effective way, and understand the version evolution of output data assets like models in an intuitive way. By investigating the provenance versions, the users can identify the root causes of the changed job results like performance or accuracy efficiently.

6 Conclusion

Provenance mechanisms are important for various data processing systems, especially for deep learning platforms which involve iterative training and inference generating complicated version relationships among different kinds of objects. Traditional mechanisms have disadvantages on non-quantified relationship records, missing supports for the versions of output objects, and weak integration with cloud-based services. OMProv, a provenance mechanism for objects in deep learning is proposed. It provides a wDAG-based version graph abstraction and a version inference algorithm that can record and infer the lineages of both inputs and outputs in an effective and intuitive way. OMProv has been implemented in the OMAI deep learning platform. It tracks and stores the provenance information for various data assets like algorithms and models. By using OMProv, the users can understand the version evolution easily and identify the causes of results efficiently. In the future, we will do more application case studies of OMProv on analyzing algorithm issues, improving model performance, and achieving model reproducibility.




Acknowledgments. We would like to thank the OMAI development team for the contributions to the high-quality implementation of this software.

References

1. Acar, U., Buneman, P., Cheney, J., Van Den Bussche, J., Kwasnikowska, N., Vansummeren, S.: A graph model of data and workflow provenance. In: Proceedings of the 2nd Workshop on Theory and Practice of Provenance, pp. 1–10 (2010)
2. Agrawal, P., et al.: Data platform for machine learning. In: Proceedings of the 2019 International Conference on Management of Data, pp. 1803–1816 (2019)
3. ArangoDB Inc: ArangoDB (2011). <https://www.arangodb.com>
4. Duarte, J.C., Cavalcanti, M.C.R., de Souza Costa, I., Esteves, D.: An interoperable service for the provenance of machine learning experiments. In: Proceedings of the 2017 International Conference on Web Intelligence, pp. 132–138 (2017)
5. Jin, R., Ruan, N., Xiang, Y., Wang, H.: Path-tree: an efficient reachability indexing scheme for large directed graphs. *ACM Trans. Database Syst.* **36**(1), 1–44 (2011)
6. Lin, J., Xie, D., Yu, B.: Research on Cloud Service Adaptation of Deep Learning. *Softw. Guide* **19**(6), 1–8 (2020). (in Chinese)
7. Miao, H., Chavan, A., Deshpande, A.: ProvDB: lifecycle management of collaborative analysis workflows. In: Proceedings of the 2nd Workshop on Human-in-the-Loop Data Analytics, pp. 1–6 (2017)
8. Miao, H., Li, A., Davis, L.S., Deshpande, A.: Towards unified data and lifecycle management for deep learning. In: Proceedings of the 33rd International Conference on Data Engineering, pp. 571–582 (2017)
9. Moreau, L., et al.: The open provenance model core specification (v1.1). *Future Gener. Comput. Syst.* **27**(6), 743–756 (2011)
10. Simmhan, Y.L., Plale, B., Gannon, D.: A survey of data provenance techniques. Technical report IUB-CS-TR618, Computer Science Department, Indiana University (2005)
11. Simon, K.: An improved algorithm for transitive closure on acyclic digraphs. *Theor. Comput. Sci.* **58**(1–3), 325–346 (1988)
12. The Free Software Foundation: GNU Wdiff (2014). <https://www.gnu.org/software/wdiff/>
13. Tsay, J., Mummert, T., Bobroff, N., Braz, A., Westerink, P., Hirzel, M.: Runway: machine learning model experiment management tool. In: Proceedings of SysML Conference 2018, pp. 1–3 (2018)
14. Vartak, M., et al.: ModelDB: a system for machine learning model management. In: Proceedings of the 1st Workshop on Human-in-the-Loop Data Analytics, pp. 1–3 (2016)
15. Wang, J., Crawl, D., Purawat, S., Nguyen, M., Altintas, I.: Big data provenance: challenges, state of the art and opportunities. In: Proceedings of the 2015 IEEE International Conference on Big Data, pp. 2509–2516 (2015)
16. Yu, J.X., Cheng, J.: Graph reachability queries: a survey. In: Aggarwal, C., Wang, H. (eds.) *Managing and Mining Graph Data*. ADBS, vol. 40, pp. 181–215. Springer, Boston (2010). https://doi.org/10.1007/978-1-4419-6045-0_6
17. Zhang, Y., Xu, F., Frise, E., Wu, S., Yu, B., Xu, W.: DataLab: a version data management and analytics system. In: Proceedings of the 2nd International Workshop on Big Data Software Engineering, pp. 12–18 (2016)



Exploiting IoT Data Crossings for Gradual Pattern Mining Through Parallel Processing

Dickson Odhiambo Owuor^{1,2}(✉) , Anne Laurent¹ ,
and Joseph Onderi Orero² 

¹ LIRMM Univ. Montpellier, CNRS, Montpellier, France
{doowuor, laurent}@lirmm.fr

² Faculty of IT, Strathmore University, Nairobi, Kenya
jorero@strathmore.edu

Abstract. Today, with the proliferation of *Internet of Things* (IoT) applications in almost every area of our society comes the trouble of deducing relevant information from real-time time-series data (from different sources) for decision making. In this paper, we propose a fuzzy temporal approach for crossing such data sets with the ultimate goal of exploiting them for temporal gradual pattern mining. A temporal gradual pattern may take the form: “*the higher the humidity, the lower the temperature, almost 15 min later*”. In addition, we apply parallel processing on our implementation and measure its computational performance.

Keywords: Data streams · Fuzzy logic · Gradual patterns · Time-series

1 Introduction

Time-series data can be defined as a sequence of data points that are temporally-oriented. Sources of time-series data are numerous; for instance it may be obtained from internal sources (e.g. a data warehouse collecting *IoT* sensor data) or from an external source (e.g. data distributed by government institutions such as weather stations) etc. Time-series data can also be defined as a *data stream* when it becomes “*a potentially infinite sequence of precise data points recorded over a period of time*” [22].

Currently, there exists a great number of statistical tools and intelligent applications that researchers use to deal with huge volumes of time-series data accumulating in real-time in order to summarize it for extraction of patterns to aid in decision making (also known as *trend analysis*) [11, 21, 22, 24]. Due to the increased popularity of *trend analysis*, frameworks and standards such as (Open Geospatial Consortium) *OGC SensorThings* have emerged to enable easy management and sharing of time-series data among different research institutions

[13, 17]. These standards aim to integrate sensor data into Spatial Data Infrastructures (SDI) so that they become additional sources of geospatial information besides traditional data types like maps.

SDIs are used to implement frameworks that facilitate management and reuse of geospatial data. As a result, time-series data offer great potential for integrating real-time observations with geospatial data for visualization [12, 15, 23]. The *SensorThings* API can be applied to any (Internet of Things) IoT use case that generates time-series data. For example, IoT for smart cities [10], or IoT-based web service protocol [14], among others. However, these use cases rely on the analysis of singleton data sets to allow for automated decision making [7, 8].

In the research community, great interest has been expressed regarding *crossing* data from different sources in order to discover new knowledge about phenomena that could not otherwise be discovered by analyzing the individual data sets in isolation. *Data crossing* enables the matching of different data sets using a pre-defined criteria and combining their data points to form a new data set [3, 11, 12].

In this paper, we propose a fuzzy model that crosses time-series data sets from different sources as illustrated in Fig. 1. By using a fuzzy model, our proposed approach becomes more robust than other crisp models that could miss a phenomenon because of small data variations. We specifically test our model on numeric time-series data sets so as to extract temporal gradual patterns afterwards. Temporal gradual pattern mining allow for extraction of relevant correlations among the attributes of a data set together with a time lag [4, 16, 21]. For instance given the data set in Fig. 1, a temporal gradual pattern may take the form: “*the higher the humidity, the higher the number of flies, almost 2 min later*”.

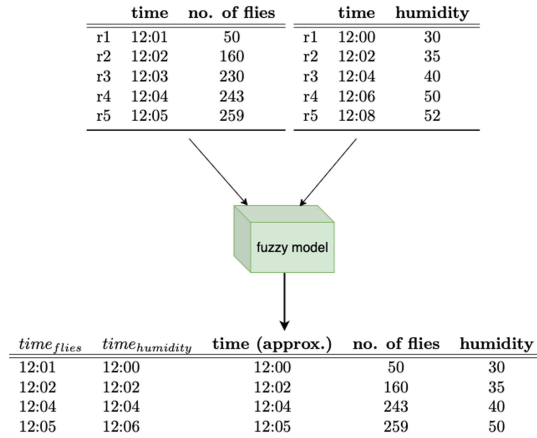


Fig. 1. An illustration of crossing 2 time-series sets

The remainder of this paper is organized as follows: we review related literature in Sect. 2; we describe our proposed fuzzy-temporal approach for crossing time-series data in Sect. 3; we analyze the performance of our proposed approach in Sect. 4; finally, we conclude and give future directions regarding this work in Sect. 5.

2 Related Works

Apart from analyzing singleton time-series data sets, it may be interesting to cross numerous such data sets in order to discover any hidden correlations. There are several previous works related to retrieving different sensor data in a *wireless sensor network (WSN)* via *query processing* [9, 25, 26].

Query processing allows users to retrieve information about sensed data values by issuing pre-defined queries to the storage engine. In fact, the nature of storage engine used by a WSN determines which type of *query processing* is applicable [9, 26]. According to [9], there are 3 groups of storage models: *local storage* model where sensors keep their on data in a distributed manner, *external storage* model, where all sensors store data in a common database in a centralized manner, and *data-centric* storage model which combines both previous models.

It is important to highlight that *query processing* relies on declarative languages (e.g. SQL - Structured Query Language) in all these models. Therefore, most research work relate to increasing the efficiency of *query processing* in either a distributed or a centralized *WSN* model [2, 9, 25]. Another key point to mention is the emergence of the term *fuzzy join*, which enables declarative languages such as SQL to generate views based on textual similarity. [18, 19] propose a fuzzy-search model which extends U-SQL to allow *fuzzy joins* on numeric data.

They achieve this by representing attribute values as fuzzy members and assigning them to a linguistic variable. However, data management frameworks especially the *OGC SensorThings* are built on top of NoSQL models. Therefore, SQL -based or U-SQL-based querying models are difficult to integrate into such frameworks.

An alternative model is proposed by [3] which extracts, transforms and loads into a data warehouse, different time-series data collected by a *WSN*. In this work, they demonstrate how to transform the data by normalizing the data types of its attributes including *date-time* before loading it into a data warehouse. The drawback with the normalization technique is the problem of *perfect matching* since it merges *tuples* based on the large *date-time* period and discards values with small granularity. Under those circumstances crossing is only possible if the largest *date-time* values match perfectly.

In this paper, we propose a model that will extract time-series data from unrelated sources, transform them using a fuzzy membership function so that crossing is possible through estimation even when the *date-time* values do not match. Additionally, we integrate this work into an *OGC SensorThings* API implementation that deals with environmental sensor data.

3 Fuzzy Temporal Crossing (*FuzzTX*) Approach

In this section, we construct a fuzzy model for crossing time-series data sets from different sources. We cross them with the intention of extracting temporal gradual patterns. For example, let us assume we have 2 sample time-series data sets as shown in Table 1.

Table 1. (a) Sample of population of flies time-series data (b) sample of humidity time-series data

	Time	No. of flies
r1	12:01	50
r2	12:02	160
r3	12:03	230
r4	12:04	243
r5	12:05	259

(a)

	Time	Humidity
r1	12:00	30
r2	12:02	35
r3	12:04	40
r4	12:06	50
r5	12:08	52

(b)

The *date-time* attribute reveals how closely simultaneous the occurrence of data points of the 2 data sets are. For example, time-series data sets having most of their data points occurring at almost the same time, when crossed, yield a data set that combines almost all the data points of the individual sets.

In this example, we notice that there exists a degree of fuzziness for any time interval that matches respective tuples in both data sets. For instance if we use a triangular membership function and we pick ‘1200 *h*’ as the center of this function - the membership degrees (MDs) of humidity data set’s ‘*time*’ attribute may be approximated as: $\{(1200, 1.0), (1202, 0.8), (1204, 0.6), (1208, 0.4), (1208, 0.2)\}$.

Similarly, the MDs of the number of flies data set’s ‘*time*’ attribute may be approximated as: $\{(1201, 0.9), (1202, 0.8), (1203, 0.7), (1204, 0.6), (1205, 0.5)\}$. Therefore, for any center that we pick between ‘1200 *h*’ and ‘1208 *h*’, the MD in the population of ‘*time*’ attribute decreases from closest value to the furthest value. This interesting (MD) feature can be harnessed and applied on a fuzzy model that may cross time-series data from different sources. We describe this model in the section that follows.

3.1 Building the *FuzzTX* Model

In *Fuzzy sets* theory, there exists a great number of membership functions that one can apply on a data set for fitting purposes [1, 20, 27]. In this paper, we pick a triangular membership function (MF) so that we can order the MDs of *date-time* population with reference to a single-value center. Automatically, this eliminates any MF whose center includes more than one value.

It is important to mention that we pick a triangular MF over the Gaussian MF since it is simpler to implement and, moreover our interest is not in fitting the

data set perfectly [20]. For instance, it is easy to construct an initial triangular MF for the *date-time* population of each time-series data by using the minimum value as the center and the smallest difference in the population to calculate the minimum and maximum extremes as shown in Fig. 2 (a) and (b).

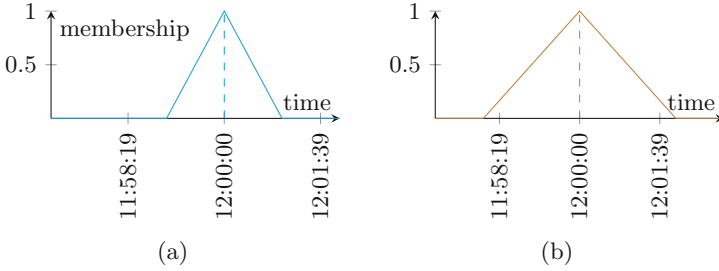


Fig. 2. (a) Membership function for pop. of flies data set (b) membership function for humidity data set

Using these 2 triangular MFs, we can build a model that crosses tuples based on MDs. We exploit this idea and propose the following *pseudo-code*:

1. pick the triangular MF with the largest boundaries
2. apply the MF on each data set’s *date-time* population to pick the *tuple-index* of the value with the largest (MD)
3. use the *tuple-index* to retrieve and cross tuples
4. slide the MF positively by its boundary, and repeat step 1 until the population is exhausted.

As an illustration, we apply the pseudo-code steps on the data sets in Table 1 (a) and (b) to obtain the data set in Table 2 (b).

Table 2. (a) Tuple indices of humidity and pop. of flies data sets after applying steps 1 and 2 (b) crossed data set after applying steps 3 and 4

Center	Humidity index (max. MD)	Flies index (max. MD)
12:00	r1 (1.0)	r1 (0.9)
12:02	r2 (1.0)	r2 (1.0)
12:04	r3 (1.0)	r4 (1.0)
12:06	r4 (1.0)	r5 (0.9)

(a)

Time	Humidity	No. of flies
12:00	30	50
12:02	35	160
12:04	40	243
12:06	50	259

(b)

Having crossed the 2 data sets, it is possible to apply gradual pattern mining techniques especially T-GRAANK (Temporal GRAdual rANKing) [21] to

extract the gradual correlation between humidity and population of flies. For instance a relevant pattern may be: $\{(humidity, \uparrow), (flies, \uparrow)_{\approx +2 mins}\}$ which may be interpreted as “the higher the humidity, the higher number of flies, almost 2 min later”.

3.2 FuzzTX Algorithm

In this section, we present Algorithm 1 which implements the FuzzTX model described in Sect. 3.1. In this algorithm, we first extract the *date-time* values from each time-series data set and store them in individual arrays (*line 4*). The smallest difference between elements in each *date-time* array is added to the boundary array \mathcal{B} (*line 8*). This array is used to determine the boundaries of the MF (*line 10*). Next, we build a triangular MF that initially starts from the smallest *date-time* value and it is positively slid by a specific boundary until it is equal to the largest *date-time* value (*line 13*).

Algorithm 1: *FuzzTX* algorithm

```

Input : time-series data sets  $DS^*$ 
Output: data set  $\mathcal{D}$ 
1  $\mathcal{B} \leftarrow \emptyset, \mathcal{D} \leftarrow \emptyset;$ 
2  $T_{min}, T_{max};$ 
3 for  $ds$  in  $DS^*$  do
4    $T_{arr} \leftarrow \text{ExtractTime}(ds);$ 
5    $T_{max} \leftarrow \max(T_{arr});$  /* iff greater */
6    $T_{min} \leftarrow \min(T_{arr});$  /* iff lesser */
7    $min_d \leftarrow \text{MinDiff}(T_{arr});$ 
8    $\mathcal{B} \leftarrow \mathcal{B} \cup min_d;$ 
9 end for
10  $bound_{sel} \leftarrow \max(\mathcal{B});$  /* largest boundary */
11  $t \leftarrow T_{min};$ 
12 while  $t \leq T_{max}$  do
13    $mf \leftarrow \text{BuildTriMF}(t - bound_{sel}, t, t + bound_{sel});$ 
14   for  $T_{arr}$  of each  $ds$  in  $DS^*$  do
15      $index \leftarrow \max(\text{FuzzyX}(mf, T_{arr}));$  /* index with largest membership degree */
16     if  $index$  then
17        $x_{tuple} \leftarrow x_{tuple} \cup ds_{tuple}[index];$ 
18        $\text{Delete}(ds_{tuple}[index]);$ 
19     else
20        $x_{tuple} \leftarrow \text{False};$ 
21        $\text{Break}();$ 
22   end for
23   if  $x_{tuple}$  then
24      $\mathcal{D}_{tuples} \leftarrow \mathcal{D}_{tuples} \cup x_{tuple};$ 
25   end if
26    $t \leftarrow t + bound_{sel};$ 
27 end while
28 return  $\mathcal{D}$ 

```

Finally, the *for-loop* implements the *pseudo-code*, described in Sect. 3.1, to determine the tuple indices of each data set that has the largest MD. The indices are used to cross data sets (*line 24*).

4 Experiments

In this section, we analyze the efficiency of the *FuzzTX* algorithm and discuss its performance results. It is important to mention that we implemented the algorithm using *Python* language in order to benefit from the language’s dynamism especially when dealing with large data sets.

4.1 Parallel Multi-processing

We analyze the multi-processing behavior of the *FuzzTX* algorithm using the *speedup* and *parallel efficiency* performance measures. *Speedup* $S(n)$ is the ratio of the execution time of a single processor to the execution time of n processors: $S(n) = T_1/T_n$. *Parallel efficiency* $E(n)$ is the average utilization of n processors: $E(n) = S(n)/n$ [5].

In the *FuzzTX* algorithm, we implement parallel processing at 2 code segments: (1) the *for-loop* between *lines* 3–8 and (2) the *while-loop* between *lines* 12–26 since each of their steps can be executed in isolation. Figure 6 (a) and (b) shows the speedup and parallel efficiency of the algorithm.

4.2 Experimental Setup

To test computational efficiency, the *FuzzTX* algorithm was executed on 7 time-series data sets obtained from OREME’s¹ data portal that recorded meteorological observations at the Puéchabon weather station between the years 2000 and 2017 [6]. Each data set has 4 attributes and 216 tuples. We performed test runs on a 2.9 GHz Intel Core *i7* MacBook Pro 2012 model, with 8 GB DDR3 RAM.

To test parallel processing efficiency, the *FuzzTX* algorithm was executed on 3 time-series data sets obtained from OREMES’s data portal that recorded swell sensor signals of 4 buoys near the coast of the Languedoc-Roussillon region between the years 2012 and 2019. Each data set has 30 attributes and tuples ranged from 15,000, 100,000 and 200,000 tuples. The test runs were performed on a (High Performance Computing) HPC platform **Meso@LR**. We used one node made up of 28 cores and 128 GB of RAM.

4.3 Experimental Results

In this sub-section we present the axes plotted from the results we obtained.

From Fig. 3, we deduce computational efficiency based on run time. Run time increases with increase in both the number of data sets and the number of the data sets’ tuples. For the purpose of getting a clearer picture of the algorithm’s computational performance, we plot axes shown in Fig. 4 (a) and (b). As can be seen the growth rate of run time is almost linearly proportional to the growth rate time-series data sets and their sizes.

Figure 5 shows the behavior of the *FuzzTX* algorithm when we apply parallel processing. Generally, run time decreases as the number of cores increase.

¹ <https://oreme.org>.

Puéchabon: 7 data sets/2-4 attributes/216-line data sets

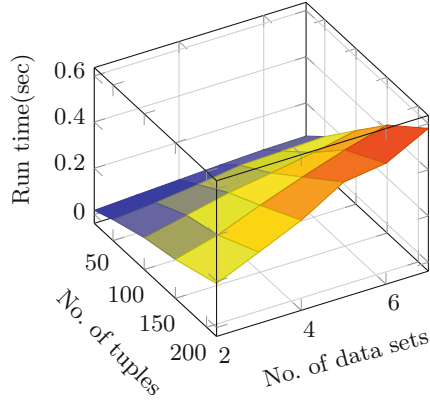


Fig. 3. Plot of run time versus data sets' tuples versus number of data sets

Puéchabon: 7 data sets/4 attributes/216-line data sets

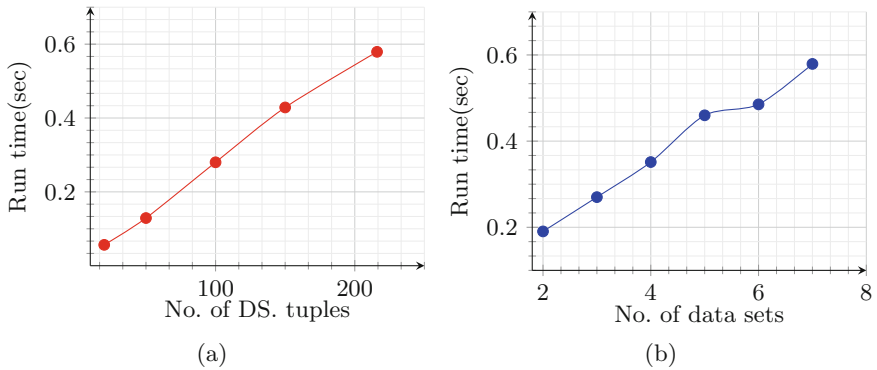


Fig. 4. (a) Plot of run time versus data sets' tuples with number of data sets held at 7 (b) plot of run time versus data sets with number of tuples held at 216

Buoys: 3 data sets/30 attributes/15k, 100k, 200k-line data sets

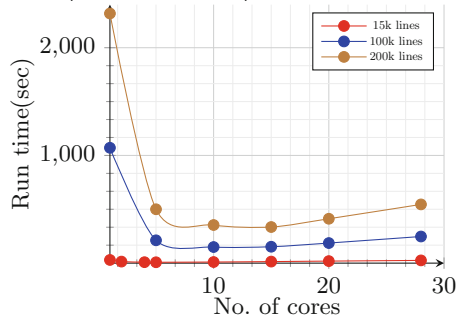


Fig. 5. Plot of run time versus number of cores

Figure 6 (a) and (b) shows the *speedup* and *parallel efficiency* of the *FuzzTX* algorithm. We observe that for each data set, there is an optimum number of processors where parallel efficiency is highest. For the 15k data set it is almost 2 processors; for the 100k and 200k data set it is approximately 5 processors.

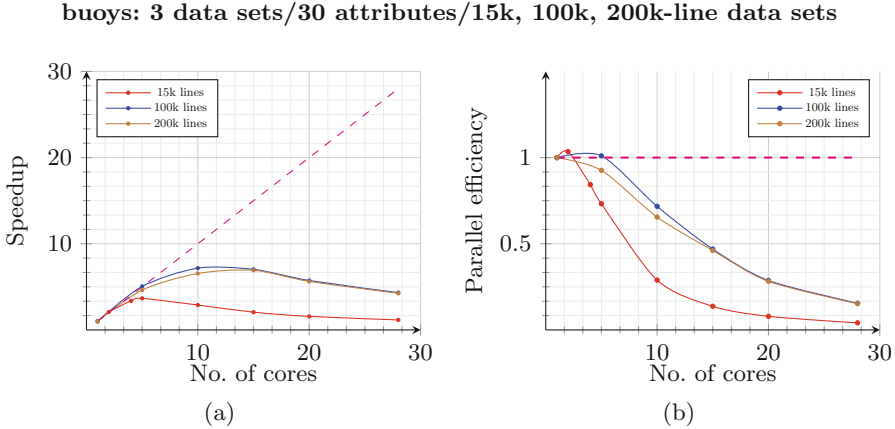


Fig. 6. (a) Plot of speed up versus number of cores (b) plot of parallel efficiency versus number of cores

4.4 Temporal Gradual Patterns

We applied the *T-GRAANK algorithm* proposed by [21] on the crossed data we obtained after applying our *FuzzTX* algorithm on data sets from the Puéchabon weather station. We obtained temporal gradual patterns shown in Fig. 7.

```

Dataset 0k
1 : puechabon_rainfall (mm)**
2 : puechabon_evapotranspiration (mm)
3 : puechabon_global_radiation (MJ.m-2)
4 : puechabon_gross_primary_production (gC/m2)
5 : puechabon_ecosystem_respiration (gC/m2)
6 : puechabon_net (gC/m2)
7 : puechabon_photosynthetic_active_radiation (mol.m-2)
8 : puechabon_air_temperature_average (°C)
9 : puechabon_soil_temperature (°C)
Pattern : Support
('2+', '1+') : 0.5121212121212121 | ~ +4.0 weeks : 0.5
('6-', '1+') : 0.5555555555555556 | ~ +4.0 weeks : 1.0
('7-', '3-', '1+') : 0.5080808080808081 | ~ +4.0 weeks : 1.0
('4+', '5+', '1+') : 0.5070707070707071 | ~ +4.0 weeks : 1.0
('9-', '8-', '1+') : 0.5666666666666667 | ~ +4.0 weeks : 1.0
    
```

Fig. 7. A sample of fuzzy-temporal gradual patterns extracted from crossed data

For instance the pattern $\{('2+', '1+'): 0.5121 \sim +4.0 \text{ weeks}: 0.5\}$ may be interpreted as: “the higher the evapotranspiration, the higher the rainfall amount, almost 4 weeks later”.

5 Conclusions and Future Works

In this paper, we propose a fuzzy model that applies a triangular membership function to cross time-series data sets. This model is most suitable for adoption by research observatories (such as OREME) that support *data lakes* which store time-series data or data streams from different sources.

In order to emphasize the applicability of our model, we integrated the *FuzzTX* algorithm into a *Docker* implementation of the *OGC SensorThings framework* to cross different data streams and extract relevant gradual patterns. The source code for this work is available at our Github repository: <https://github.com/owuordickson/cloud-api.git>.

In future, we intend to extend this work with the aim of constructing a *heuristic bot* that mines a data lake’s catalog in order to identify relevant time-series data sets that may produce interesting knowledge when crossed. A data lake might contain thousands of large and unrelated data sets; therefore, identifying and crossing only interesting data sets instead of crossing all the available data sets is a prudent strategy for saving time and computational resources.

Acknowledgment. This work is part of a Ph.D. thesis and the authors would like to thank the French Government through the office of Co-operation and Cultural Service (Kenya) and the office of Campus France (Montpellier) for their involvement in creating the opportunity for this work to be produced. This work has been realized with the support of the High Performance Computing Platform: **MESO@LR** (<https://meso-lr.umontpellier.fr/faq/>), financed by the Occitanie/Pyrénées-Méditerranée Region, Montpellier Mediterranean Metropole and Montpellier University.

Availability of Materials. The source code for our *FuzzTX* algorithm is available at our GitHub repository: <https://github.com/owuordickson/data-crossing.git>. All the results of our test runs are available at our GitHub link: <https://github.com/owuordickson/meso-hpc-lr/tree/master/results/fuzztx>. Data employed in the research study came from OREME’s Coastline Observation System (<https://oreme.org/observation/lc/>) and an OREME observatory which recorded the meteorological measurements at the Puéchabon site. This data is licensed under a Creative Commons Attribution 4.0 License and the site is annually supported by Ecofor, Allenvi and ANAEE-F (<http://www.anaee-france.fr/fr/>).

References


1. Ayouni, S., Yahia, S.B., Laurent, A., Poncelet, P.: Fuzzy gradual patterns: what fuzzy modality for what result? In: Proceedings of the 2010 International Conference of Soft Computing and Pattern Recognition, SoCPaR 2010, pp. 224–230 (2010). <https://doi.org/10.1109/SOCPAR.2010.5686082>

2. Boukerche, A., Mostefaoui, A., Melkemi, M.: Efficient and robust serial query processing approach for large-scale wireless sensor networks. *Ad Hoc Netw.* **47**, 82–98 (2016). <https://doi.org/10.1016/j.adhoc.2016.04.012>
3. da Costa, R.A.G., Cugnasca, C.E.: Use of data warehouse to manage data from wireless sensors networks that monitor pollinators. In: 2010 Eleventh International Conference on Mobile Data Management, pp. 402–406, May 2010. <https://doi.org/10.1109/MDM.2010.72>
4. Di-Jorio, L., Laurent, A., Teisseire, M.: Mining frequent gradual itemsets from large databases. In: Adams, N.M., Robardet, C., Siebes, A., Boulicaut, J.-F. (eds.) IDA 2009. LNCS, vol. 5772, pp. 297–308. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03915-7_26
5. Eager, D.L., Zahorjan, J., Lazowska, E.D.: Speedup versus efficiency in parallel systems. *IEEE Trans. Comput.* **38**(3), 408–423 (1989). <https://doi.org/10.1109/12.21127>
6. Ecofor, A.: Flux measurements and garrigue ecosystem functioning: Puéchabon site (2019). <https://data.oreme.org/puechabon/graphs>
7. Fernández, A.M., Gutiérrez-Avilés, D., Troncoso, A., Martínez-Álvarez, F.: Real-time big data analytics in smart cities from LoRa-based IoT networks. In: Martínez Álvarez, F., Troncoso Lora, A., Sáez Muñoz, J.A., Quintián, H., Corchado, E. (eds.) SOCO 2019. AISC, vol. 950, pp. 91–100. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-20055-8_9
8. Galicia, A., Talavera-Llames, R., Troncoso, A., Koprinska, I., Martínez-Álvarez, F.: Multi-step forecasting for big data time series based on ensemble learning. *Knowl.-Based Syst.* **163**, 830–841 (2019). <https://doi.org/10.1016/j.knosys.2018.10.009>
9. Gonçalves, N.M., dos Santos, A.L., Hara, C.S.: Dysto-a dynamic storage model for wireless sensor networks. *J. Inf. Data Manag.* **3**(3), 147 (2012)
10. Grothe, M., van den Broecke, J., Linda, C., Volten, H., Kieboom, R.: Smart emission - building a spatial data infrastructure for an environmental citizen sensor network. In: Geospatial Sensor Webs Conference 2016, vol. 1762, pp. 29–31, August 2016
11. Hajj-Hassan, H., et al.: Multimapping design of complex sensor data in environmental observatories. In: Proceedings of the 6th International Conference on Web Intelligence, Mining and Semantics WIMS 2016, pp. 2:1–2:10. ACM, New York (2016). <https://doi.org/10.1145/2912845.2912856>
12. Hajj-Hassan, H., Arnaud, N., Drapeau, L., Laurent, A., Lobry, O., Khater, C.: Integrating sensor data using sensor observation service: towards a methodology for the o-life observatory. *Sens. Transducers* **194**(11), 99 (2015)
13. Hajj-Hassan, H., Laurent, A., Martin, A.: Exploiting inter- and intra-base crossing with multi-mappings: application to environmental data. *Big Data Cogn. Comput.* **2**(3) (2018). <https://doi.org/10.3390/bdcc2030025>
14. Huang, C.Y., Wu, C.H.: A web service protocol realizing interoperable internet of things tasking capability. *Sensors* **16**(9) (2016). <https://doi.org/10.3390/s16091395>
15. Kotsev, A., et al.: Extending INSPIRE to the Internet of Things through Sensor-Things API. *Geosciences* **8**(6) (2018). <https://doi.org/10.3390/geosciences8060221>
16. Laurent, A., Lesot, M.-J., Rifqi, M.: GRAANK: exploiting rank correlations for extracting gradual itemsets. In: Andreasen, T., Yager, R.R., Bulskov, H., Break Christiansen, H., Larsen, H.L. (eds.) FQAS 2009. LNCS (LNAI), vol. 5822, pp. 382–393. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04957-6_33

17. Liang, S., Huang, C.Y., Khalafbeigi, T.: OGC SensorThings API part 1: sensing, version 1.0. (2016)
18. Malysiak-Mrozek, B., Lipińska, A., Mrozek, D.: Fuzzy join for flexible combining big data lakes in cyber-physical systems. *IEEE Access* **6**, 69545–69558 (2018). <https://doi.org/10.1109/ACCESS.2018.2879829>
19. Malysiak-Mrozek, B., Stabla, M., Mrozek, D.: Soft and declarative fishing of information in big data lake. *IEEE Trans. Fuzzy Syst.* **26**(5), 2732–2747 (2018). <https://doi.org/10.1109/TFUZZ.2018.2812157>
20. Mandal, S.N., Choudhury, J., Chaudhuri, S.B.: In search of suitable fuzzy membership function in prediction of time series data. *Int. J. Comput. Sci. Issues* **9**, 293–302 (2012)
21. Owuor, D., Laurent, A., Orero, J.: Mining fuzzy-temporal gradual patterns. In: 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1–6. IEEE, New York, June 2019. <https://doi.org/10.1109/FUZZ-IEEE.2019.8858883>
22. Pitarch, Y., Laurent, A., Poncelet, P.: Summarizing multidimensional data streams: a hierarchy-graph-based approach. In: Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V. (eds.) PAKDD 2010. LNCS (LNAI), vol. 6119, pp. 335–342. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13672-6_33
23. Ronzhin, S., et al.: Next generation of spatial data infrastructure: lessons from linked data implementations across europe. *Int. J. Spat. Data Infrastruct. Res.* **14**, 84–106 (2019)
24. Sahoo, D., et al.: FoodAI: food image recognition via deep learning for smart food logging. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD 2019. ACM Press (2019). <https://doi.org/10.1145/3292500.3330734>
25. Vaidehi, V., Devi, D.S.: Distributed database management and join of multiple data streams in wireless sensor network using querying techniques. In: 2011 International Conference on Recent Trends in Information Technology (ICRTIT), pp. 594–599, June 2011. <https://doi.org/10.1109/ICRTIT.2011.5972459>
26. Wang, L., Chen, L., Papadias, D.: Query processing in wireless sensor networks. In: Aggarwal, C. (ed.) *Managing and Mining Sensor Data*, pp. 51–76. Springer, Boston (2013). https://doi.org/10.1007/978-1-4614-6309-2_3
27. Zadeh, L.: Fuzzy sets. *Inf. Control* **8**(3), 338–353 (1965). [https://doi.org/10.1016/S0019-9958\(65\)90241-X](https://doi.org/10.1016/S0019-9958(65)90241-X)



Cooking Related Carbon Footprint Evaluation and Optimisation

Damien Alvarez de Toledo¹(✉) , Laurent d’Orazio² , Frederic Andres³ ,
and Maria C. A. Leite⁴ 

¹ ENSSAT, 22300 Lannion, France
dalvarez@enssat.fr

² Univ Rennes, CNRS, IRISA, 22300 Lannion, France
laurent.dorazio@irisa.fr

³ National Institute of Informatics, Tokyo 101-8430, Japan
andres@nii.ac.jp

⁴ University of South Florida, St. Petersburg, FL 33701, USA
mcleite@usf.edu

Abstract. Carbon Footprint of foods has been a major concern for the past few years. Food production is responsible for a quarter of all GHG (Green House Gas) emissions. Many food’s Carbon Footprint calculators can be found online but most of them give individual results per ingredient and do not offer a perspective of the whole recipe’s Carbon Footprint. Many factors have to be taken into account for this calculation as the origin of the food, the location of the cooker, but also the way to cook and to assemble ingredients. In this paper, we present the **CROPPER (CaRbon fOotprint reciPe oPtimizER)** that improves an input recipe by updating its ingredients (origin, type) and its cooking procedures to reduce its Carbon Footprint while keeping it savory.

Keywords: Carbon Footprint · Recipe · Food · Health · Climate change

1 Introduction

The Earth’s atmosphere is warming faster than it probably ever has. In some cases weather patterns, climates and natural environments are changing quicker than wildlife or people can adapt. GHG as carbon dioxide, nitrous oxide and methane form emissions have been recognized as partly responsible for the accelerated rate of the climate change [6]. Several reports including standards [5, 10, 11] pointed out the relationship between food consumption and climate change patterns. Also, food production has been recognized to contribute greatly for the anthropogenic environmental disturbances [13]. On the other hand, the

D. Alvarez de Toledo has been a research student at NII during the research period that led to the production of this paper.

choice of our ingredients and ways of cooking has a direct influence on the GHG emissions we produce as citizens (e.g., [16, 18]). Thus, it is important to increase our awareness on both how our food can contribute towards the problem of climate change, and how climate change threatens the supply of ingredients we take for granted. In particular, a conscious approach to the choice of ingredients is vital to reduce GHG and, consequently, our personal environmental impact. The concept of food Carbon Footprint is being increasingly studied by researchers around the world. It has not yet reached, however, a high awareness amongst consumers. The general public as well as specialised chefs need a concise and simple method to evaluate their cooking GHG emissions and fulfilling this awareness. Most information around food's carbon footprint is detailed for separate ingredients and does not take into account the impact that their combination can have as a recipe [3, 6, 15]. Our research, part of the CRWB project [1] aims to overcome this boundary and to assist reducing the individual's Carbon Footprint when cooking a specific recipe, by submitting the recipe to a Carbon footprint optimizer algorithm, the CROPPER.

This paper is structured as follows. Section 2 reviews the state of art. Section 3 describes the Low Carbon Footprint recipe optimiser and our implementation choices. Section 4 introduces the preliminary results. Finally, Sect. 5 concludes and introduces the future works.

2 State of the Art

The concept of Carbon Footprint (CF) appeared around 2006, but the climate impact of products has been calculated for decades as part of full Life cycle assessment (LCA) [15]. Numerous papers about the emitted CF of individual and specific ingredients (e.g., [15, 17] and references therein), but also Danish diets [2] and foods life-cycle in Finnish households [12] are accessible nowadays. However, the computation of CF is still a challenging endeavor. Availability of online CF calculators is quite recent and they focus on single specific ingredients only [3, 6]. Some other calculators try to offer an opportunity for the user to optimize a recipe's environmental impact by swapping ingredients, but they lack an interesting variety of ingredients in their database and are designed for out-of-home catering sector [17]. They are not yet available for private households and their design is in a single language only, not extended to English, which limit their usage by general public worldwide. Furthermore, the available online calculator do not incorporate the CF and budget optimizer feature that is been built into CROPPER. Previous works can be considered as stepping stones for an upcoming system that can evaluate the CF of a recipe in a complete manner, the CROPPER, that "crops" the Carbon Footprint (CF) of an input recipe to make it more environmentally friendly. This system will contain a multilingual service and database, including most European and Asian languages.

3 CROPPER Model

3.1 CROPPER Theoretical Approach

The following theoretical approach aims to reduce the Carbon Footprint (named *CF*) of an input recipe by considering a Desired Carbon Footprint (named *DCF*) and a Money Threshold, both entered by the user. We assume the ingredient similarity between the input and output recipe for this research. When the ingredient is swapped, it can only be replaced by an ingredient from the exact same kind *i.e.* a banana by another banana (different origin, organic instead of conventional, etc.).

We define the **output_recipe_CF**, the CF estimated for the output recipe, as follows:

$$output_recipe_CF = \sum_{i=1}^{nb_Ingredients} CF(ingredient_i), \quad output_recipe_CF \leq DCF$$

where **nb_Ingredients** and **CF(ingredient_i)** denotes, respectively, the number of ingredients and the price of the *ith* ingredient in the recipe.

If **DCF** cannot be attained through ingredient swapping, then the closest **output_recipe_CF** to **DCF** will be returned jointly to the updated recipe. The price of the processed recipe (**recipePrice**) must respect the condition of being inferior or equal to the user’s budget (**Money_Threshold**) as follows:

$$recipePrice = \sum_{i=1}^{nb_Ingredients} Price(ingredient_i), \quad recipePrice \leq Money_Threshold,$$

where **Money_Threshold** is the budget of the User and **Price(ingredient_i)** is the price of the *ith* ingredient of the recipe. The CaRbon fOotprint recipe oPtimizER (CROPPER) ecosystem of this study is represented in (Fig. 1a).

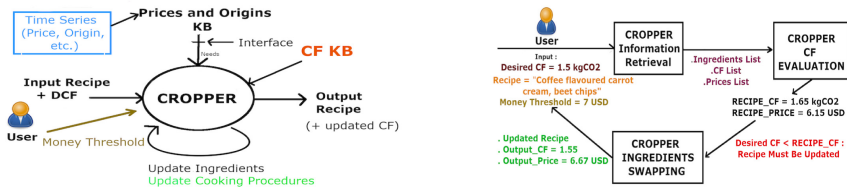


Fig. 1. (a) CROPPER Ecosystem schematic representation. (b) Use-case result tracing.

The user selects a recipe and a desired CF value as an input in addition to a money threshold, which corresponds to the user’s budget. Two conditions must be met for the recipe to be updated (ingredients swapping): **1)** The user must provide the required amount of money for swapping. **2)** The recipe will

be updated when its current CF is higher than the desired one, for a given budget. The final goal of this research is to develop a service that interacts with two knowledge bases **FoodPriceBazaar**¹ and **CFKB**. The former (Prices and Origins KB in Fig. 1a) contains information related to a product’s shop, price and origin. The latter (CF KB in Fig. 1a) provides data on each ingredient’s CF according to their origin. Both of the knowledge bases will be design so that, as opposed to the existing one, each ingredient will have its own semantic signification. Thus, making them more suitable for use compared to raw data included in a conventional database.

3.2 CROPPER Pseudo-code Implementation

The key parts of the CROPPER algorithms are (1) the evaluation of the actual Carbon Footprint (CF) and the cost of the recipe **CF-Evaluation()** in Algorithm 1, and (2) the CF reduction accomplished through the use of **Ingredients-Swapping()** according to the user’s requirements in Algorithm 2.

By default the first ingredients retrieved by **Information-Retrieval()** algorithm, which is not the focus of this article and, therefore, its description it is not included, are the ones that are geographically closest to the cooker’s location (the distance is given in km). The implicit assumption is that these ingredients have a smaller CF. This may not always be true as it has been shown in some recent studies [18]. However, the United Nations recommend to ‘Eat local’ aiming to reduce food related CF. As this is a first developed system that can evaluate and optimise the CF of a recipe, we made this assumption as a starting point. Its relaxation is seen as a future study.

Algorithm 1. CF_Evaluation

```

1: CF_Evaluation (I, P, CF, Money_Threshold, DCF)
2: {
3:   SCF ← 0 # Sum of CFs
4:   SP ← 0 # Sum of Prices
5:   nb_ingredients ← I.sizeof()
6:   # Evaluation of the actual carbon footprint
7:   for i in range(nb_ingredients) do
8:     SCF ← SCF + CF[i]
9:     SP ← SP + P[i]
10:  end for
11:  # Comparison between the CF of our recipe
12:  # and the user desired DCF
13:  if SCF < DCF then
14:    print("The given recipe meets your
15:    requirements.")
16:  else
17:    print("Your recipe needs to be updated.")
18:    Ingredients_Swapping
19:    (I, P, CF, SCF, SP, Money_Threshold, DCF).
20:  end if
21: }
```

Algorithm 2. Ingredients_Swapping

```

1: Ingredients_Swapping
  (I, P, CF, SCF, SP, Money_Threshold, DCF)
2: {
3:   Same_Recipe ← False
4:   i ← 0
5:   Old_Recipe ← I
6:   New_Recipe ← I
7:   while SCF > DCF and ¬(Same_Recipe) do
8:     # Retrieval of a "better ingredient" i.e. the one
      with the closest yet lower CF.
9:     BetterI ← retrieve_better_ingredient(I[i])
10:    PBetterI ← retrieve_price(BetterI)
11:    if (SP - P[i] + PBetterI) < Money_Threshold
      then
12:      New_Recipe[i] ← BetterI
13:      # Retrieval of the ingredient’s Carbon Foot-
      print.
14:      CFBetter ← retrieve_CF(BetterI)
15:      SCF ← SCF - CF[i] + CFBetter
16:      end if
17:      i ← i + 1
18:      if i == nb_ingredients - 1 then
19:        i ← 0
20:        if New_recipe == Old_recipe then
21:          Same_Recipe ← True
22:        else
23:          Old_recipe ← New_recipe
24:        end if
25:      end if
26:    end while
27:    return New_Recipe, SCF
28: }
```

¹ <https://bit.ly/3cF7hP7>.

Observe that if **DCF** cannot be attained by the algorithm `Ingredients_Swapping`, the condition `Old_Recipe == New_Recipe` must be met for the algorithm to stop. The number of ingredients in the database (either the Dummy Database or the future KB) is **finite** and the choice of the ingredients in the algorithm is always oriented towards a “better” one (lower yet closest CF compared to the previous ingredient). We can thus infer that it is not possible to go back to choosing an ingredient with a higher CF and `New_Recipe` is a loop variant. When `New_Recipe` cannot change anymore (no “better” ingredients left in the database) redundancy is met with `Old_Recipe`. The algorithm then stops.

The Knowledge Bases **FoodPriceBazaar** and **CFKB** are yet not available. Hence, the study was conducted on a dummy database created by retrieving information from distinct sources for ingredients CF [4, 8, 9, 14] and prices^{2,3}.

4 Implementation and Results

Our use-case was coded in Python, whilst the dummy database presents three different origins for each ingredient **e.g.** Sour Cream from France, USA or Argentina. The code implementation and the dummy database are available on bitbucket⁴. A study was performed to determine whether it is possible to develop an algorithm that can swipe ingredients in a given recipe aiming to reduce its CF with a pre-specified budget (in USD). The Use-Case used in this study was the dish: “Coffee flavoured carrot cream, beet chips” (see Fig. 1b). After launching our three sub-algorithms altogether, the default recipe is firstly determined with the cooker’s location accordingly. We then obtained an update of our recipe with a lower **CF (1.65 kgCO₂ to 1.55 kgCO₂)** yet a higher price (**6.15 USD to 6.67 USD**). The results of our first implementation remain substantially simplistic in the sense that the choice of variety for each ingredient is not very broad in our dummy database (three for each). More significant results will be obtained when the connection with our two Knowledge Bases (Prices and CF) is materialized.

5 Conclusion and Future Work

Climate change is happening at high speed and has many origins, and our diet is one of them. Developing models to reduce our environmental impact when cooking/eating is thus a valuable goal. In this study, we present a Carbon Footprint (CF) recipe optimiser which has a number of benefits over existing calculators. Current CF calculators for foods focus on one ingredient at a time and thus lack the assembling feature for getting the environmental impact of a whole recipe.

² Walmart, www.walmart.com. Last accessed 25/4/2020.

³ Climate change food calculator: What’s your diet’s carbon footprint?, www.bbc.com/news/science-environment-46459714. Last accessed 23/4/2020.

⁴ <https://bit.ly/2zbpQNe>.

The findings in this paper can be used as a base for a better recipe updater that takes into account ingredients pairing and also the proper connectivity to the relevant knowledge bases. The latter will provide a large scope of ingredients variety. Future features are its scalability and bulking, allowing the simultaneous feeding of many recipes as an input, which will yield more diversified results.

Our current model and preliminary tests show that it is possible to compute a new recipe given a certain money budget and a desired CF. Thus, it satisfies our query for a better CF for an input recipe. This novel study has the advantage of proposing a clear design of a recipe CF optimiser, easily understood and visualized through the pseudo-code and model exposed. Although the process is limited to summations for now, it offers the capability of processing a whole recipe at once, and not forcing the user to enter each ingredient with its respective volume. The three main limitations of our approach is the transport CF of ingredients, find eco-friendly cooking procedures, and the access to the target knowledgebases have not yet fully implemented. Additionally, to improve the quality of the optimised recipes, the Ingredient pairing feature will be included.

Acknowledgements. We would like to express our deepest appreciation to the National Institute of Informatics for the ongoing research support.

References

1. Andres, F.: The CRWB RSBench: towards a cooking recipe benchmark initiative. In: IEEE ICDE Workshops 2018, pp. 154–156 (2018). <https://doi.org/10.1109/ICDEW.2018.00032>
2. Bruno, M., Thomsen, M., Pulselli, F.M., Patrizi, N., Marini, M., Caro, D.: The carbon footprint of Danish diets. *Clim. Change* **156**(4), 489–507 (2019). <https://doi.org/10.1007/s10584-019-02508-4>
3. Carbon food calculator-the vegan society. <https://bit.ly/2Xwo5U2>. Accessed 29 Apr 20
4. Chart: The carbon footprint of the food supply chain. <https://bit.ly/2MtIDHx>. Accessed 25 Apr 20
5. Clark, M., et al.: Multiple health and environmental impacts of foods. *Proc. Nat. Acad. Sci.* **116**, 23357–23362 (2019)
6. Climate change indicators: greenhouse gases. <https://www.epa.gov/climate-indicators/greenhouse-gases>. Accessed 29 Apr 20
7. Coffee's invisible carbon footprint. <https://bit.ly/2U9Yoqi>. Accessed 25 Apr 20
8. Do you know the carbon footprint of these common foods. <https://bit.ly/2Xwp06W>. Accessed 25 Apr 20
9. Émissions indirectes - autres, Restauration, Repas. <https://bit.ly/2AJZcew>. Accessed 25 Apr 20
10. Folk and Knife: Food & CO₂, what are the carbon emissions of different foods? (2019). <https://folkandknife.com/wp-content/uploads/2019/05/Food-CO2-2.pdf>
11. Guide to PAS 2050: How to assess the carbon footprint of goods and services. British Standard PAS 2050, p. 58 (2008). ISBN 978-0-580-64636-2
12. Kauppinen, T., et al.: Carbon footprint of food-related activities in Finnish households. *Prog. Ind. Ecol. Int. J.* **7**, 257–267 (2010)

13. Poore, J., Nemecek, T.: Reducing food's environmental impacts through producers and consumers. *Science* **360**(6392), 987–992 (2018)
14. Réduire sa consommation de sel: pourquoi? <https://bit.ly/3aERwXk>. Accessed 25 Apr 20
15. Rööös, E.: Analysing the carbon footprint of food. *Acta Universitatis Agriculturae Sueciae (1652–6880)* 2013(56), 0–96 (2013)
16. Sandström, V., et al.: The role of trade in the greenhouse gas footprints of EU diets. *Global Food Secur.* **19**, 48–55 (2018)
17. Speck, M., et al.: Creating sustainable meals supported by the NAHGAST online tool - approach and effects on GHG emissions and use of natural resources. *Sustainability* **12**, 1136 (2020)
18. Weber, C.L., Matthews, S.: Food-Miles and the relative climate impacts of food choices in the United States. *Environ. Sci. Technol.* **42**(10), 3508–3513 (2008)

**2nd Workshop on Modern Approaches
in Data Engineering and Information
System Design (MADEISD 2020)**



CrEx-Wisdom Framework for Fusion of Crowd and Experts in Crowd Voting Environment – Machine Learning Approach

Ana Kovacevic^{1,2}(✉) , Milan Vukicevic¹ , Sandro Radovanovic¹ ,
and Boris Delibasic¹ 

¹ Faculty of Organizational Sciences, University of Belgrade, Belgrade, Serbia
ak20195017@student.fon.bg.ac.rs,
{milan.vukicevic, sandro.radovanovic,
boris.delibasic}@fon.bg.ac.rs

² Saga LTD, Belgrade, Serbia

Abstract. In recent years crowd-voting and crowd-sourcing systems are attracting increased attention in research and industry. As a part of computational social choice (COMSOC) crowd-voting and crowd-sourcing address important societal problems (e.g. participatory budgeting), but also many industry problems (e.g. sentiment analyses, data labeling, ranking and selection, etc.). Consequently, decisions that are based on aggregation of crowd votes do not guarantee high-quality results. Even more, in many cases majority of crowd voters may not be satisfied with final decisions if votes have high heterogeneity. On the other side in many crowd voting problems and settings it is possible to acquire and formalize knowledge and/or opinions from domain experts. Integration of expert knowledge and “Wisdom of crowd” should lead to high-quality decisions that satisfy crowd opinion. In this research, we address the problem of integration of experts domain knowledge with “Wisdom of crowds” by proposing machine learning based framework that enables ranking and selection of alternatives as well as quantification of quality of crowd votes. This framework enables weighting of crowd votes with respect to expert knowledge and procedures for modeling trade-off between crowd and experts satisfaction with final decisions (ranking or selection).

Keywords: Crowd voting · Experts · Machine learning · Clustering · Matrix factorization

1 Introduction

Inclusion of crowd in decision-making processes may not only result in greater crowd satisfaction, but also higher quality and timeliness of decisions even if they are compared to decisions made by limited number of experts [5]. This is due to phenomena of “Wisdom of Crowd” or “Collective Intelligence” that have theoretical roots in Condorsets Jury theorem. This theorem states that given a group of independent voters (a “jury”) that

have probability of correct outcome (alternative) $1 \geq p \geq 0$ and incorrect outcome of $1-p$, probability of choosing correct outcome by majority voting increases by adding more voters if probability of the correct outcome of each voter is greater than random choice (e.g. $p > 0.5$ in case of binary outcome). Even though this theory has strong assumption on voter independence that is not fulfilled in many real-world scenarios and several other limitations, it showed cutting edge results in many application areas and problems of ranking, selection, prediction, etc. Over the last few years, Collective Intelligence (CI) platforms have become a vital resource for learning, problem-solving, decision-making, and predictions [8] and led to development of numerous frameworks. Adequate technology support and desirable properties of crowd-voting systems led to wide acceptance of crowd/voting as a tool for solving both industry and societal problems [22].

In societal problems inclusion of crowd in decision making should lead to greater satisfaction and welfare. Additionally, “Wisdom of crowds” may be exploited in order to make high-quality decisions, while satisfying crowd opinion. Collection of votes from the general population can be encouraged by the following main reasons [22]: democratic participating in political elections and policymaking (e.g., law regulation [18]); solving issues from common interest (e.g., budget allocation – Knapsack voting and participatory budgeting [19] or resolving a different kind of issues in the field of education, health, etc.).

For many industry problems, companies adopt “Crowd Intelligence” in order to automate processes increase quality of their products and services and reduce costs. For example: choosing innovative ideas that should be adopted [20]; giving feedback on creative works [10]; making recommendations based on users’ critical rating [21]; stock market predictions [3]; selecting winners in competitions (e.g. TV music competitions such as Eurovision Song Contest, American Idol, etc.), and others.

However, implementation of knowledge and patterns identified in information collected from crowd in both societal and industry settings poses a significant challenge. Most of the problems that are inherited from assumptions of Condorcet’s jury theorem that are not fulfilled in most of the real world application problems. Some of the major problems are:

- Incompetence, lack of interest, favoritism and manipulation of the crowd for problem at hand [9],
- Bias in ordinal voting systems [6],
- Sparse and imbalanced data generated from crowd votes [2],
- Etc.

We hypothesize that exploitation of expert knowledge (even with single or limited number of experts) may address many problems of crowd voting quality, while preserving advantages of “Wisdom of Crowd” and “Collective Intelligence”.

In this paper, we present a framework that enables fusion of experts’ domain knowledge based on unsupervised machine learning approach. The main idea of the framework is to use limited number (or single) of expert inputs in order to weight crowd votes. In this way, we pose the problem of vote aggregation as a minmax problem: minimization of distance from experts and maximization of crowd satisfaction. We address this problem

by estimation of density and similarity of votes between crowd and experts through clustering and outlier detection. Additionally, we address the problem of sparseness of votes by using matrix factorization techniques that showed cutting edge results in the area of recommender systems based on collaborative filtering. Such factorization enables not only dimensionality reduction and solving sparsity problem, but also extraction of latent features that represent affinities of crowd and expert voters. Affinities in dense format enable definition of good quality distance/similarity measures but also estimation of voters' preferences towards alternatives that they did not voted for (or gave rating). In experimental part of this paper, we show usefulness of our approach on the Eurosong contest ranking problem. We compare the results in terms of both expert and crowd satisfaction by final ranks with two benchmarks: official Eurosong voting aggregation procedure and newly weighted voting procedure that does not exploit benefits of latent feature space.

The contribution of this paper is twofold:

1. We propose a framework for unsupervised machine learning based aggregation of crowd and expert opinions.
2. We provide an experimental evaluation of the framework and make additional insights on crowd performance based on characteristics of crowd and experts opinions.

2 State-of-the-Art

Exhaustive and systematic review on Collective Intelligence (CI) platforms including 9,418 scholarly articles published since 2000 recently is presented in [8]. Additionally, in our previous work [22] we provided detailed review and analyses of advantages and disadvantages of expert-based and crowd-based decision making systems that are summarized in Table 1. Thus, in this literature review we will focus only on research that is closest to current research with special focus on similarities and differences and compatibility between similar approaches and the one proposed in this paper.

Usage of matrix factorization in CI is not a new idea. There are a numerous examples where latent features are extracted to help the process of decision-making. One such example is filling missing values in crowd judgments [2]. Majority of the voters in the CI process express their judgments for only several alternatives (out of a much larger set of alternatives) thus leaving votes sparse and imbalanced. Consequently, decision-making process yields in undesirable solutions. As a part of solution one can employ probabilistic matrix factorization techniques. As a result, votes are going to be imputed with the most probable values. By having a full voters data matrix more reliable solutions can be obtained.

However, matrix factorization is seldom used for imputation of missing values. More often one uses matrix factorization to investigate crowd characteristics and for validation of the crowd. One such example is presented in paper [5]. Namely, factorization using pBOL method is used for validation of crowdsourced ideas based on expert opinions. The method provides idea-filtering techniques that reduces the number of crowdsourced ideas that will be manually evaluated by experts. This is achieved by creating a predictive model

based on latent features which predict opinion of each expert about the crowdsourced idea. In order to reduce false negatives, the task is transformed from selecting the good ideas to eliminating the poor ones. Compared to pBOL, our framework is used for crowd and expert weighting (instead of filtering), thus allowing automated estimation of importance of crowd votes as well as aggregation of the final solution.

Table 1. Experts vs. crowd – different aspects of collective decision-making [22]

Different aspects		Experts	Crowd
Voting properties [10]:	Selector qualification	Small groups of qualified judges	Many voters with unknown qualifications
	“Selector-selectee” relationship	Experts as an independent body	Voters and candidates can be from the same crowdsourcing community
	Selection process	A systematic voting process with justified evaluation criteria	An uncertain voting process in which participants have their own evaluation criteria
Decision making relies on...		...intuition and reasoning at the same time [11]	...collective intelligence and “wisdom of the crowd” [12]
Application		Complex decision-making problems (e.g., MCDM methods with mutual interdependencies of criteria [13])	Social choice topics with respect to individual preferences as a central topic of AI [14]
Challenges		Possible difficulties in multidisciplinary decision making [15]	Impossibility theorems in Social Choice Theory (e.g., Arrow’s theorem [16], the Condorcet paradox [17])

It is worth to mention SmartCrowd framework proposed by [7] that allows 1) characterization of the participants using their social media posts with summary word vectors, 2) clustering of the participants based on these vectors, and 3) sampling of the participants from these clusters, maximizing multiple diversity measures to form final diverse crowds. They show that SmartCrowd generates diverse crowds and that they outperform random crowds. They estimate the diversity based on external data (tweets). In a sense, this research also tries to estimate diversity of crowds but with respect to both crowd and expert members and without external information.

Expert weighting has also been done in the CI area. One can find such an example in paper [4] where the task was to assign weight to voters for stock pick decisions. This

was done using metaheuristics, namely genetic algorithm. Information about previous judgments and their accuracy as well as additional information (i.e. sentiment analysis from social media) are inserted in genetic algorithm that produces a probability that a crowd voter is an expert. As a result, a framework has a predictive model that can be used for future crowd voters. They showed better average performance than the S&P 500 for two test time periods, 2008 and 2009, in terms overall and risk-adjusted returns. However, this approach assumes existence of historical data (and other additional information) to be available at the predicting model-learning phase and for evaluation of a new crowd voter. In majority of CI examples, one cannot expect to have such an amount of information about crowd voters. Thus it allows weighting and aggregation of crowd and expert votes without collection of additional data. Unsupervised approach seems like an intuitive solution. Unsupervised approach would represent identification of the experts from the crowd voters by using only current votes. We propose one such approach based on similarity matching of experts and crowds. As a result, it is expected to have better decision-making process with greater satisfaction of both crowd and experts.

However, bias in crowd-voting systems can exist. In paper [6], one can find an investigation of the influence of bias in crowd-voting systems with a special focus on ordinal voting. They showed that ordinal rankings often converge to an indistinguishable rating and demonstrated this trend in certain cities for the majority of restaurants to all have a four-star rating. Additionally, they also show that ratings may be severely influenced by the number of users. Finally, they conclude that user bias in voting is not a spam, but rather a preference that can be harnessed to provide more information to users. Based on analyses of global skew and bias they suggest explicit models for better personalization and more informative ratings. Even though research of [6] does not model expert and crowd votes, their research is highly applicable to framework that we are proposing in this paper, because performance of the framework is highly dependent on skew and bias in the data.

3 Framework for Expert-Crowd Voting

Based on opportunities and challenges of crowd voting, as well as potential of benefits of integration of crowd and domain expert knowledge we propose CrEx-Wisdom (Crowd and Expert Wisdom) framework for fusion of experts and crowd “Wisdom” for problems of participatory voting and ranking. The main idea of the framework is to utilize knowledge from a limited number of experts in order to validate and weight crowd votes. Another important aspect that we want to model is the agreement (variance) of both expert and crowd votes as well as their mutual agreement in order to address problems of bias in crowd-voting (described in previous sections). It is important to note that the proposed framework can work in a completely unsupervised manner. This means that expert efforts are reduced to giving opinion on the problem by ranking or grading subset of alternatives, without the need for validation of crowd votes or tracking of crowd voters’ performance history or adding external data. Finally, we try to address the problem of sparsity and aggregation of crowd and expert votes. The data flow of the proposed framework is depicted on Fig. 1.

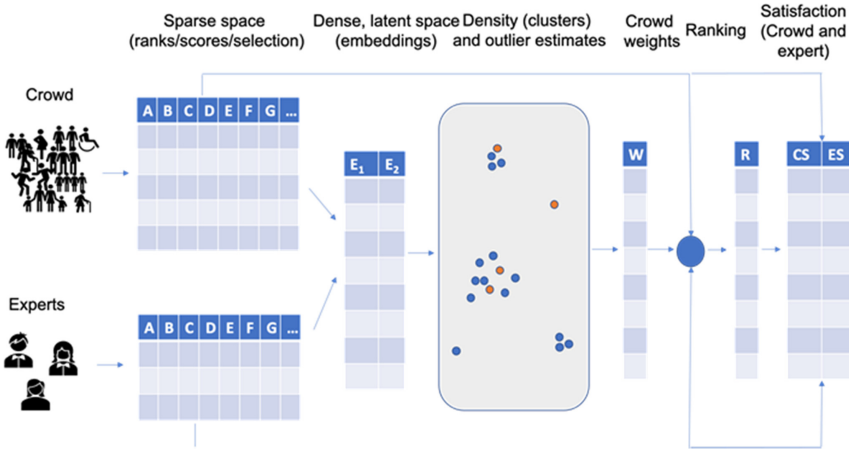


Fig. 1. CrEx-Wisdom framework – data flow

General data and process can be described in following way:

- Experts and crowd are providing votes (ranks, grades, etc.) that are stored in a sparse format.
- Votes of both expert and crowd groups are aggregated in one dataset.
- Latent features (embeddings) are identified based on machine learning (i.e. collaborative filtering) methods.
- Based on latent space of features agreement between experts and crowd (and their mutual agreement) is quantified with machine learning methods such as clustering and outlier detection.
- Based on estimated agreement levels votes of both experts and crowd are weighted on individual level (each voter may have unique weight).
- Votes are aggregated based on traditional methods (e.g. weighted majority) and converted to ranks or grades.
- After aggregation expert satisfaction and crowd satisfaction are measured, and a Pareto front of non-dominated solutions is generated.

CrEx-Wisdom framework provides quite general guidance for fusion of crowd and expert votes in terms of selection of methods and techniques in each step.

In the latent features identification phase, we use the matrix factorization algorithm Alternating Least Squares (ALS) [23] in order to learn latent user and alternative factors. Matrix factorization assumes that each user can be described by k attributes (factors), and each alternative can be described by an analogous set of k attributes (factors). The final prediction (rating) is obtained by multiplication of these two matrices of the voter and alternative factors in order to get a good approximation of missing user ratings. Final model can be represented as (1):

$$\hat{r}_{ui} = x_y^T \cdot y_i = \sum_k x_{uk} y_{ki} \tag{1}$$

where \hat{r}_{ui} represents prediction for the true rating r_{ui} , and $y_i(x_u^T)$ is assumed to be a column (row) vector of user and items called latent vectors of low-dimensional embeddings. Loss function that we used is minimizing the square of the difference between all points in our data (D). Formula of loss function is given in (2):

$$L = \sum_{u,i \in D} (r_{ui} - x_u^T \cdot y_i)^2 + \lambda_x \sum_u \|x_u\|^2 + \lambda_y \sum_u \|y_i\|^2 \quad (2)$$

We also added on two regularization terms in order to prevent overfitting of user and alternative vectors.

ALS algorithm is selected because of cutting edge performance in terms of ranking quality, but also because of its scalability that enables work with big data. Additionally, ALS (and other matrix factorization algorithms) provides convenient representation of both voter and alternative spaces. As such it is important since it allows characterization and application of clustering and/or outlier detection techniques in space of the voters as well as in the space of alternatives.

On the other hand, many different popular techniques may be used e.g. autoencoders, Word2Vec [24], Glove [25], and similar algorithms that showed cutting edge performance in NLP (Natural Language Processing) problems.

Similarly, in this research, we used the well known K-means algorithm [26] (clustering) and Isolation forest [27] (outlier detection) for estimation of voters agreement (density, variance), but we acknowledge that other types of algorithms may be used and possible achieve even better results. However, this investigation is out of the scope of this research since the objective is to show value of integration of crowd and expert votes with machine learning approach.

Considering that the goal of this research is to maximize crowd satisfaction with respect to expert opinion, we used two metrics. The first metric is Satisfaction, which we define as expected value of alternatives number that overlaps with the final decision. This metric does not take into account the ranks of alternatives; we consider that one is satisfied if their favorite alternative is chosen in the first ten ranks. The formula for this metric is given in (3):

$$\text{Overlap}_{wi} = \sum_{j=1}^n (x_{wj} * x_{ij}) \quad (3)$$

$$E(\text{Overlap}) = \sum_{i=0}^{k=10} p(\text{overlap}) * \text{overlap}$$

Where:

n – Number of alternatives (countries, songs);

k – Number of selected (winning) alternatives;

x_{wj} – A boolean value of j-th alternative;

x_{ij} – A boolean value of j-th alternatives for i-th voter.

We considered that the rank difference is more important and in order to capture it, we evaluated our methods using the average points difference from winning combination of alternatives. The formula of this metric is given in (4).

$$\text{avg PD} = \frac{1}{m} \sum_{i=1}^m \sum_{j=1}^n |x_{wj} - x_{ij}| \quad (4)$$

Where:

m – Number of voters;

n – Number of alternatives;

x_{wj} - Winning alternative points at rank j ;

x_{ij} - Alternative points of i -th user at rank j .

4 Experimental Evaluation

In this research, we analyzed the problem of aggregation of crowd and expert votes from the Eurovision song contest. In this contest crowd is represented by televoting participants for each country.

4.1 Data

Votes are aggregated for every country by experts and crowd (televoting), which means, that we had the same number of instances for experts and for crowd. Data used in our experiments are from three years: 2016, 2017, and 2018 for all types of events, which include first-semifinal, second-semifinal, and grand-final. Votes from each country, for both experts and crowd (televoting participants) have a total points range of 58 which is distributed as 1 to 8, 10 and, 12 points. The number of countries that have the right to vote in grand-final is 42 and they can choose from 26 countries that took part in the final contest. In the semifinals, the number of countries that can vote is 21 each and they have the option to choose from 18 available songs. Out of fairness is not allowed for countries to vote for themselves. In a current Eurovision voting setup, the final decision is made by weighting crowd and experts evenly.

4.2 Experimental Setup

We conducted several experiments for different voting methods. In order to compare our methods we used two benchmark methods, one is the current Eurovision weighting method and we created a simple “Single Weighting Crowd” method based on distance from experts.

The Benchmark model that we created is based on weighting voters based on distance from experts. Distance is calculated for each crowd participant to every expert, and then the minimum value is converted to similarity, which represents the weight of a particular voter in the crowd. Every crowd vote is multiplied with its calculated weight, and then the crowd data is summarized together with expert votes in order to get final winning ranking. It is important to note that in this similarity definition latent features (matrix factorization) were not used for representation of voter and alternative space, but rather sparse ranks from original data.

In order to find latent factors (embeddings) of voters and alternative spaces, and consequently define similarities between voters we optimized Alternating Least Squares (ALS) that we trained using Mean Absolute Error. Training is done by splitting data on train and test set. Expert and crowd votes are used together and part of their votes has

been masked and used for measuring error on test set. Several hyper-parameters have been optimized in order to minimize error on test data. Greed search of parameters is shown in the table below (Table 2).

Table 2. Hyperparameter greed search optimization of ALS

Parameter	Grid of values
Number of latent factors	[2, 3, 5, 7, 10, 15, 20]
Regularizations	[0.01, 0.1, 1.0, 10, 100]

After this procedure the best parameters were found, single weight of every crowd is determined as a distance of crowd factor data and expert factor data, which is converted to similarity and used to weigh every crowd vote with corresponding similarity weight.

Further, we tried to identify homogenous groups of experts and describe them with representatives (centroids). These representatives enabled us to simulate situation of much smaller number of experts. Additionally, we used these centroids for measuring similarity with the crowd and assign weights to each crowd participants. Based on exploratory analysis of factor data we saw that there are experts that form homogeneous groups. Hence we used K-means algorithm where we optimize the number of clusters for every data set using Silhouette index as a measure of clusters quality. Here K-means can be replaced with any other cluster algorithm with different measures of quality of detected homogenous groups.

Additionally, we identified outliers in embedded space and conducted the same experimental procedure but with outliers removed from the data.

It can be concluded from description of CrEx-Wisdom framework and experimental setup, crowd votes are weighted based on similarity with expert votes. This means that overall satisfaction of expert voters should increase compared to the current contest voting method (aggregation of expert and crowd votes with equal weights). Therefore, we evaluate the proposed methods in Pareto terms: maximize satisfaction of experts while minimizing “dissatisfaction” of crowd compared to the current voting procedure.

4.3 Results and Discussion

As explained earlier, we used two evaluation metrics one that takes into account only overlapping of selected alternatives with crowd and expert votes and another one that includes rank differences using the number of points given to each rank. Due to space limitation, we will discuss only results of average points difference.

Figure 2 shows percentage of change of voter and expert satisfaction (blue and orange bars, respectively) compared to current voting system:

- for each method (x-axis)
- each event and each competition year (y-axis)

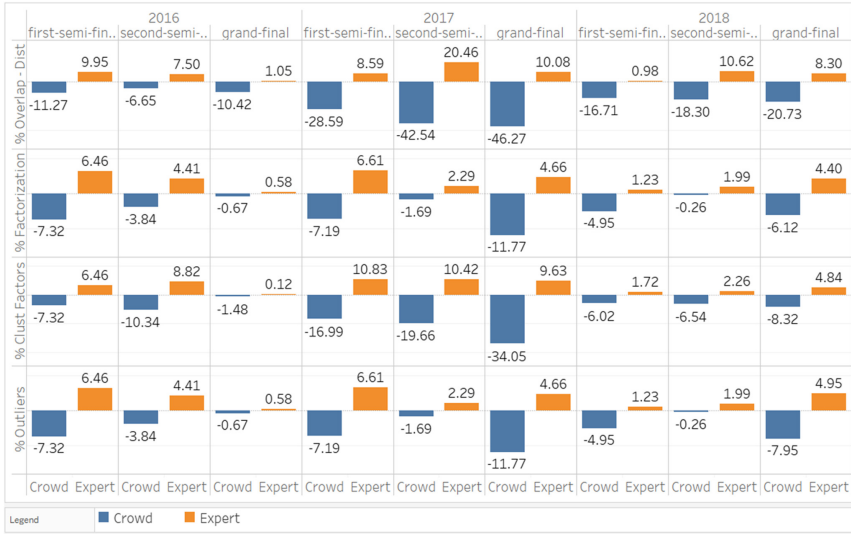


Fig. 2. Relative change in points difference with regard to Eurovision voting (Color figure online)

It can be seen that these changes vary over both years, events and proposed methods.

In order to easier spot differences in performance between methods, the relative change of satisfaction is expressed as ratio of absolute crowd percentage change over absolute expert change (showed in detail in Fig. 2) and presented in Table 3. This ratio practically shows decrease of crowd satisfaction for unit increase of expert satisfaction. Meaning that best results are achieved with minimal values in Table 3.

Table 3. The ratio between crowd and expert change in points

Year	Event type	Overlap	Factorization	Clust – Factors	Outliers
2016	first-semi-final	1.13	1.13	1.13	1.13
2016	grand-final	9.90	1.15	12.64	1.15
2016	second-semi-final	0.89	0.87	1.17	0.87
2017	first-semi-final	3.33	1.09	1.57	1.09
2017	grand-final	4.59	2.53	3.54	2.53
2017	second-semi-final	2.08	0.74	1.89	0.74
2018	first-semi-final	17.00	4.03	3.50	4.03
2018	grand-final	2.50	1.39	1.72	1.61
2018	second-semi-final	1.72	0.13	2.90	0.13

It can be seen from Table 3 that factorization and outlier detection methods are the best performing in most of the cases. However, there are some exceptions. In year 2016

in the first semifinal, we can see that all methods had the same results. We conducted more detailed inspection of embedded data (Fig. 3) that we compressed using T-SNE algorithm in order to visualize points in a two-dimensional space. On these graphs, every point is colored - blue for crowd group, and orange for expert group. It is important to note that the shape represents corresponding cluster labels and that for the convenience of visualization the whole crowd is represented as one cluster (labeled “-1” since only expert data were clustered).

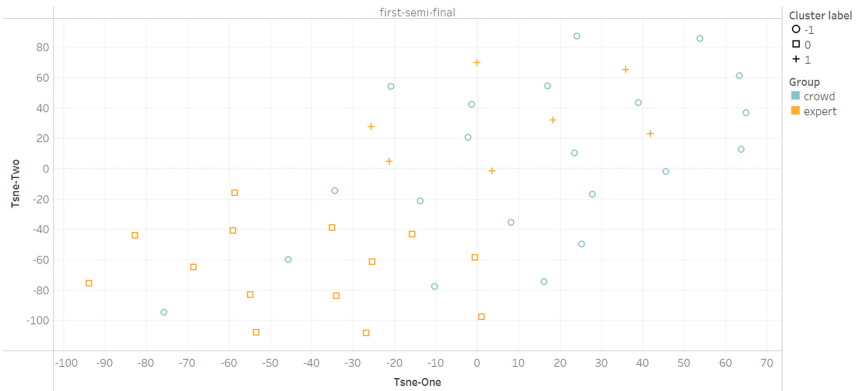


Fig. 3. TSNE 2016 first-semifinal (Color figure online)

Analyzing Fig. 3 we can conclude that the same performance of all voting methods is because of a high dispersion of data. It is clear that there are no homogenous groups neither within expert group or crowd group. Similarly, there are no outliers in this data. On the other hand, in 2016 in grand final, it can be seen that factorization notably outperforms clusters. On Fig. 4 are shown data of this event.

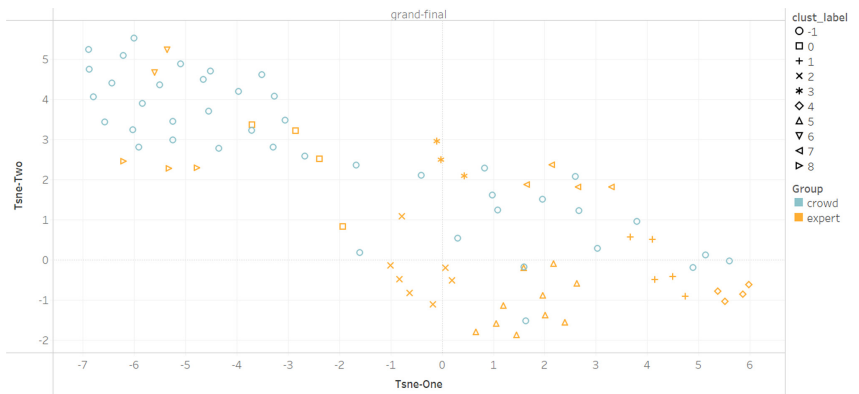


Fig. 4. TSNE grand final in 2016

It can be seen from Fig. 4. That our clustering optimization method found 9 clusters of experts. Such a large number of clusters with respect to number of instances (several clusters have only two or three members) reveals high diversity in expert opinions that is emphasized even more by representing cluster of experts with centroids. We hypothesize that usage of other types of clustering algorithms such as hierarchical clustering could lead to better quality grouping with respect to cluster density. From Fig. 4 it can be seen that most of the experts and a significant number of crowd voters are grouped in the lower right part of the space. This could mean that the final decision (ranking) should be positioned in that part of the space in order to maximize satisfaction of experts and minimize dissatisfaction of crowd.

Additionally, in 2018 in first-semifinal there is a situation where clusters outperform factorization. On Fig. 5 we can see that the cluster algorithm found five quite homogeneous clusters which diminish the variance of expert votes. Based on those groups similarity of crowd is better generalized and thus result from Table 3 is better compared to other methods.

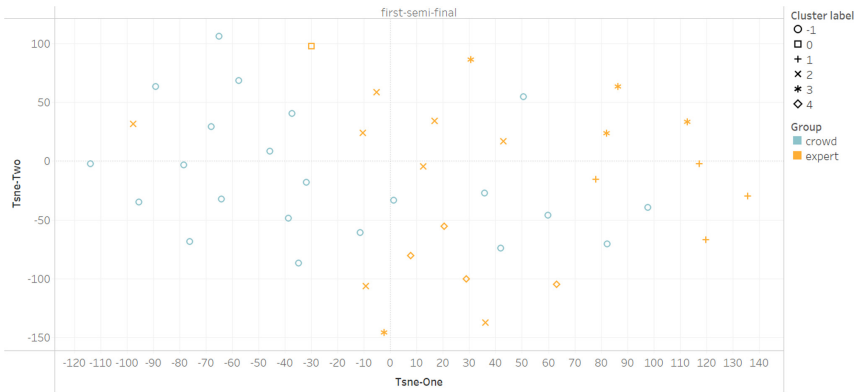


Fig. 5. TSNE 2018 first-semifinal data

In addition, one of the reasons for these results might come from the nature of data used for the experiments. Pop music culture is an area where subjective opinions are highly expected. Moreover, the bias in voting between neighboring countries is present and could be seen from the history of voting. Despite all these unfavorable factors, we showed that in cases where at least part of voters (crowd and/or experts) are homogeneous it is possible to increase crowd/expert satisfaction.

5 Conclusion and Future Research

In this paper, we proposed a framework for integration of expert and crowd votes with the idea of achieving good quality solutions that respect to expert opinion and crowd satisfaction. Results showed that weighting of crowd voters on the individual level, representation of votes in latent space and estimation of consensus level between voters with

clustering and outlier detection procedures can have good impact on finding solutions that compromise between crowd and experts, even if these groups are quite different. In future work, we plan to evaluate more machine learning methods for embedding of votes in latent spaces, clustering and outlier detection. Additionally, we plan to analyze results from this research on theoretical level in terms of voters bias, mutual information between experts and crowd, and densities of crowds and experts. Additionally we plan to validate approach against different voting data (e.g. curriculum creation, best paper awards etc.) where we expect less bias and more consistent voting from experts.

Acknowledgments. This paper is a result of the project ONR - N62909-19-1-2008 supported by the Office for Naval Research, the United States: *Aggregating computational algorithms and human decision-making preferences in multi-agent settings.*

References

1. Traunmueller, M., Fatah, G., Schieck, A.: Introducing the space recommender system: how crowd-sourced voting data can enrich urban exploration in the digital era. In: Proceedings of the 6th International Conference on Communities and Technologies, pp. 149–156 (2013)
2. Jung, H.J., Lease, M.: Inferring missing relevance judgments from crowd workers via probabilistic matrix factorization. In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1095–1096 (2012)
3. Hong, H., Du, Q., Wang, G., Fan, W., Xu, D.: Crowd wisdom: the impact of opinion diversity and participant independence on crowd performance. In: Twenty-second Americas Conference on Information Systems (2016)
4. Hill, S., Ready-Campbell, N.: Expert stock picker: the wisdom of (experts in) crowds. *Int. J. Electr. Commer.* **15**(3), 73–102 (2011)
5. Garcia, A.C., Klein, M.: pBOL: an idea filtering method based on negative multi-voting and Pareto aggregation (2017). <http://dx.doi.org/10.2139/ssrn.3175329>
6. Lees, A., Welty, C.: Discovering user bias in ordinal voting systems. In: Companion Proceedings of the 2019 World Wide Web Conference, pp. 1106–1110 (2019)
7. Bhatt, S., Chen, K., Shalin, V.L., Sheth, A.P., Minnery, B.: Who should be the captain this week? Leveraging inferred diversity-enhanced crowd wisdom for a fantasy premier league captain prediction. In: Proceedings of the International AAAI Conference on Web and Social Media, vol. 13, no. 01, pp. 103–113 (2019)
8. Suran, S., Pattanaik, V., Draheim, D.: Frameworks for collective intelligence: a systematic literature review. *ACM Comput. Surv. (CSUR)* **53**(1), 1–36 (2020)
9. Dodevska, Z.A.: Computational social choice and challenges of voting in multi-agent systems. *Tehnika* **74**(5), 724–730 (2019)
10. Chen, L., Xu, P., Liu, D.: The effect of crowd voting on participation in crowdsourcing contests. In: Working paper (2019). 39 pages
11. Bennet, A., Bennet, D.: The decision-making process for complex situations in a complex environment. In: Burstein, F., Holsapple, C.W. (eds.) *Handbook on Decision Support Systems 1*, pp. 3–20. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-48713-5_1
12. Yu, C., Chai, Y., Liu, Y.: Literature review on collective intelligence: a crowd science perspective. *Int. J. Crowd Sci.* **2**(1), 64–73 (2018)
13. Mandic, K., Bobar, V., Delibašić, B.: Modeling interactions among criteria in MCDM methods: a review. In: Delibašić, B., Hernández, J.E., Papathanasiou, J., Dargam, F., Zaraté, P., Ribeiro, R., Liu, S., Linden, I. (eds.) *ICDSST 2015. LNBP*, vol. 216, pp. 98–109. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-18533-0_9

14. Rossi, F.: Preferences, Constraints, Uncertainty, and Multi-Agent Scenarios. ISAIM (2008)
15. Jackson, S.E.: The consequences of diversity in multidisciplinary work teams. In: West, M.A. (ed.) *Handbook of Work Group Psychology*, pp. 53–75. Wiley, Chichester (1996)
16. Miller, N.R.: Reflections on Arrow’s theorem and voting rules. *Publ. Choice* **179**(1–2), 113–124 (2019). <https://doi.org/10.1007/s11127-018-0524-6>
17. Herings, P.J.-J., Houba, H.: The Condorcet paradox revisited. *Soc. Choice Welfare* **47**(1), 141–186 (2016). <https://doi.org/10.1007/s00355-016-0950-7>
18. Aitamurto, T., Landemore, H., Galli, J.S.: Unmasking the crowd: participants’ motivation factors, expectations, and profile in a crowdsourced law reform. *Inf. Commun. Soc.* **20**(8), 1239–1260 (2017)
19. Goel, A., Krishnaswamy, A.K., Sakshuwong, S., Aitamurto, T.: Knapsack voting for participatory budgeting. *ACM Trans. Econ. Comput. (TEAC)* **7**(2), 1–27 (2019)
20. Ghezzi, A., Gabelloni, D., Martini, A., Natalicchio, A.: Crowdsourcing: a review and suggestions for future research. *Int. J. Manage. Rev.* **20**(2), 343–363 (2017)
21. Isinkaye, F.O., Folajimi, Y.O., Ojokoh, B.A.: Recommendation systems: P, methods and evaluation. *Egyptian Inform. J.* **16**(3), 261–273 (2015)
22. Dodevska, Z., Kovacevic, A., Vukicevic, M., Delibasic, B.: Two sides of collective decision making - votes from crowd and knowledge from experts. In: *ICDSST 2020, EWG-DSS 6th International Conference on Decision Support System Technology (2020, in press)*
23. Takács, G., Tikk, D.: Alternating least squares for personalized ranking. In: *Proceedings of the Sixth ACM Conference on Recommender Systems (2012)*
24. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013)
25. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, October, pp. 1532–1543 (2014)
26. Lloyd, S.P.: Least squares quantization in PCM. *IEEE Trans. Inf. Theory* **28**(2), 129–137 (1982)
27. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: *2008 Eighth IEEE International Conference on Data Mining, 15 December 2008*, pp. 413–422. IEEE (2008)



Temporal Network Analytics for Fraud Detection in the Banking Sector

László Hajdu^{1,2,3}✉ and Miklós Krész^{1,2,3}

¹ Innorenew CoE, Livade 6, 6310 Izola, Slovenia
{laszlo.hajdu,miklos.kresz}@innorenew.eu

² University of Primorska, 6000 Koper, Slovenia

³ University of Szeged, Szeged 6720, Hungary

Abstract. A new methodology in temporal networks is presented for the use of fraud detection systems in the banking sector. Standard approaches of fraudulence monitoring mainly have the focus on the individual client data. Our approach will concentrate on the hidden data produced by the network of the transaction database. The methodology is based on a cycle detection method with the help of which important patterns can be identified as shown by the test on real data. Our solution is integrated into a financial fraud system of a bank; the experimental results are demonstrated by a real-world case study.

Keywords: Cycle detection · Fraud detection · Transaction network

1 Introduction

In the banking sector the financial institutions developed and digitized their infrastructure and services over the decades. Presently a dynamically increasing number of tools are available for the customers to improve the service level. From the side of the clients this evolution offers a comfortable and safe control over their financial tasks. However, as a by-product, through the established system, nowadays financial institutions are able to collect data about our financial activities. The produced data of the client activities contain important information about the use of the actual financial system, and can be useful in many different ways [3, 7]. With the help of the extracted information, the institutions can offer personalized products, improve the quality and traceability of the whole system, maximize their profit or even detect and prevent the illegal fraud activities of

The Authors gratefully acknowledge the European Commission for funding the InnoRenew CoE project (Grant Agreement 739574) under the Horizon2020 Widespread-Teaming program, and the Republic of Slovenia (Investment funding of the Republic of Slovenia and the European Union of the European Regional Development Fund).

L. Hajdu—Supported by the project "Integrated program for training new generation of scientists in the elds of computer science", no EFOP-3.6.3- VEKOP-16-2017-0002 (supported by the European Union and co-funded by the European Social Fund).

© Springer Nature Switzerland AG 2020

L. Bellatreche et al. (Eds.): ADBIS/TPDL/EDA 2020 Workshops and Doctoral Consortium, CCIS 1260, pp. 145–157, 2020.

https://doi.org/10.1007/978-3-030-55814-7_12

individuals or companies. From the viewpoint of the financial institutions the key challenge is the extraction of the suitable and required information from the activity data.

In this research we define a special case of the banking fraud, the fraud through cycle in the transaction network, and introduce a method which is able to detect the suspected fraud activity in this network. Different fraud techniques can be connected with the cycle transactions such as if a group of companies makes financial transactions along a cycle through false orders and for each transaction they book false expenses to avoid the taxation and “launder the money”. The other typical fraud with this method is, if a company or an individual needs to prove that having enough fund or spare to get a loan. Nevertheless, in banking sector there are many other versions of the frauds through the cycle transactions.

The main motivation behind this research was a real world request from an anonymous bank which integrated a fraud module in its information system and as a key element of this module it was requested to detect transaction cycles to identify strange usage of the system. The developed methodology integrates the temporal network topology analysis with the relevant connected data, which is the amount of the transfer in this case. The above characteristics of this approach can set the basis of new solutions for future use of temporal network queries in graph databases. The research methodology follows the guidelines for conducting and evaluating design-science research [19,20].

The structure of the paper is the following. After the Introduction first we give an overview of related work, then we will present the concept of fraud detection with cycle search. In Sect. 4 the main contribution is described by giving the formal problem definition and the detailed summary of the methodology. In Sect. 5 the use case is presented on real data provided by a Hungarian bank. Finally we close the paper with a Summary.

2 Related Work

Data analytics based fraud detection solutions on digital data have been a relatively long history. Banks together with telecommunication and insurance companies were among the first in the use of statistical based methods in industry [1]. However, it turned out that even with significant amount of data (which was already available decades ago in the above sectors) the statistical inference with fraudulent activities causes one of the main challenges: there is a high number of legitimate records for each fraudulent one. Therefore AI based anomaly detection methods have become a central approach in the field [18].

Business information systems containing fraud detection module need to consider different layers of customer actions. According to Gartner [9] five layers can be distinguished (see Fig. 1). As it can be seen, for the detailed analysis an in-depth network structure of the problem (Layer 5) is required. This structure reflects the consequence of the above mentioned statistical inference: using individual data analysis has a limitation for identifying fraudulent patterns.

Motivated by these constraints and the explosion of network analysis in the last 20 years, recently a few graph based solutions have been developed for fraud

detection: see e.g. [5,11,13]. However, these solutions were applied on online networks of data streams for identifying new simple patterns and did not consider the temporal dimension.

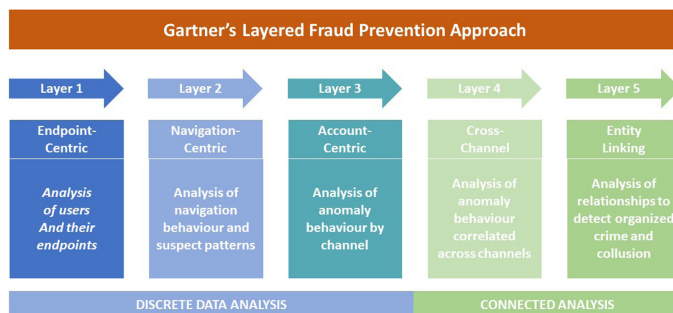


Fig. 1. The 5 layers of fraud detection, based on Gartner [9]

The significance of network based methods is also reflected by the evolving development of graph databases, which need sophisticated methods for structural graph queries [2]. The intensive business use of these platforms naturally raises the question with respect to their applicability for fraud analysis [15]. However, standard solutions are not available for two reasons. On one hand the network based methodology is not widely adapted yet; on the other hand the time dimension has a central issue in fraud detection, which is handled on data level only, but not with an integrated manner in network topology. Apart from a preliminary case study [4], no specific approach has arisen in this direction.

The basic barrier for the technological development discussed in the previous paragraph is the sporadic results on efficient algorithmic methods towards graph structure queries in temporal networks. Nevertheless, in recent years a few efficient algorithmic solutions were developed for networks where edge labels have time stamps. The authors of [12] worked out a general methodology for identifying different graph motifs. However, this approach is too general for specific questions like cycle detection. The approach of [8] is targeting related specific problems, but their method is identifying simple cycles (no repetition in nodes) only and the related data (attributes of the nodes and edges of the network) are not considered in the potential queries.

Concerning the classical methods for cycle detection and elementary cycle (simple cycle without internal node repetition) detection in static networks, two different approaches were investigated. The general cycle detection [17] is mostly based on a depth-first-search, and a cycle is found from a given node if the DFS [16] finds a back edge to the ancestor node. The main problem with this approach is, it concentrates on the nodes and not on the edges, but from our point of view the edges contain the important information about the financial transactions. The other approach [6] is the elementary cycle detection, which

lists all of the cycles in the network. The above methodologies are also extended for the analysis of large graphs in a distributed manner [14]. Even though these approaches principally could be used for the fraud detection, because in the elementary cycle problem the main emphasis is on the edges, but they cannot be applied directly as in our problem the timestamps and the specific attributes (e.g. transferred amounts) are also important.

In summary, relevant results were published on AI based fraud detection, graph databases, temporal network pattern recognition and static graph cycle detection, but an integrated approach is not available yet.

3 Network Based Fraud Analytics in the Financial Sector

In most cases the financial institutions have the proper data about the client activities and financial transactions, but they are not able to extract the information needed to identify the suspected fraudsters. In a “money laundering” the fraudsters try to hide the illegal source of their income and to transform into legal one. Financial institutions are interested in “fighting against” this situation, because the reputation is highly important in this area: if a fraud is revealed publicly, it can cause a huge loss indirectly as well. The inflow of illegal financial source into the legal economy can distort the balance of financial markets, thus identifying frauds is a common interest of the banking sector. In the real life the data mining based fraud detection methods are very important tools to identify the fraudsters based on the produced data, thus extracting and detecting the suspected patterns in the financial system is a key task. A good overview about the financial frauds and a classification of the methods can be found in [10] and [18]. It can be stated that most of the financial fraud detection systems are based on machine learning methods which can be powerful tools to classify the clients into different groups with respect to “fraud-like” activities. Our detection method can be a part of this type of systems and can explore the frequency of clients in transaction cycles.

From the viewpoint of the government detecting frauds in the banking system is also very important: most of the tax evasions and fictitious companies are detectable through the banking data. A fictitious company is a unreal company which can be identifiable if it has the following properties. The headquarter is in an address where hundreds or thousands of companies are registered, the owner has unrealistically many companies, and the company is in connection with transaction cycles where tax evasions and cycle billings can happen. In our research the transaction cycles are important from this viewpoint also. These types of frauds are in connection with transfer cycles very often; the detection of the cycles can support the institutions and with the extracted information it is possible the detect the fictitious companies.

The transactions are executed in a “continuous manner” with respect to time dimension and not every transfer is important from the viewpoint of the bank. Since the frauds happen in a way that through the detected cycle the amounts on the edges can differ from each other, but the amounts are close to

each other-, so there is a gap between transfers, it possible that a cycle is a valid suspicious activity from one node, but invalid cycle from another. In summary, it can be stated that the fight against the fraud in the banking sector is really important and can be a big challenge. Therefore it is very important to develop new methodologies to improve the efficiency of detection systems and identify the suspected fraudsters in an integrated manner of temporal network topology and data.

4 Problem Formulation and Solution Methodology

It can be seen by the previous section that monitoring fraudulent activities in financial networks of banking systems is a crucial business question in everyday operation. We have shown that cycle detection in temporal network data can play a key role in this process. In this section we will formalize the investigated problem and will provide the detailed description of our new solution methodology.

4.1 Financial Transaction Network

The financial transaction network has two different aspects. The first one is the transaction itself in which there is a transaction data where each transfer is recorded. It means that for every transfer we have a “from” and “to” account, an amount, and a date when the transfer realized. As the other aspect we have a data table about the clients which contains key-value pairs where the key is the client and the values are the corresponding bank accounts to the actual client. One bank account can belong to one client (individual or company) and one client can have multiple accounts.

First let B be the set of the bank accounts where each $b \in B$ is an account identified by the IBAN number. To define the first network let $G(B, E)$ be a directed network where B is the set of the bank accounts, so every node is an account in the bank and E is the set of every transaction among the bank accounts. The transactions have different attributes, consequently let t_e denote the timestamp and a_e be the amount of the edge e , where $e \in E$ and e is a transaction which realized in time t_e and the transferred amount is a_e through the transaction. An example of the network G is shown on Fig. 2. a).

Since a client can have multiple account in the bank, and in the fraud detection the cycles are interesting between the clients and not just between the bank accounts, let C be the set of the clients. Let define a new network which is a directed hypergraph $H = (P, E)$ where every $p_i \in P$ is a set for all $i \in C$ so p_i is the set of the client i so it contains every bank account which corresponds to the actual client, and E is the set of the edges from the network G . The input of our algorithm is the hypergraph H , so the nodes are the clients where every node contains each bank account of the actual client, and the edges are the original transactions between the accounts.

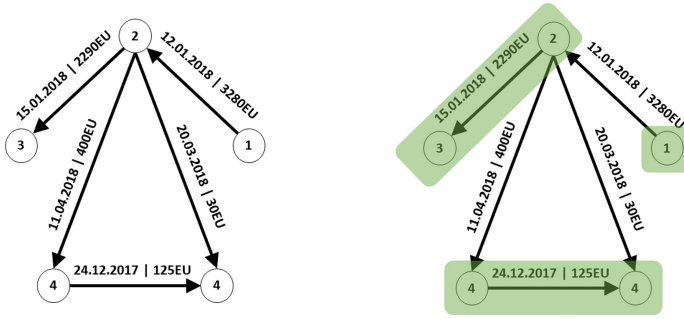


Fig. 2. An example of the money transfer network. a) The $G = (B, E)$ network so the transfers between the simple bank accounts. b) The hypergraph $H = (P, E)$ where the nodes are the clients and one node can contain multiple bank accounts from the original network. The red rectangles indicate the clients in the H network.

4.2 Cycle Detection

To introduce what we can consider as a cycle from our point of view, we define the basic cycle definitions and expand it to the transaction cycles in banking sector. Based on a repetition of nodes or edges multiple different cycle definition can be distinguished. In the general method a cycle is a path through the nodes wherein a node is reachable from itself. In this simple case, an edge and a node (except the starting node) can not be repeated in one cycle so the nodes and the edges are unique along the cycle. An another type of the cycles is the circuit, where in a cycle only nodes can be repeated so it is a closed walk which also can be a simple cycle.

To define transaction cycles, we will apply temporal data networks, where both the edges are attributed with time stamps and transaction data. We also note that the following definition of transaction cycles can be generalized in a natural way both on the time dimension and the related data records: different constraints can be considered for identifying an edge to be “living” from the point of view of a given cycle. For example we can make restrictions by limiting the (relative) difference of the transaction amounts or the time difference of two consecutive transactions. The developed method can be also easily modified accordingly, we are using these constraints for demonstration as they were defined by experts of the bank of the use case.

Therefore, let a transaction cycle be a closed walk where:

- The repetition of the nodes along a cycle is allowed while for the edges is not allowed
- Let $t_e^0, t_e^1 \dots t_e^{n-1}, t_e^n$ be the sequence of the timestamps along a valid cycle where every $t_e^i \leq t_e^{i+1}, i = 0 \dots n - 1$ so the timestamps are in ascending order.
- Let a_e^0 be the amount of the first transaction and a_e^i be the amount of the further transactions with $i = 1 \dots n$. Then $a_e^0 \cdot (1 - \alpha) \leq a_e^i \leq a_e^0 \cdot (1 + \alpha)$,

meaning that the difference between the amount of the first transaction and other transactions is bounded by the previously given real value $0 \leq \alpha \leq 1$. The α value is a parameter of the methodology, it is defined by the specific user requirements.

It is easy to see, neither general detection methods nor the elementary cycle detection is able to consider all of the requirements: in general methods the repetition of the nodes is not allowed, while the elementary cycle detection method is not able to consider the time and amount constraints. In the general cycle detection the input is a network, in our case it is the hypergraph $H = (P, R)$, and the output is the set of the cycles in the network. Table 1. shows an example input file with different transaction data.

Table 1. Example produced input file where the clients are already connected to their bank accounts.

ClientFromID	ClientToID	IBANFrom	IBANTo	Date	Amount
3	1	IBAN3-1	IBAN1-1	26.09.2018	250 Euro
1	2	IBAN1-1	IBAN2-1	03.09.2018	255 Euro
2	3	IBAN2-1	IBAN3-2	20.09.2018	245 Euro
3	4	IBAN3-1	IBAN4-1	19.09.2018	390 Euro
4	5	IBAN4-2	IBAN5-1	17.09.2018	1000 Euro
5	2	IBAN5-1	IBAN2-2	01.09.2018	768 Euro

The presented input format contains the information about the clients such as the bank accounts, so every client have a ClientID which encapsulates all of the bank accounts of the client. The data represents 6 different transactions between 5 different clients, where the clients with ClientID 2, 3, 4 have two different bank accounts. In the defined graph every clientID will be a node, and every line in the input file will be an edge with the actual properties between the clients. The corresponding graph H , where the nodes are the clients and the edges are transactions can be seen in Fig. 3.

4.3 Special Cases

It is important to detect all cases which fit into the definition because if a cycle remain undetected and the method is not able to recognize a special case, the fraudsters can use this to hide their frauds. The main challenge in our research was to understand the nature of the transaction cycles and define the special and possible cases of the structure. Since the order of the nodes along a cycle is restricted, every cycle can be a valid cycle only from one starting node, if the amounts are within the specified limits. In real life any type of money movements can be a cycle from the point of view of the financial institute, even multiple back and forth transactions between only two clients. Since our algorithm based on

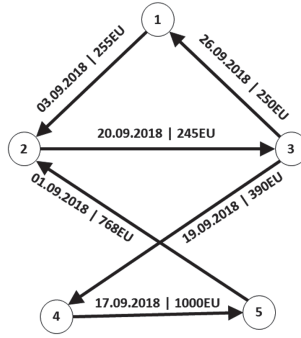


Fig. 3. An example of a graph which contains transactions cycle. This figure shows the transactions of 5 different clients. Based on the cycle definition, along the 1-2-3-1 closed walk the timestamps are in ascending order and it is a cycle if the $\alpha \geq 0.1$ nevertheless for example the 5-2-3-4-5 is not because of timestamps and amounts.

DFS, the special and interesting cases are the ones where two or more cycles are part of each other along transactions or when a client provides loan for multiple clients. It means that cycles can be part of each other along transactions or nodes, or a node can be repeated through a cycle. Figure 4 shows two different examples for the above mentioned special cases.

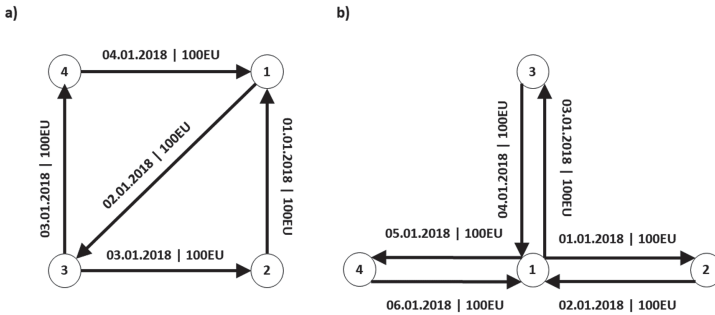


Fig. 4. Two special cases of the cycle definition. Figure a) shows two cycles (1-3-4-1 and 2-1-3-2) which are part of each other. The starting transaction of the first cycle is 1-3 which realized on 02.01.2018 while of the second cycle is 2-1 on 01.01.2018. This structure can occur with arbitrary number of cycles in any rotations. Figure b) shows the 1-2-1-3-1-4-1 cycle which represents back and forth transactions from client 1 to the others. Naturally any combinations of the special cases can occur in the network.

From theoretical point of view it is important to deal with these cases even if real life these are not so common: if a possible fraudster knows the detection method, he can take an advantage and make a fraud in the way what the method can not detect. Since it is a structural pattern and our method can detect each case based on the definition, the method is undeceivable.

4.4 Method

In most cases bank data partially unstructured (depending on the actual bank); hence as a first step the framework cleans the data and transforms the IBAN numbers into common format in the transaction table and in the client-bank account data. After cleaning the client nodes are produced based on the key value pairs and the hypergraph is built (“hypergraph” refers in this case the fact that all bank accounts of the same client is combined into one node). As Algorithm 1 shows the input of the method is the hypergraph $H(B, E)$ and the edge list ordered by the date stamps. The visiting technique sets every edge to “actually” visited in every root call, while the first transaction of the cycle will be “finally” visited: it means that every cycle from the actual ancestor transaction will be identified by the first edge. After every root call the algorithm sets the actually visited edges to “nvisited” but the finally visited edges will not be visited again by the algorithm. To ensure the correctness of the algorithm, ordering is needed as the DFS will be called in sequence for every edge based on the ordered edge list. The reason for the above solution is justified with the following example: in Fig. 4. a) case the algorithm finds the 1-3-4-1 cycle first, then the 2-1-3-2 remains undetected since the state of the 1-3 edge will be “finally” visited. Nevertheless, if we call the recursive function based on the ordered edge list, and the algorithm searches the cycles by date, every cycle can be detected. Algorithm 1 shows the pseudo-code of the detection method.

In the pseudo-code the inputs are the following: $H(B, E)$ is the hypergraph and O denotes the ordered edge list by date stamps. First the method initializes set C , which is the set of the detected cycles, v_0 as the ancestor node, e_0 which is the first transaction, and $depth$ for the recursion depth. The recursion depth is recorded in every call and it denotes the length of the actual walk in the network. From line 9–22 the recursive function is presented with a node, edge, α , $maxlength$ parameters, where n is the actual node wherein the recursion is called, e_0 is the first transaction, α is the bound for the difference of amounts, and the $maxlength$ is a limit for the maximal $maxlength$ line 11, a cycle is found if the actual node v is equal to the node v_0 (starting node), the depth is between 1 and the parameter $maxlength$, as well as the length of the cycle is maximal so it cannot be extended with additional transactions. The $out_edges(v)$ in line 17 means the out edges of node v : if the recursion depth is greater than one, the algorithm searches the cycles through every unvisited neighbours where the date stamp and the amount of the transaction meets the conditions. In other words, the recursion searches in a direction where the amount and the time parameters are correct. The kernel of the method is the calling order which can be seen in line 28, where DFS is called for every ordered edge in the list O . The output of the algorithm is the set C of detected cycles. The complexity of the algorithm in one iteration is $O(|B| + |E|)$ and in overall case it depends on the number of the cycles in the network. Nevertheless, in the case of real life transaction networks, parameters α and $maxlength$ limit the depth of the recursion with providing a reasonable running time. The advantage of the algorithm is that it is able to handle different properties in connection with the transactions, like

Algorithm 1 Cycle detection Method

```

1: Input:
2:  $H = (B, E)$ 
3:  $O$  (ordered edge list by timestamps)
4: Method:
5:  $C \leftarrow \emptyset$ 
6:  $v_0 \leftarrow \emptyset$ 
7:  $e_0 \leftarrow \emptyset$ 
8:  $depth \leftarrow 0$ 
9: function CDM( $v, e_0, \alpha, maxlen$ )
10:    $depth ++$ 
11:   If:  $c$  cycle is found so  $v == v_0$  where  $depth > 1$  and  $depth < maxlen$ 
12:      $C \leftarrow c$ 
13:     If:  $depth == 1$ 
14:        $e.visited \leftarrow visited_{actual}$ 
15:       CDM( $e_0.to(), e_0, \alpha, maxlen$ )
16:     Else:
17:       For:  $\forall e$  in  $out\_edges(v)$  where  $(abs(e_0.a_e - e.a_e) < \alpha \cdot e_0.a_e \ \& \ e_0.t_e < e.t_e)$ 

18:         If: ( $e.visited == unvisited$ )
19:            $e.visited \leftarrow visited_{actual}$ 
20:           CDM( $e.to(), e_0, \alpha, maxlen$ )
21:          $depth --$ 
22: End function
23: For  $\forall o$  in  $O$ 
24:    $v_0 \leftarrow o.from()$ 
25:    $e_0 \leftarrow o$ 
26:   CDM( $v_0, e_0, \alpha, maxlen$ )
27:    $e_0.visited \leftarrow visited_{final}$ 
28:   For  $\forall v$  in  $V$ 
29:     If ( $e.visited == visited_{actual}$ )
30:        $e.visited \leftarrow unvisited$ 
31: Output:  $C$  set of found cycles

```

date stamps, different amounts, or any restriction which can be formalized as a constraint before the next recursive call.

5 Case Study

Our methodology is general with respect to different parameters, however the demonstrated use as outlined earlier, is defined by a Hungarian bank introduced a new fraud system.

The fraud detection system monitors the activities of the clients and the administrators. In the system there are more than 50 indicators to indicate the suspected frauds and one of the key indicators is the transaction cycles using our method. The reason is that the method is potentially integrated with any attribute connected to the edges or the nodes, the methodology can be combined

with other filtering procedure defined by the indicators. During the method the constraints for edge selection can be evaluated dynamically, for example the bound for the amount can depend on the values of other attributes as well.

Transaction cycles (like generally fraudulent actions) are very rare, especially if the length of the actual cycle is longer than 3 and the amounts are closely similar. In this case study real transaction cycles identified by the system presented in a way that the IBAN numbers of the clients are masked, but the dates and the amounts are real. As a case study network we received transactions of a part of the year of 2016. The network contains 139242 bank accounts and 1296815 transactions. Table 2 shows the results on the real transaction network with different parameters. The bank preferred the $\alpha = 0.1$, $Min\ length = 3$, $Max\ length = 6$ parameters, because in real life the longer cycles could be random transaction cycles also.

Table 2. Results of the cycle detection algorithm on the real transaction network

Parameters	Number of cycles	Number of bank accounts	Number of clients
$\alpha = 10\%$ $Min\ length = 3$ $Max\ length = 6$	276	440	437
$\alpha = 20\%$ $Min\ length = 3$ $Max\ length = 6$	789	1026	997
$\alpha = 10\%$ $Min\ length = 3$ $Max\ length = 10$	277	449	447
$\alpha = 20\%$ $Min\ length = 3$ $Max\ length = 10$	835	1122	1087
$\alpha = 10\%$ $Min\ length = 3$ $Max\ length = 20$	277	449	447
$\alpha = 20\%$ $Min\ length = 3$ $Max\ length = 20$	839	1148	1111

We tested our algorithm with 2 different α values and 3 different maximal length values, thus with 6 different parameter combinations. The minimum length was 3 in each case, because in this case the back and forth transactions were excluded which are not important from the fraud point of view. As Table 2 shows, if we allow 10% difference between the amounts along the cycles, the length of the allowed transaction cycles doesn't have to much effect to the number of the founded cycles. The number of the corresponding bank accounts is

very low compared to the number of the bank accounts in the whole network. In this case the number of the clients is not higher than the number of the bank accounts which means that the clients were not using multiple bank accounts through the cycles. As the table shows in the second, fourth and sixth cases the α has big influence to the number of cycles and corresponding accounts. In these cases some clients were using multiple accounts, which could mean random cycle, or suspected fraud activities also. The running time of the algorithm was within 1.5–2 h in every cases, but in real life applications the running time is not so critical because mostly the bank runs the algorithm only for a shorter period of the year, even though the method is able to handle longer periods (even several years). The complete system has been working at the institute since 2017 and it detects 2–4 suspected fraud activities in every month and it also detected some fraud-related client base.

6 Summary

Fraud detection systems play central role in the life of financial institutions, but it is generally problematic to detect structural patterns in the data. The main challenge is to turn the real processes into the data level, understand the structure and develop a method which is able to detect all of the occurring pattern which are interesting for the financial institute. In this paper we introduced a DFS based solution for the cycle detection problem, which is able to detect the real life structures in the network and can handle time, amount and other parameters. The main part of the algorithm is the visiting technique and the ordered edge list based root calling sequence. It is a powerful tool to detect the different type of transaction cycles in the network, because it offers a wide parameterization in which the bank can set their preferences. The research was initiated by a Hungarian bank and the produced method has been used in the real life also since 2017. The methodology can be used also to detect cycles with additional preferences, where for example the time between the transactions is limited, or other time or amount based regulations are defined.



References

1. Bolton, R.J., Hand, D.J.: Statistical fraud detection, a review. *Stat. Sci.* **17**(3), 235–249 (2002)
2. Bonifati, A., Dumbrava, S., Queries, G.: From theory to practice. *SIGMOD Rec.* **47**, 5–16 (2019)
3. Bóta, A., Csernenszky, A., Györfy, L., Kovács, G., Krész, M., Pluhár, A.: Applications of the inverse infection problem on bank transaction networks. *CEJOR* **23**(2), 345–356 (2014). <https://doi.org/10.1007/s10100-014-0375-2>
4. Cattuto, C., Quagiotto, M., Panisson, A., Averbuch, A.: Time-varying social networks in a graph database: a Neo4j use case. In: *GRADES 2013* (2013)
5. Chau, D.H., Faloutsos, C.: Fraud detection using social network analysis: a case study. In: *Encyclopedia of Social Network Analysis and Mining*, 2nd (edn) (2018)

6. Johnson, D.B.: Finding all the elementary circuits of a directed graph. *SIAM J. Comput.* **4**, 77–84 (1975)
7. Krész, M., Pluhár, A.: Economic network analysis based on infection models. In: *Encyclopedia of Social Network Analysis and Mining*, 2nd (edn) (2018)
8. Kumar, R., Calders, T.: 2SCENT: an efficient algorithm to enumerate all simple temporal cycles. *Proc. VLDB Endow.* **11**, 1441–1453 (2018)
9. Litan, A.: *The Five Layers of Fraud Prevention and Using Them to Beat Malware*. Gartner, Stamford (2011)
10. Ngai, E.W.T., Hu, Y., Wong, Y.H., Chen, Y., Sun, X.: The application of data mining techniques in financial fraud detection: a classification framework and an academic review of literature. *Decis. Support Syst.* **50**(3), 559–569 (2011)
11. Pandit, S., Chau, D.H., Wang, S., Faloutsos, C.: Netprobe: a fast and scalable system for fraud detection in online auction networks. In: *WWW 2007* (2007)
12. Paranjape, A., Benson, A.R., Leskovec, J.: Motifs in temporal networks. In: *WSDM 2017* (2017)
13. Qiu, X., Cen, W., Qian, Z., Peng, Y., Zhang, Y., Lin, X., Zhou, J.: Real-time constrained cycle detection in large dynamic graphs. *Proc. VLDB Endow.* **11**, 1876–1888 (2018)
14. Rodrigo, C.R., Bhalchandra, D.T.: Distributed cycle detection in large-scale sparse graphs. In: *Simpósio Brasileiro de Pesquisa Operacional (SBPO)* (2015)
15. Sadowksi, G.G., Rathle, P.: *Fraud detection: discovering connections with graph databases*, Neo4j White Paper (2015)
16. Tarjan, R.: Depth-first search and linear graph algorithms. *SIAM J. Comput.* **1**(2), 146–160 (1972)
17. Tucker, A.: Covering circuits and graph colorings. In: *Applied Combinatorics*, 6th (edn), p. 49 (2006)
18. West, J., Bhattacharya, M.: Intelligent financial fraud detection: a comprehensive review. *Comput. Secur.* **57**, 47–66 (2016)
19. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. *MIS Q.* **28**(1), 75–105 (2004)
20. Peffers, K., Tuunanen, T., Rothenberger, M., Chatterjee, S.: A design science research methodology for information systems research. *J. Manag. Inf. Syst.* **24**(3), 45–77 (2007)



Abdominal Aortic Aneurysm Segmentation from Contrast-Enhanced Computed Tomography Angiography Using Deep Convolutional Networks

Tomasz Dziubich¹(✉) , Paweł Białas¹, Łukasz Znaniński² , Joanna Halman², and Jakub Brzeziński²

¹ Computer Vision & Artificial Intelligence Laboratory,
Department of Computer Architecture, Faculty of Electronics,
Telecommunications and Informatics,
Gdańsk University of Technology, Gdańsk, Poland
tomasz.dziubich@pg.edu.pl

² Cardiac and Vascular Surgery,
Medical University of Gdańsk, Gdańsk, Poland
lznaniński@uck.pl
<http://cvlab.eti.pg.gda.pl/>

Abstract. One of the most common imaging methods for diagnosing an abdominal aortic aneurysm, and an endoleak detection is computed tomography angiography. In this paper, we address the problem of aorta and thrombus semantic segmentation, what is a mandatory step to estimate aortic aneurysm diameter. Three end-to-end convolutional neural networks were trained and evaluated. Finally, we proposed an ensemble of deep neural networks with underlying U-Net, ResNet, and VNet frameworks. Our results show that we are able to outperform state-of-the-art methods by 3% on the Dice metric without any additional post-processing steps.

Keywords: Abdominal aortic aneurysm · Deep learning · Image segmentation · Computed tomography

1 Introduction

The segmentation of three-dimensional medical images is a process based on identifying the desired areas of the medical image by assigning each image pixel or voxel to a specific category/class and enabling volumetric-quantitative analysis. The segmentation accuracy depends on used methods, amongst which deep convolutional neural networks are currently the most popular, and are able to identify the boundaries of the entire object or its interior. The automatic computer segmentation makes the process of determining given structures much

faster and does not require human participation. Considering the latest trends in machine learning, some automatic methods can achieve precision similar or even exceeding human precision. Xie et al. [13], presented ImageNet model that can achieve an accuracy of 98.7% (top-5) which is comparable to the so-called Noisy Student’s accuracy (in fact 2.9% higher).

The automatization of segmentation is crucial. In one medical case, there are up to 1000 two-dimensional images. It requires a lot of time, for every case to be segmented manually. For example, full abdominal aortic aneurysm segmentation takes up to half an hour. A successful automatic segmentation can then help build a 3D model of an organ by stacking segmented two-dimensional images and combining them into one to provide mechanisms that accelerate specific calculations for diagnosis. Such a three-dimensional model or bio-parameters will help medical professionals to recognize the pathological condition of an organ.

In this paper, we focus on Abdominal Aortic Aneurysm (AAA) segmentation. AAA is a permanent pathological dilatation of the abdominal aorta. The disease affects mainly the elderly males (4-fold increase in incidence in comparison to females). The prevalence over age 65 is 4–7%. The disease is asymptomatic in the majority of cases. Unfortunately, if undiagnosed and untreated, it leads to aneurysm rupture and massive internal bleeding. When ruptured, the mortality rate is 70–80%.

In the majority of cases, the surgery is prophylactic, performed to prevent the rupture. The risk of rupture rises with aortic diameter and exceeds the risk of surgery when diameter reaches >55 mm. The gold diagnostic standard is contrast-enhanced Computed Tomography Angiography (CTA). The size of AAA is measured by a time-consuming manual measurement, which is prone to error. Intra-observer variability is within an acceptable level of ± 5 mm in over 90% of observations, however, inter-observer variability exceeds an acceptable level of ± 5 mm in 87% cases, leading to false reporting. This can result in delays with a referral to vascular specialists and delayed access to surgery in patients with understated AAA diameters on one hand, and a high burden of false-positive cases on the other hand.

Many AAAs are diagnosed incidentally, in abdominal CT scans performed for other indications. However, only 65% of AAAs that show incidentally on the CT scans are being actually diagnosed [2]. Omitting AAA on the scan can lead to delayed treatment and allow the disease to become life-threatening.

The treatment can be either surgical or endovascular. Endovascular Aortic Repair (EVAR) is based on endovascular exclusion of the aneurysm from circulation with the use of aortic endoprosthesis (stent-graft). One of the delayed complications after EVAR is an endoleak, defined as blood flow between aneurysm wall and endoprosthesis. It has been established as a marker for a poor outcome. Endoleak leads to AAA sac enlargement, and in 20% of cases needs further invasive treatment. It is diagnosed and monitored by CTA, with AAA sac measurements critical for timely decisions to intervene. The process of manual endoleak detection is time-consuming, with high inter-observer variability as well.

There is a pressing need to develop a system to augment the human ability to diagnose, monitor, and follow-up AAAs after interventions, in order to increase

the diagnostic accuracy and to decrease the work-load burden of automatic tasks on health-care workers.

We propose to resolve the major problem outlined above by designing a novel neural network architecture. Here we introduce a deep learning based framework, called AAANet, to segment AAAs using a pipeline of three networks, all trained end-to-end. The pipeline receives whole-volume CTA images as input and outputs the segmented mask of AAA. Our method requires minimal pre-processing and utilizes features from all plans of CTA scan (i.e. axial, coronal, and sagittal) to segment anatomical region. More specifically, our major contributions include the following:

- building a data set with 30 labeled CTA scans with AAA;
- accuracy evaluation of various two- and three-dimensional models, i.e. ResNet, 3D-Unet and VBNet;
- applying an ensemble to improve segmentation precision.

The remainder of the paper is organized as follows. Related work is characterized in Sect. 2. The proposed pipeline architecture is described in Sect. 3. In Sect. 4 Description of our experimental setup and in Sect. 5 report and discussion about accuracy of the results results for our solution, measured on test set consisting of 5 images. Summary of the paper and our propositions of future work directions in Sect. 6.

2 Related Works

A manual assessment of an aneurysm and endoleak in CTA is quite difficult to achieve due to high variability in spatial location and small size of some structures (like thrombus and stent-graft) and therefore requires long-standing clinical experience [6]. Many different solutions have been proposed for automatic and semi-automatic detection and segmentation of the various anatomical structures connected with an abdominal artery and aneurysm. Many of these previous aids used classical computer vision techniques that required prior knowledge, such as external seed points for initialization [3, 11, 14].

One of the most interesting paper using classical techniques has been published by Lareyre, F. et al. [8]. The author’s main goals are to asses the aneurysmal localization in the aorta, the distance to the renal and iliac arteries, the presence of calcification and intraluminal thrombus. The proposed pipeline is based on image pre-processing (Gaussian, median and bilateral filters and fast non-local means denoise, intensity computation), segmentation of the aortic lumen (boundary propagation and active contour method), segmentation of the aortic thrombus (a morphological snake ACWE), and segmentation of the aortic calcifications (morphological operators and thresholding). The overall evaluated segmentation error is at $93\% \pm 4\%$ (the mean Dice coefficient) and for the segmentation of the thrombus – $88\% \pm 12\%$.

With the recent increased popularity of deep learning, new machine learning methods have been developed. The very first work was conducted by Lopez-Linares [10] who tried to use DCNN (DetectNet architecture) to achieve fully

automatic detection and segmentation of AAA thrombus. This is a key issue in assessing the presence and volume of an aneurysm. The used data set consisted of 13 post-operative CTA scans. The authors pointed out the lack of other ground truth databases. They achieved a mean Dice similarity coefficient of $82\% \pm 7\%$.

Bai et al. [1] proposed a method that combined a recurrent neural network with a fully convolutional network (FCN) for aortic image sequence segmentation. They were able to incorporate both temporal and spatial information. The method gained an average Dice metric of 96% for the ascending aorta and 95% for the descending aorta. The data set was acquired from the UK Biobank and consisted of 500 subjects on an aortic MR image set. Unfortunately, MR imaging is not the standard diagnostic procedure in AAA detection.

Lu et al. [9] presented the first algorithm for AAA segmentation which used three-dimensional segmentation based on a variant of 3D UNet architecture combined with ellipse fitting (an average Dice coefficient of 91% for segmentation, sensitivity 91%, and specificity 95% for detection). It is noteworthy that the solution can work on both contrast and non-contrast CT series. For training and testing 321 CTA scans which involved 153 scans with contrast were used. In [4] authors on the base above mentioned Deep AAA algorithm had tried to detect endoleaks and estimate measurements of aneurysm diameter, area, and volume. They proposed multi-class detector (AAA, stent-graft, other) that consisting of three stages:

- the detection a smaller region containing the aneurysm (2D bounding box) by using RetinaNet;
- the localization and binary endoleak detection by using ResNet-50 CNN (on 2D slices);
- finally, three-dimensional multi-class segmentation by using 3D UNet CNN.

The accuracy of AAA segmentation was $91\% \pm 5\%$ (Dice coefficient) and $95\% \pm 3\%$ for endografts. 334 scans were used for classifier’s training but only 33 were completed and 68 were partially completed for segmenting AAA and endograft.

3 Proposed Algorithm

In this part, we present the description of our deep learning model, which allows delineating aneurysm on the abdominal CTA image. Our model receives whole-volume CTA images as an input and outputs the 3D binary masks of all AAAs. The dimensions of a typical abdominal CTA are about $590 \times 512 \times 512$, but they can vary among patients, due to factors like image cropping and different settings.

Our proposed approach has three main components: (1) a ResNet-based network for detection and localization of AAA; (2) a group of 2D and 3D base-learners that are trained to explore the training data from different geometric perspectives; (3) an ensemble learning framework that uses a deep learning based bagging technique to combine the results from the base learners. A schematic overview of our proposed framework is shown in Fig. 1. The first step is a trans-

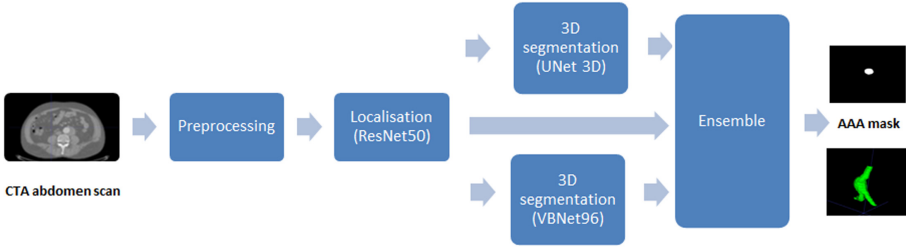


Fig. 1. Proposed model for AAA segmentation.

formation of a source image using adaptive distance normalization and setting a tissue window. The tissue window is an intensity band-pass filter, which only keeps the intensities within the band and censors the intensities beyond the maximal/minimal values. We set the window level with a level of 45 HU and a width of 710 HU (it covers soft tissues on contrast CT, blood, and thrombus). Outcomes are converted to grey-scale png-format files. Intensity values between are linearly normalized into the range $[-1,1]$. Intensities smaller than the minimum are set to -1 and those greater than the maximum are set to $+1$.

During the second step, the aorta is being detected by specifically trained ResNet50 network. This allows to limit the volume for analysis and thus increases the performance of three-dimensional networks 3D Unet and VNet96, which are the next stage in the pipeline. An overlapped sliding windows strategy and the average of the probability maps were used to crop sub-volumes and to get the whole volume prediction. The sub-volume size was also $96 \times 96 \times 96$ and the stride was $10 \times 10 \times 10$.

The last phase is the generalization of the results. We have proposed the ensemble technique that uses multiple learning algorithms to obtain better predictive performance. It is especially useful in random-like approaches like machine learning. It does that by reducing the possibility of overfitting. For the purpose of this paper, we used the bagging technique. Bagging is a method which involves training multiple learning algorithms and then combining their predictions by introducing a layer which average their predictions. In segmentation tasks, it comes down to weighted majority voting. We tested that approach with various levels of threshold (TH).

4 Method and Materials

4.1 Data Set

Image data consists of 30 contrast CT scans of the abdomen and pelvis performed between January 2015 and December 2019 by Medical University of Gdańsk Department of Radiology, which were acquired from 30 patients with previously diagnosed abdominal aortic aneurysms (AAA). The investigators obtained local

Institutional Review Board approval for the project. The mean age of in our study group was 70,6 years. As for the patients, there were 23 males and 7 females.

The team responsible for segmentation of the aorta consisted of three medical professionals - one vascular surgery specialist with expertise in the field of open and endovascular aortic aneurysms repair, and two residents in vascular surgery. All segmentations were cross-checked between the members of the team, and in questionable cases, the expert surgeon segmentation was considered ground truth. CTA examinations were fully segmented with binary masks representing AAA and artery area. Segmentation was carried out using ITK-SNAP version 3.8.0 running on OSX 10.15 platform. During the first step, automatic segmentation of contrast was carried out, using active contour mode from the level of distal thoracic aorta down to the level of common femoral arteries. Automatic, active contour-based segmentation of proximal parts of main aortic branches (celiac trunk, superior mesenteric, and renal arteries) as well as proximal parts of internal iliac arteries were also executed during this step. Parameters of the active contour mode were set as a result of series of experiments, in order to let the mask contain not only contrast, but leak just to outline the outer wall of the healthy aortic/iliac segment. In the second step manual segmentation of AAA, from the level of renal arteries to the level of common iliac arteries bifurcation was performed, manually outlining the outer wall of the aorta at each slice, and was saved as a separate label. Labels were subsequently combined together using Python script, giving the manual segmentation a superiority in determining aortic outline at the level of AAA and common iliac arteries over automatic segmentation. Scans consisted of 450–1300 slices, with the necessity of manual segmentation in about 40–60% of slices. Imaging, as well as ground truth labels, were provided in an anonymized NIFTI format.

4.2 Selected Metrics

There are several metrics for evaluating segmentation and localization of the medical image. Each of these metrics is based on a statistical interpretation of the prediction in regard to the ground truth. In [12], the authors have presented 20 metrics from different categories. We decided to choose Dice coefficient for the segmentation problem that is defined in Eq. (1).

$$Dice = \frac{2TP}{2TP + FP + FN} \quad (1)$$

where: TP – true positives, FP – false positives, FN - false negatives.

To evaluate the localization performance of the employed method, the Euclidean distances from the automatically determined bounding boxes to the reference bounding boxes were determined. We also involved mean average-precision metric (*mAP*). The threshold value t in this metric is selected depending on the problem and the expected accuracy of the detector. Values from 0.5 to 0.95 are typically used to evaluate the results on the COCO set for the 2D detection problem. Due to much larger searched space of a potential cuboid than a

two-dimensional rectangle, it is justified to use lower threshold values, especially in the case of medical images, where even the inaccurate location of lesions is often sufficient (on detection stage). For the problem of tumor detection on CT scans, Jaeger et al. [5] used the threshold value t equals to 0.1. This threshold value was also adopted in our work.

4.3 Training

The whole data set is split into a training set (16 scans and 8126 slices), a validation set (4 scans and 2279 slices) and a test set (10 scans). All networks are trained with Adam optimizer [7] with adapting a learning rate. We use the binary cross-entropy loss and early stopping mechanism to avoid over-fitting. Data augmentation was used to functionally increase the size of data sets fed into machine learning models via spatial or contrast-based data transformations. We apply randomly flipping and shifting in vertical and horizontal directions and small rotations as data augmentation. It is worth noting that merely in the case of 2d network training, online augmentation was used due to insufficient resources. The pipeline was implemented using Keras and Tensorflow as a backend. Training took approximately 7 h for ResNet and 14 h for 3D models on a Nvidia GeForce 2080TI/Intel Core i9/64 GB RAM/1 TB NVMe. At test time, it took 10 s to segment CTA image sequence.

5 Results and Discussion

5.1 Localisation Accuracy

We conducted quantitative evaluation on a test set consisting of 10 CTA scans. The mean and standard deviation of Euclidean distances between the automatically obtained and reference bounding boxes for all scans in the test set equals $11.24 \text{ mm} \pm 15.26 \text{ mm}$. The highest mAP achieved on the test data set using the Euclidean distance was 40.4% ($t=0.1$). The achieved mAP values may seem low, but given the complexity of the problem and the size of the data set, the average precision achieved by our models above 0.4 is a satisfactory result. In comparison in the paper published by Jaeger et al. [5] for cancer detection on breast MRI mAP 35.8 was achieved.

5.2 Segmentation Accuracy

In order to provide best segmentation accuracy, we compared the weights of the ResNet50 with the pre-trained weights of a SegThor model as well as trained without pre-training. There was no significant difference between these two approaches (93.722% and 93.739% respectively, Dice coefficient). The segmentation accuracy of the proposed method with single-class Dice loss and the ensemble of multiple models were evaluated online using 10 testing CTA scans. Table 1 shows the Dice metrics of the three models and the ensemble.

Table 1. Accuracy of the tested models.

Model	ResNet50	3D-UNet	VBNet	Ensemble
Dice coefficient	93.73	92.74	91.62	94.00

Our results show that we are able to outperform state-of-the-art methods by 3% on the Dice metric without any additional post-processing steps (like in [9]). While it is difficult to make comparisons between the methods due to differences in data sets, the results of our method are comparable to those of the semi-automated and classical methods of [8]. The ensemble outperformed previous known fully automatic algorithms for AAA segmentation on CTA images. However our training was done only on contrast-enhanced CTA scans.

Ensemble predictions from 3D UNet, 3D VBNet, and 2D UNet with Resnet50 as the backbone gave the best result of 94% Dice. Table 2 presents the impact of each network on the overall performance. The most significant benefit comes from ResNet50, at the threshold $TH = 0.4$. Surprisingly 2D model performed better than 3D models, which was probably caused by differences in online augmentation. In 2D network during training, we were randomly rotating, scaling, flipping, and shifting data set. In 3D networks, we were only flipping and shifting data because other operations would have lasted too long if they had been used in online augmentation.

Table 2. Accuracy of the ensemble models.

Accuracy (Dice coefficient)								
Threshold	Model	Weights						
	ResNet50	0.34	0.4	0.5	0.6	0.2	0.2	0.2
	VBNet	0.33	0.3	0.25	0.2	0.5	0.3	0.4
	3D UNet	0.33	0.3	0.25	0.2	0.3	0.5	0.4
0.3		93.67	93.79	93.84	93.77	93.20	93.90	93.52
0.4		93.99	94.00	93.71	92.96	93.61	93.97	93.97
0.5		93.91	93.93	92.96	92.96	93.70	93.84	93.84
0.6		93.48	93.56	93.13	92.96	93.07	93.38	93.35
0.7		92.16	92.12	92.91	93.10	92.08	92.50	92.19

The three examples of the predictions of ensemble model, with ground truth for reference and results on augmented data set are presented in Fig. 2.

In Fig. 2a some outlines can be seen which comes from a superior mesenteric artery. In Fig. 2b the segmented area has an indented shape what indicates the need for smoothing filters or ellipse fitting. Nevertheless, the achieved accuracy is very high.

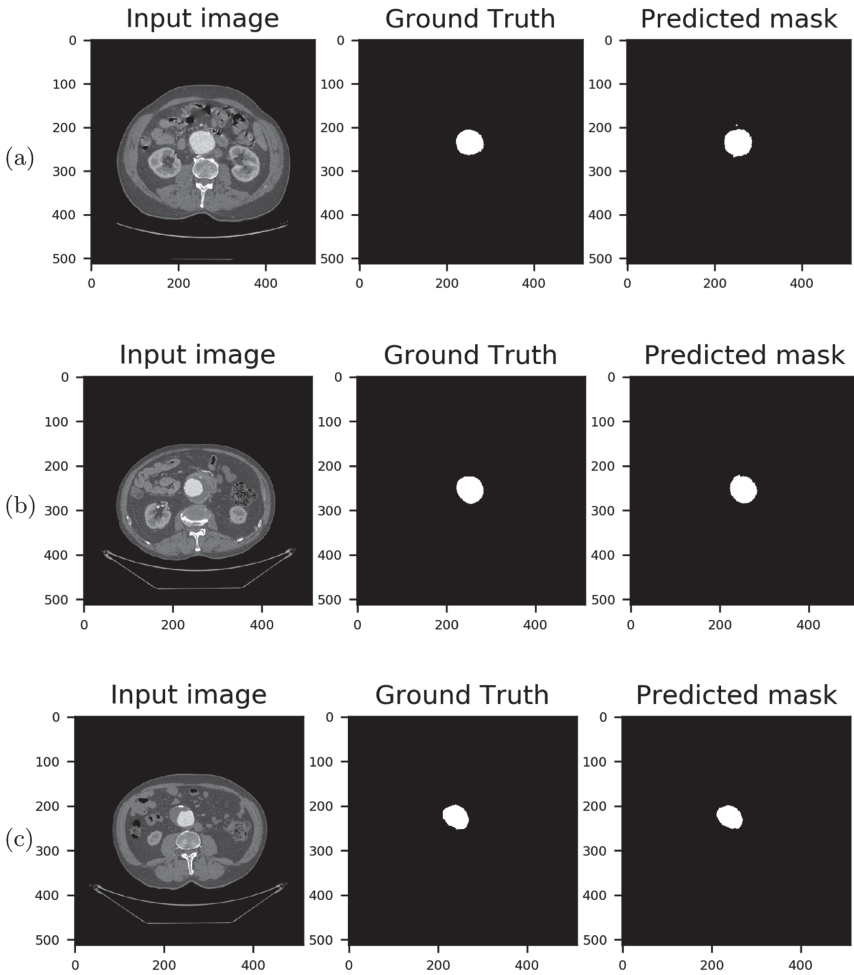


Fig. 2. Examples of visualization of the predictions performed for the ensemble model.

6 Summary

Automatic AAA segmentation is the solution of the future in the environment of constant shortage of medical workforce, with an increasing emphasis on productivity. Solutions, where automated tools suggest diagnosis to the physician, would help reduce time spent on tedious tasks, allowing the medical personnel to train and improve, and focus on issues where human participation is not to be replaced in the near future. In this paper we show that algorithmic AAA segmentation is possible with high accuracy. This is the first step on the pathway to automate diagnosis and treatment preparation as well as treatment result monitoring of arterial conditions. Properly trained algorithms may play a role in not only diagnosis and follow-up of AAA, but also in a surgical procedure

planning. Algorithmic solutions may help evaluate the early procedure success. Long term follow-up may be improved and simplified, with an automatic detection of late complications. A number of missed diagnoses may be diminished in cases where contrast-enhanced CT is performed for other than AAA indications. Proper data for algorithm training is key to success. Semi-automatic tools for anatomic structure labeling are already available. However, exact pixel-by-pixel delineation of anatomic structures is controversial in many cases, and requires interpretation with considerable inter-observer variability. Expertise is key here, but once algorithms get trained with high-quality data, this inconsistency in CT interpretation may be reduced, which would lead to decreased false-positive as well as false-negative results.

Acknowledgements. This work has been partially supported by Statutory Funds of Electronics, Telecommunications and Informatics Faculty, Gdansk University of Technology and grants from National Centre for Research and Development (Internet platform for data integration and collaboration of medical research teams for the stroke treatment centers, PBS2/A3/17/2013).

References

1. Bai, W., et al.: Recurrent neural networks for aortic image sequence segmentation with sparse annotations. In: Frangi, A.F., Schnabel, J.A., Davatzikos, C., Alberola-López, C., Fichtinger, G. (eds.) MICCAI 2018. LNCS, vol. 11073, pp. 586–594. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00937-3_67
2. Claridge, R., Arnold, S., Morrison, N., van Rij, A.M.: Measuring abdominal aortic diameters in routine abdominal computed tomography scans and implications for abdominal aortic aneurysm screening. *J. Vasc. Surg.* **65**(6), 1637–1642 (2017). <https://doi.org/10.1016/j.jvs.2016.11.044>
3. Duquette, A.A., Jodoin, P.M., Bouchot, O., Lalande, A.: 3D segmentation of abdominal aorta from CT-scan and MR images. *Comput. Med. Imaging Graph.* **36**(4), 294–303 (2012). <https://doi.org/10.1016/j.compmedimag.2011.12.001>. <http://www.sciencedirect.com/science/article/pii/S0895611111001480>
4. Hahn, S., Perry, M., Morris, C.S., Wshah, S., Bertges, D.J.: Machine deep learning accurately detects endoleak after endovascular abdominal aortic aneurysm repair. *Vasc. Sci. JVS* **1**, 5–12 (2020)
5. Jaeger, P.F., et al.: Retina U-NET: embarrassingly simple exploitation of segmentation supervision for medical object detection. arXiv preprint [arXiv:1811.08661](https://arxiv.org/abs/1811.08661) (2018)
6. Joldes, G.R., Miller, K., Wittek, A., Forsythe, R.O., Newby, D.E., Doyle, B.J.: BioPARR: a software system for estimating the rupture potential index for abdominal aortic aneurysms. *Sci. Rep.* **7**(1), 1–15 (2017)
7. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
8. Lareyre, F., Adam, C., Carrier, M., Dommerc, C., Mialhe, C., Raffort, J.: A fully automated pipeline for mining abdominal aortic aneurysm using image segmentation. *Sci. Rep.* **9**(1), 13750 (2019). <https://doi.org/10.1038/s41598-019-50251-8>

9. Lu, J.-T., et al.: DeepAAA: clinically applicable and generalizable detection of abdominal aortic aneurysm using deep learning. In: Shen, D., et al. (eds.) MICCAI 2019. LNCS, vol. 11765, pp. 723–731. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32245-8_80
10. López-Linares, K., et al.: Fully automatic detection and segmentation of abdominal aortic thrombus in post-operative CTA images using deep convolutional neural networks. *Med. Image Anal.* **46**, 202–214 (2018). <https://doi.org/10.1016/j.media.2018.03.010>. <http://www.sciencedirect.com/science/article/pii/S1361841518301117>
11. Siriapisith, T., Kusakunniran, W., Haddawy, P.: Outer wall segmentation of abdominal aortic aneurysm by variable neighborhood search through intensity and gradient spaces. *J. Digital Imaging* **31**(4), 490–504 (2018). <https://doi.org/10.1007/s10278-018-0049-z>
12. Taha, A.A., Hanbury, A.: Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. *BMC Med. Imaging* **15**(1), 29 (2015). <https://doi.org/10.1186/s12880-015-0068-x>
13. Xie, Q., Hovy, E., Luong, M.T., Le, Q.V.: Self-training with noisy student improves ImageNet classification. arXiv preprint [arXiv:1911.04252](https://arxiv.org/abs/1911.04252) (2019)
14. Zhuge, F., Rubin, G.D., Sun, S., Napel, S.: An abdominal aortic aneurysm segmentation method level: set with region and statistical information. *Med. Phys.* **33**(5), 1440–1453 (2006). <https://doi.org/10.1118/1.2193247>. <https://aapm.onlinelibrary.wiley.com/doi/abs/10.1118/1.2193247>



Automated Classifier Development Process for Recognizing Book Pages from Video Frames

Adam Brzeski, Jan Cychnerski , Karol Draszawka , Krystyna Dziubich,
Tomasz Dziubich  , Waldemar Korłub, and Paweł Rościszewski 

Computer Vision & Artificial Intelligence Laboratory,
Department of Computer Architecture,
Faculty of Electronics, Telecommunications and Informatics,
Gdańsk University of Technology, Gdańsk, Poland
{tomasz.dziubich,pawel.rosciszewski}@pg.edu.pl
<http://cvlab.eti.pg.gda.pl/>

Abstract. One of the latest developments made by publishing companies is introducing mixed and augmented reality to their printed media (e.g. to produce augmented books). An important computer vision problem that they are facing is classification of book pages from video frames. The problem is non-trivial, especially considering that typical training data is limited to only one digital original per book page, while the trained classifier should be suitable for real-time utilization on mobile devices, where camera can be exposed to highly diverse conditions and computing resources are limited. In this paper we address this problem by proposing an automated classifier development process that allows training classification models that run real-time, with high usability, on low-end mobile devices and achieve average accuracy of 88.95% on our in-house developed test set consisting of over 20 000 frames from real videos of 5 books for children. At the same time, deployment tests reveal that the classifier development process time is reduced approximately 16-fold.

Keywords: Book page classification · Machine learning · Business information management · Convolutional neural networks · Computer vision

1 Introduction and Motivations

Advancements of *mixed and augmented reality* (MAR) [1] find their applications, among others, in printed media [2]. The ubiquity of mobile devices and sensors opens new challenges for book publishers: books can be distributed along with software that augments their content with supplementary multimedia material, both audio and visual (augmented books). The utility of such applications often requires interactivity, meaning that the supplementary material should be

© Springer Nature Switzerland AG 2020

L. Bellatreche et al. (Eds.): ADBIS/TPDL/EDA 2020 Workshops and Doctoral Consortium, CCIS 1260, pp. 169–179, 2020.

https://doi.org/10.1007/978-3-030-55814-7_14

related to the contents of the book based on data from sensors such as camera of a mobile device. Additionally, usability imposes the requirement that the adjustment of the supplementary material has to be made in real time.

In this paper we focus on a specific *classification* task: given a video frame from a mobile device camera, recognize if the camera is facing a book page and if so, give the exact corresponding page number. It should be noted, that the camera can potentially face only small fragments of the page (printed page number does not have to be necessarily visible), from various angles. Solutions to this problem are crucial for MAR applications that provide different material depending on the currently observed page.

Arguably, the most challenging requirement towards such a classifier is that it has to be developed based on a very limited data set, namely the digital version of the book, consisting of one camera-ready original of each book page. Acquisition and labeling of camera recordings of printed books are infeasible for the publishing companies. At the same time, the classifiers should be robust towards many varying conditions, that are hard to anticipate, such as video quality and resolution, illumination, way of targeting the book with the camera, etc. These changing conditions can be particularly unpredictable if the application is used by small children (which is often the case), because they tend to be reluctant to perform the demanding task of accurate aiming.

Another critical requirement towards the classifier is related to computational efficiency. Classification has to be done in real time while utilizing potentially limited computational power of the target mobile device. This requirement leads to narrowing of the set of feasible machine learning models, as well as implies a need for model performance evaluation in real environment. The main contribution of this paper is an automated classifier development process that allows to train book page classification models based on *convolutional neural networks* (CNNs) that:

- can be successfully trained on limited data sets (one digital original version of each book page);
- do not require manual feature extraction and tuning during the development process;
- achieve high accuracy on a test set consisting of frames from real videos with various realistic conditions, including lighting, angles and distances to books, camera stabilization and page folding;
- are suitable for real time execution on mobile devices.

The paper is organized as follows. Related work is characterized in Sect. 2. The traditional and the proposed classifier development process is described in Sects. 3 and 4. In Sect. 5 we describe our experimental setup and in Sect. 6 we report and discuss results of classifier accuracy tests, as well as deployment and end user usability tests. Finally, summary of the paper and future work directions are given in Sect. 7.

2 Related Works on Automatic Book Page Classification

Identification of book pages and page number reading have multiple applications in the publishing process. One example is recognition of duplicate pages in paper-based books in an automated manner. It is also essential in some context-aware applications. Another example of a practical use is book digitization. During this process, flipping is often made by scanners. It is a challenging task that includes flipping the pages at random speed and direction. Thus the validation of page number is of crucial importance to improve achieved accuracy.

There are two common methods of page number recognition: based on add-on devices or using computer vision algorithms.

The former are related to context-aware and ubiquitous processing. They allow automatic retrieval and presentation of digital information by recognizing the contents or the tag printed on a page, and direct manipulation of digital information by gesture recognition. Page number identification is based on an accumulation of page-flipping events, which limits the resolution of the reading position in a book to two pages.

In [3] authors proposed a flip sensor - the device (brush-shaped) mounted on the spine of a book and connected to a microcontroller which detects page-flipping. The accumulated number of page-flipping events can approximate the current reading position and provide "context-aware" multimedia information. In contrast, in [4] authors used RFID technology to recognize the opened book page. A thin flexible transponder tag with a unique ID is embedded in the paper of each page, and a special tag-reader is affixed to the binding of the back of the book. As the pages turn, the tag-reader notices which tags are within its read range and which have moved out of its range.

The second group of computer-vision-based approaches is mostly represented by solutions that utilize OCR libraries e.g. Tesseract. After initial preprocessing steps (e.g. denoising or rotating), source image is submitted for OCR analysis and, in the next step, passed to a classifier or other machine learning techniques.

Garris [5] described the process of scanning the Federal Register and validating the document images (ca. 67 000) within the database. The following flow has been used: scanning a page, saving it to a graphic file which name includes the page number, OCR, localizing the block with arabic numbers, creating subimages, recognition of a page number block, acquiring page number and verifying with filename. It should be noted that the document format was well-known.

As OCR software, the authors facilitated HSFYSYS 2.0 that uses a Multi-Layered Perceptron (MLP) to classify a segmented character. For 64.384 sub-images, system accuracy for OCR of the page number field achieved an 88.1% overall correct recognition. Usage of such system allowed to automatically validate and exclude over 83% of the images from being adjudicated by a human. Over 90% of the remaining 17% of images were validated by an operator adjudication.

In [6] authors consider to removing duplicate pages from video stream. They recognize and exploit information about position of page edges and direction of the flipping. They obtained 95% accuracy (processing time ca. 1.3s, resolution

1920 × 1080, 50 fps). However, the proposed method gives no information about neither a current position nor number of flipped pages. The authors did not provide any information about system's response to flipping more than one page at a time.

Chang et al. [7] presented Interactive Multimedia Storybook Demonstration System that allows users to browse multimedia content by turning pages of a physical storybook, scribbling with an infrared pen, and interacting with a virtual keyboard. One of the three basic functionalities is page number recognition. Unfortunately, page number is not expressed as an arabic number, but represented by circle-shaped object at the top. The proposed algorithm is a combination of computer vision techniques. The image is smoothed using a Gaussian filter to remove noise and segmented using the Otsu's thresholding to either foreground or background objects. Further, morphological processes are used to refine the shapes. Finally, the connected-component labeling is used to count the number of the connected-components (i.e., red shapes), representing the current page number. Owing to these facts, system response time is very low (<0.1s per page) and results in excellent user experience.

Nowadays, due to the increasing popularity of mobile devices, many software libraries and platforms for augmented reality are developed that offer image recognition and image tracking functionalities. The most popular method is a sensor-based one that utilizes built-in sensors (e.g. gyroscopes, accelerometers) and consists of two steps: measuring the location of the image through the sensor and matching the sensor coordinate system to the image coordinate system. The latter is a markerless tracking step, that is a method of tracking image features naturally without attaching artificial markers [8]. The images that the software can detect and track are called Image Targets (ITs) (e.g. book page with or without a page number). ITs are created from JPG or PNG input images and represented as sets of features (markers) extracted from these images, stored in a database and used for run-time comparisons. A special collection with sample background images can be provided either by the MAR platform producer or by the application developer. That method has an advantage of recognizing lighting changes, direction, rotation angle, partial overlap based on the ITs features. Implementation methods include Scale-Invariant Feature Transform (SIFT) and Histogram of Oriented Gradient (HOG). SIFT compares the original image to the feature points of objects (markers). Even if the IT is covered it can still be identified through its feature points. This method is suitable if the IT has been rotated or if the inner pattern is simple and the object can be identified from the target outline. However, markerless tracking method has the disadvantage that it cannot guarantee its real-time performance in a smartphone environment due to lower performance.

Examples of solutions using markerless tracking are Vuforia SDK [9] and Wikitude SDK [10]. Vuforia SDK uses a set of algorithms to detect and track the features that are present in an image by comparing them against a database of known features. Once detected, they are tracked along the camera's field of view. The SDK can detect and track up to five targets simultaneously depending

on the CPU and GPU load. Number of IT datasets in its resource database can be swapped at run time but are limited to approximately 50. Wikitude SDK performs well when ITs are not close to the user (IT of A4 size can be recognized from 2.4m away). The producer recommends to limit the number of concurrently trackable targets (up to 5), set the threshold to register distance changes (up to 10mm) and set up the IT physical dimensions due to a performance optimization.

3 Traditional Classifier Development Process

In the traditional development process for MAR applications considered in this study, book page classification is a crucial step, both in terms of algorithm complexity and human involvement. At the very first stage, a developer prepares a set of ITs, where each page is a single IT (see Fig. 1). For a typical children book this means around 30 ITs. Next, each IT is entered into an AR framework and a data set is built in an automated manner (a feature vector is determined by the utilized framework). After preparing the full data set (including all ITs and backgrounds) and assigning the camera's basic point of view, the application is built and deployed on a mobile device. The developer subjectively evaluates the efficiency and accuracy of classification. In the case of poor outcomes, he modifies the scene and the whole process is repeated. Otherwise, the application is ready to deploy on an end user device.

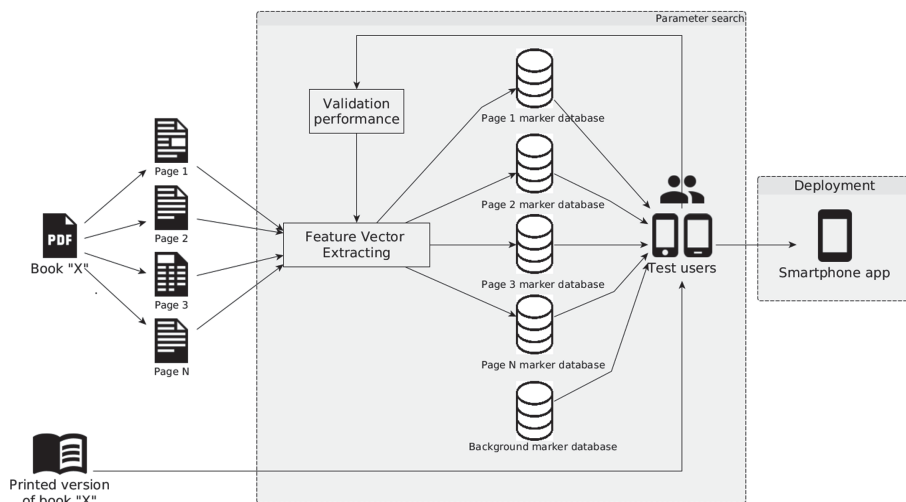


Fig. 1. Traditional classifier development process.

Because of the necessity of such iterative development, for a typical children book, the described process takes approximately 80 working hours. This significantly limits scalability of introducing new books into the MAR application, so

an improved classifier development processes that would automate this step are highly desirable.

4 Proposed Automated Classifier Development Process

To tackle the described problem, we designed our solution in the form of the overall pipeline which is presented in Fig. 2.

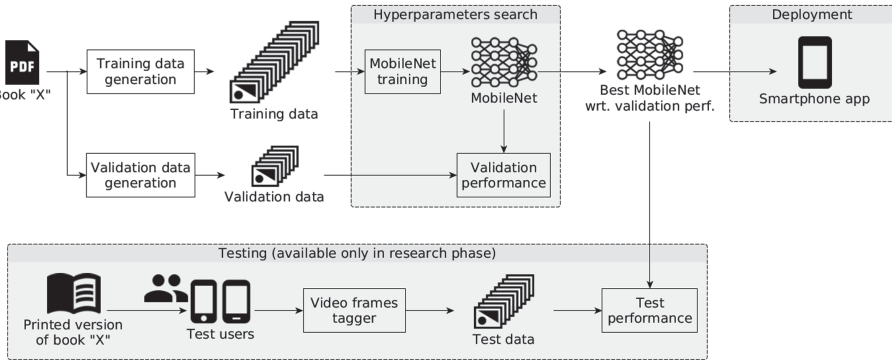


Fig. 2. The pipeline of our solution.

This pipeline is centered around the training of a deep CNN model for book pages classification. The only input to the training algorithm is a digital version of the book. The need for test users working with the printed version of a book is no longer needed in a production process of new classifiers for new books.

To train the model, it is necessary to generate synthetic low-resolution cropped images similar to frames from a video with book pages taken by a mobile device camera. In order to achieve this, the input PDF of a N -paged book is first divided into N raw high-resolution PNG images of individual book pages. Then, a large training set is built in a heavy augmentation process, consisting of image crops, rotations, noise addition as well as more nuanced lighting change simulations, warps and perspective changes. To make the classifier robust towards images which present a page content only partially, or even inputs that do not present page content at all, we also use a set of generic background images in the process. The final result is a large image data set with of $N + 1$ classes (additional *not-a-page* class).

The training and validation sets obtained this way are used in the CNN training phase, which results in a classifier model suitable for deployment in a mobile application. Because the whole process has to be fully automated, the training phase is repeated a few times with different hyperparameter settings and the model with best validation performance is eventually chosen.

5 Experimental Setup

In this section we describe our experimental setup, starting with the rationale behind selecting a specific CNN model in Sect. 5.1. The development process proposed in this paper does not require printed version of a book item to develop a page classifier for this item. The methodology adopted during our research includes testing whether training and validation with semi-synthetic data do indeed lead to classifiers that generalize well to real video frames input. Hence, our in-house pipeline depicted by Fig. 2 is enriched with additional testing branch, which is described in Sect. 5.2.

5.1 Considering Hardware Constraints in CNN Model Selection

The purpose of this research is to develop an efficient method for classification book pages, that can be run on a mobile platform. As a requirement, classification time on a low-end mobile phone (e.g. Maxcom Smart MS453 - available for around \$50 for a new device) should not exceed 500 ms. Moreover, data generation, hyperparameter tuning and training time are also considered important—the whole process on a high-end consumer-class CPU and GPU (like NVIDIA RTX 2080) should not exceed 24 h.

To meet these requirements, we decided to use MobileNet-v2 [11], as it is one of the most successful neural networks designed for mobile platforms. Also, this network allows to adjust its size freely, which allowed us to find a compromise between efficiency and processing time. Our deployment tests on Huawei P10 Lite, Motorola Moto X Play and Maxcom MS453 showed that setting MobileNet’s $\alpha = 0.35$ is enough to achieve required classification speed.

5.2 Test Sets and Metrics

To test the performance of our approach in the real setting, for each book we collected a test set in the following way. We asked volunteers to record short (5–10 s) videos of each book page by hovering their mobile devices over physical copies of the book. They were asked to take the recordings in various lighting conditions, from different angles and distances to books, with and without stabilization of their cameras. The name of each video encodes the book id and page number it presents. This way we obtained as diverse and realistic data as we could. All the frames from these videos put together constitute a test set for a given book.

Such test set was then used to estimate the overall classification accuracy of our system (denoted by $h(\cdot)$ function) in the real setting:

$$Acc(D_{test}) = \frac{100\%}{|D_{test}|} \sum_{(x_i, y_i) \in D_{test}} \mathbf{1}(h(x_i) = y_i) \quad (1)$$

However, to possess detailed performance characteristics of our system, we decided to measure (1) separately for 3 types of movie frames:

1. Frames in which the whole or almost whole book page is visible (roughly 90% or more of the page is fits into the frame) – these frames form test data D_{page} , and accuracy measured on this set (denoted as $Acc(D_{page})$) we call *page test accuracy*.
2. Frames in which the page is only partially visible (typically because the recording device being very close to the book) – these frames form test data D_{zoom} , and $Acc(D_{zoom})$ is referenced as *zoom test accuracy*.
3. Frames in which, for some reason, the page is not present (or just a marginal part of it) so that the frame contains almost exclusively some background objects – these frames form test data $D_{negatives}$, and $Acc(D_{negatives})$ we call *negatives test accuracy*. All samples from this set are labeled as *negative*, which is treated during training and testing as a separate class (so our classifier has to distinguish between $n_{pages} + 1$ classes). To enlarge $D_{negatives}$, we aggregate all negative frames (from all the books) – all books have the same negatives test set.

To obtain detailed statistics, all collected videos were carefully labeled *frame by frame* by our team to one of frame types: page, zoom, negative, as well as macro and blur. This way, for each book, we split its D_{test} into disjoint sets so that:

$$D_{test} = D_{page} \cup D_{zoom} \cup D_{macro} \cup D_{blur} \cup D_{negatives}$$

The frames of the last two types were discarded during our performance tests, as they typically contained not enough information even for human annotator to correctly classify the pages they show (or, rather fail to show).

In total, we collected over 20 000 labeled frames, namely: macro – 2175, negatives – 2382, blur – 204, page – 7254, zoom – 8232.

Because neighboring frames are typically relatively similar, in order to speed up the testing procedure, for our experiments we used every fifth labeled frame.

6 Results and Discussion

Overall and detailed results for *page*, *zoom* and *negatives* subsets were presented in Table 1. The accuracy results achieved by the acquired classifiers for the evaluated set of 5 books are fairly high even for the highly difficult *zoom* test subset (average of 74.5%) and for the out-of-domain, negative subset (average of 94.63%). Error analysis of the classifiers over the test set show 2 major sources of wrong predictions. Firstly, some of the books include pairs of highly similar pages, that can differ only in small graphic details or presence or absence of small block of text. This case is especially problematic for *zoom* test subset, where image crops extracted from different pages may be exactly identical, causing accurate prediction to be impossible. Data ambiguity is therefore a part of the complexity of the investigated problem, however models can be still improved in that field. The second source of errors are blank or plain pages. Such pages in many cases are classified as non-page (negative), due to high similarity to the background class images. This is obviously undesired and it constitutes a weakness of the approach.

Table 1. Best accuracy results for train, validation and test set out of 3 runs of hyperparameter search. Max epochs = 100, early stopping on overall validation accuracy with patience = 30.

Book	Pages	Train	Val	Test			
				Page	Zoom	Neg	Total
Book 1	33	99.91	99.25	98.24	77.36	93.81	89.80
Book 2	26	99.99	99.85	95.73	73.93	91.45	87.04
Book 3	26	99.98	99.94	96.23	66.07	92.97	85.09
Book 4	26	99.96	99.91	99.39	71.33	95.83	88.85
Book 5	26	99.99	99.91	98.99	83.80	99.07	93.95
			Average:	97.72	74.50	94.63	88.95

Since one of the purposes of the developed models was to fulfill an imposed classification latency limits of 500 ms for possibly low-performance mobile devices, we conducted performance tests on a set of budget smartphones, including Huawei P10 Lite, Motorola Moto X Play and Maxcom MS453. The tests confirmed that the latency remained under 500 ms for all of the devices.

In order to further assess the quality of the complete MAR applications developed within the proposed process, we organized end user tests involving children users, in April 2019. 20 children from 2 to 9 years old were invited to the tests (10 of each gender, each child was under control of a legal guardian). Two test applications have been prepared and were tested on smartphones with both Android and iOS operating systems. During the experiment, the legal guardians were asked to determine the children’s emotional state during MAR application testing. The results showed that curiosity appeared in 90% of children, engagement – 65%, enjoyment – 45% and excitement – 20%. There have also been states such as boredom – 15% and indifference – 15%. The legal guardians were asked to rate specific areas (on a scale of 1 to 5, where 1 – poor, 5 – excellent). The average grades were as follows: application performance – 4.45, ease of use (intuitiveness) – 4.45, adaptation to preschool children – 3.95, adaptation to early school children – 4.8.

Finally, we also performed the proposed classifier process for additional 37 books and evaluated the average time required to prepare a single publication, which was approximately 5 h of human work (3 h of IT specialist’s work + 2 h of tester’s work) and approximately 6 h of machine learning computations.

7 Summary and Future Work

The main contribution of this work is a new image classifier development process for MAR applications utilizing book page recognition from video streams. Unlike the traditional development process in this field, the proposed solution automates the step of feature vector extraction and performance validation, which significantly reduces human labour time required to introduce MAR capabilities to a

new book item. The proposed automated classifier development process allows to train accurate book page classifiers on limited training data. Specifically, only the minimum possible input is required, in a form of single digital copy of each book page. Hence, there is no need for obtaining printed book copies in order to develop classifiers.

Tests performed on our in-house dataset of densely annotated real video frames show that the proposed method achieves good page classification accuracy, while end user tests of the complete MAR applications revealed good results of subjective usability assessment. Due to utilization of computationally efficient MobileNet-v2 architecture for classifier models, low prediction time was achieved, enabling providing low latencies in real applications even for low-price smartphones. At the same time, deployment tests show that the human work required to introduce MAR capabilities to a new book item was reduced from approximately 80 to approximately 5 h.

One of the directions of future work is improving automatic training and testing procedure. Instead of artificially generated validation set, real data containing videos other books might be used to initialize network weights and to estimate classifier accuracy. Moreover, training the network with real data (instead of only generated data) shall improve its performance. Furthermore, boosting procedures might be used during training, as some book pages are less distinctive than others. It should be possible to automatically detect such pages during training and adjust neural network optimizer to focus on them.

Acknowledgements. This work has been partially supported by Gdańskie Wydawnictwo Oświatowe and Statutory Funds of Electronics, Telecommunications and Informatics Faculty, Gdansk University of Technology.

References

1. Costanza, E., Kunz, A., Fjeld, M.: Mixed reality: a survey. In: Lalanne, D., Kohlas, J. (eds.) *Human Machine Interaction*. LNCS, vol. 5440, pp. 47–68. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00437-7_3
2. Hull, J.J., et al.: Paper-based augmented reality. In: *17th International Conference on Artificial Reality and Telexistence (ICAT 2007)*, Denmark, November 2007, pp. 205–209. IEEE (2007)
3. Fujinami, K., Inagawa, N.: Page-flipping detection and information presentation for implicit interaction with a book. *Int. J. Multimed. Ubiquitous Eng.* 4(3), 20 (2009)
4. Back, M., Cohen, J., Gold, R., Harrison, S., Minneman, S.: Listen reader: an electronically augmented paper-based book. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI 2001*, Seattle, Washington, USA, pp. 23–29. ACM Press (2001)
5. Garris, M.D., *Creating and validating a large image database for METTREC*. Technical report NIST IR 6090, National Institute of Standards and Technology, Gaithersburg, MD (1997)

6. Chakraborty, D., Roy, P.P., Alvarez, J.M., Pal, U.: Duplicate open page removal from video stream of book flipping. In: 2013 Fourth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), Jodhpur, India, December 2013, pp. 1–4. IEEE (2013)
7. Chang, Y.-H., Liao, H.-L., Jeng, L.-D., Chiu, Y.-C.: An interactive multimedia storybook demonstration system. *Multimed. Tools Appl.* **74**(17), 6709–6728 (2014). <https://doi.org/10.1007/s11042-014-1926-1>
8. Jang, S.-W., Ko, J., Lee, H.J., Kim, Y.S.: A study on tracking and augmentation in mobile AR for e-Leisure. *Mobile Inf. Syst.* **2018**, 1–11 (2018)
9. PTC. Developer’s guide (2019)
10. Wikitude GmbH. Developer’s Guide (2020)
11. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C.: MobileNetV2: inverted residuals and linear bottlenecks, March 2019. [arXiv:1801.04381](https://arxiv.org/abs/1801.04381). [arXiv: 1801.04381](https://arxiv.org/abs/1801.04381)

**1st Workshop on Scientific Knowledge
Graphs (SKG 2020)**



DINGO: An Ontology for Projects and Grants Linked Data

Diego Chialva^(✉)  and Alexis-Michel Mugabushaka 

ERCEA, Place Charles Rogier 16, 1210 Brussels, Belgium
{Diego-Valerio.Chialva,Alexis-Michel.Mugabushaka}@ec.europa.eu

Abstract. We present DINGO (Data INtegration for Grants Ontology), an ontology that provides a machine readable extensible framework to model data relative to projects, funding, actors, and, notably, funding policies in the research landscape. DINGO is designed to yield high modeling power and elasticity to cope with the huge variety in funding, research and policy practices, which makes it applicable also to other areas besides research where funding is an important aspect. We discuss its main features, the principles followed for its development, its community uptake, its maintenance and evolution.

Keywords: Ontology · Linked data · Research funding · Research projects · Research policies

1 Introduction, Motivation, Goals and Idea

Services and resources built around Semantic Web and linked (open) data technologies have been increasingly impacting research and research-related activities in the last years. Development has been intense along several directions, for instance in “semantic publishing” [36], but also in the aspects directed toward the reproducibility and attribution of research and scholarly outputs, leading also to the interest in having Open Science Graphs interconnected at the global level [21]. All this has become more and more essential to research practices, also in light of the so-called reproducibility crisis affecting a number of research fields (see, for instance, the huge list of latest studies at <https://reproduciblescience.org/2019>).

In fact, the demand of easily and automatically parsable, interoperable and processable data goes beyond the purely academic sphere. The research landscape comprises a vast number and type of activities, with multiple and diverse stakeholders, actors and with impact on several aspects and sectors of society. One aspect of huge relevance is the funding of research, together with the related policies for science development and sustainability.

Disclaimer. The views expressed in this paper are the authors’. They do not necessarily reflect the views or official positions of the European Commission, the ERC Executive Agency or the ERC Scientific Council.

Machine-actionable, inter-operable data is in huge demand in those respects. On the one hand, for instance, research funding agencies face increasing pressure to report on impact derived from their activities. This has to be seen in a broader context of the increased role that research assessment play in research policy debates. On the other hand, researchers and research organisations are asked more and more to conform to policy specifications in order to obtain and secure their funding. The compliance to funding and research polices is also part of the wider debate about best research practices such as Open Science, Open Access, FAIR data and sustainable research.

Research assessment and compliance verification at any level involves collection, management and analysis of a great increasing deal of data of different types and from multiple sources . The classical way to meet this demand has been to collect data directly from various research actors. This increases the burden on researchers, university administration and funding agencies, as those data has to be managed and curated. Moreover, the information, typically collected in an “ad hoc” way and in isolation, is not available to others. This results also in duplication of efforts, due to the necessity to re-do the linking and processing of data. The difficulty of data linking and semantic interpretation across different realities and agencies also entails that data and analysis are of limited value when it comes to put them in broader perspective.

Solving these problems entails having data that can be easily parsed, processed and interpreted computationally. This requires expressive shared machine-processable descriptions and models on the Web. Technologies as RDF, RDFS, OWL, and SPARQL provide building blocks towards that goal and have favoured the development of ontologies to describe various aspects of the research domain.

However, the development of ontologies for the funding aspects of research and their relations to research activities, actors is still quite in its early stages. In particular, while few ontologies exist (see Sect. 2), they mostly envisage only some of the important semantic elements (typically those relative to projects and grants), as we will show.

This note presents a novel ontology, developed to manage data on research grants and projects, but also notably to conceptualise funding policies and instruments, facilitating the integration and interoperability of such information with other data and from various sources in the framework of the so-called Linked Data. The ontology has been dubbed DINGO (Data INtegration for Grants Ontology). It provides an extensible, interoperable framework for formally modeling the relevant parts of data in this knowledge area.

DINGO particularly facilitates the effort of putting analysis of funding activities and policies in broader context and comparative perspectives, which is much needed when assessing research, policies and their impact. In this way, DINGO will be beneficial in practice at several levels. For instance, by increasing the capacity of analysis to inform policy and strategic discussions, as well as reducing the effort of researchers and officers in giving evidence of policy compliance.

Indeed, one specific characteristic of the knowledge area DINGO aims to describe is its variety. The existing funding activities and policies show a large

spectrum of practices, with remarkable diversity and complex semantics. This constitutes a serious difficulty when trying to put funding activities and policies in context and comparative perspectives. DINGO has therefore been specially designed to cope with this, by a rigorous conceptualisation of commonalities via a number of ontology classes and properties, together with other classes that allow tuning semantic specializations to the specific cases when modelling data.

This also allows DINGO not only to be effectively used as a pure domain ontology specific to research activities, but in fact to perfectly model even other domains where funding activities play a relevant roles (such as the arts, cultural conservation, and many others). DINGO has therefore, in some respects, also the multi-domain usability typical of more upper ontologies (we use here the classification and definitions of ontologies by Guarino [16]).

DINGO is fully documented at <https://w3id.org/dingo>, and a machine readable version of the ontology is available at <https://w3id.org/dingo> in RDF-Turtle by redirection when visiting with the “text/turtle” header (it is also available at <https://dcodings.github.io/DINGO/DINGO-OWL.ttl>).

This article is organised as follows. Section 2 discusses related work. Sections 3, 4, 5 and 6 and subsections thereof present the aims, development guidelines, community uptake, maintenance and evolution, and main features of DINGO (we leave the detailed description of the ontology to its documentation, available online). We conclude in Sect. 7, where we also comment on future potential directions of development.

2 Related Work

A few works exist modelling data related to funding and research, although to our knowledge none has been dealing with the aspects pertaining to research (funding) policies together with the rest.

One of the earliest efforts to create a data model for the management of research funding data is CERIF (Common European Research Information Format), [22]. It is an extremely rich and detailed vocabulary for research management, with a considerable number of entities and relations, and a high granularity. However, it does not conceptualise aspects related to policies.

CERIF, conceived for CRIS (Current Research Information Systems), has deep roots in relational database modeling more than in the semantic/knowledge graph one, as visible from some of its characteristics. For example, one of its main features is the presence of “link entities” such as project-organisation, project-person, and so on. They are in fact relationships rooted in relational database reification practices (which differ from what reification is in the framework of knowledge graphs and semantic web). Such “link entities” have however less straightforward interpretation in terms of semantic concepts (they often represent couples of concepts), which would affect inferences. We will show how DINGO avoids this problem and yet manages to capture the aspects of interest.

Related to CERIF is the OpenAire data model [24]. *OpenAire* [23] is an infrastructure that links research outcomes to their creators, enabling discoverability, transparency, reproducibility and quality-assurance. The OpenAire data

model uses part of the CERIF vocabulary (including some of the “link entities”) and combines them with the OpenAire guidelines.

A few OWL-based ontologies exist describing funding in research. Compared to CERIF, they are fully framed in semantic modeling. The most well-known ones (and in fact the only ones to our knowledge) are FRAPO (Funding, Research Administration and Projects Ontology) [14,29], and the Springer Nature Sci-Graph Ontology [34].

These are actually part of larger ontologies or ontology collections mainly aiming at categorizing scholarly data, such as publications and other similar outputs, rather than focusing exclusively on the funding and research landscape. They are thus tuned for those other purposes and have specific limitations. For example, the SciGraph one does not appear to distinguish the concept of “grant” as funding from the concept of “research project” and thus would not allow to easily model for many existing funding practices and uses cases (for instance, the case of projects with multiple grants, either co-occurring or in sequence). FRAPO instead lacks classes and properties for relevant concepts such as “principal investigators” and others. Moreover, neither ontology conceptualises the domain of funding policies.

In addition to these, there is a growing number of initiatives addressing other dimensions of research data than the funding-project ones. To cite a few: *OpenCitations* [30], which is dedicated to open scholarship and open bibliographic and citation data; *SMS* (Semantically Mapping Science) [3], a platform integrating heterogeneous datasets for science, technology, innovation studies; *VIVO* [9] an open source software and ontology for representing scholarship and scholarly activity. Finally one can mention also *CASRAI* (Consortia Advancing Standards in Research Administration Information) [7], which does not provide an ontology, but a glossary of research administration information.

We will discuss the part of schema.org [33] dealing with funding data in Sect. 3, as it was in fact inspired by DINGO.

We finally would like to mention the FP Ontologies [26]. They do not deal with research funding, but model some aspects of projects. Web-searching them points to the webpage at [26], but in fact we could not find documentation nor download any serialisation from that page.

3 Community Uptake and Use of DINGO

DINGO has been first presented to the public in the late 2018, and has led to a number of uses, both directly for data modeling and knowledge bases creation, and as a basis or inspiration for related ontology modeling efforts.

The first public presentation of DINGO has occurred at the workshop “Wikidata for research”, Berlin, 17–18 June 2018, where feedback and input were exchanged with a working group of participants, which lead to the linking of DINGO with the Wikidata graph.

DINGO also inspired the part of the schema.org model specific for grants and funding (as mentioned explicitly at the issue 343 of the schema.org release of 2019-04-01¹). Schema.org's model covers however only a subset of DINGO's.

Furthermore, DINGO has been adopted to model the knowledge base of the European Commission data hosted and available now in the OpenAire LOD service (at <http://lod.openaire.eu/eu-open-research-data>), and as one of the basis of the schema for the GRANTID initiative of CrossRef [10] (one of the authors of this article, D.C., has been a member of the technical group for the schema²).

4 Ontology Mapping, Reuse and Extensions in DINGO

Ontology mapping is a key challenge of the Sematic Web and of Linked (Open) Data for several reasons. Ontology reuse is also a good knowledge engineering practice, increasing the interoperability of systems.

In the framework of semantic modeling and the Semantic Web, reuse and mapping are particularly complex. On the one hand, the de-centralised nature of the web favours the development of several ontologies and data models, which often overlap partially. On the other hand, the single ontologies are generally created with specific goals, and thus even when they are developed to model data from the same domain(s), they will generally present subtle semantic differences even in seemingly general concepts.

In the case of research data, mapping and reuse are further complicated by the multiplicity of actors and the diversity of types of funding practices, policies and data. But on the other hand, this same issue prompts to maximise the semantic modeling power of an ontology by linking it with overlapping ones in order to achieve maximum interoperability.

DINGO was therefore built from the start with a particular attention to ontology mapping. Pure reuse has been possible only to a certain extent, because ontologies covering overlapping knowledge areas (such as those mentioned in Sect. 2) do indeed present subtle but relevant semantic specificities.

The mapping in DINGO makes use of the SKOS ontology/data model mapping properties (documented at [1]) and RDF and OWL class and properties axioms such as owl:equivalentClass and owl:equivalentProperty when applicable. In fact, the establishment of mapping using the latter owl axioms is generally quite complicated, as they require establishing that the full extension of the relative classes/properties are equal. This is typically a difficult task in the case of a complex knowledge area such as the one of research, and has therefore being done carefully and rather conservatively in DINGO.

DINGO is presently mapped to the Wikidata data model, to schema.org and to the FRAPO ontology. There is also interest in linking DINGO with the vocabulary provided by CERIF, and future developments have been already planned in that sense.

¹ Visible at <https://schema.org/docs/releases.html>.

² See <https://www.crossref.org/working-groups/funders/>.

Besides that, DINGO also reuses several other ontologies, such as SKOS, schema.org and DublinCore [11], and is inspired by the FAIR principles [39] for data publication.

Finally, DINGO has been designed to be easily extensible to adapt to the various possible use cases and diversity of data and existing practices. The ontology presents “hook properties” (such as *product_or_material_produced*) that allow to extend DINGO linking, for instance, to data modeled with the many ontologies dealing with scholarly and publishing data (such as the SPAR ontologies [29], the Semantic Web Journal (SWJ) ontology [20], the Semantic Web Conference (SWC) ontology [37], the Semantically Annotated LaTeX (SALT) ontologies [15], the Nature Ontologies [18], the SciGraph Ontologies [19], the Conference Ontology [25], BIBFRAME [4] and *bibliotek-o* [5]).

5 The DINGO Ontology

5.1 Aim of the Ontology

The principal aim of the DINGO ontology is to provide a machine readable extensible framework to model data relative to projects, funding, policies and actors. The original intended users for such frameworks were the stakeholders in the research landscape with their very different use cases.

As discussed in Sect. 1, semantical modeling of that knowledge area faces, among others, one main difficulty: there exist a huge variety of funding, policies, practices and research activities. Due to the aim of being able to cope with this, as we illustrate also in Sect. 5.3, DINGO is finally applicable also to domains different than the one of pure research where the funding aspects are relevant, for example in the arts, cultural conservation and the like.

DINGO’s development was also driven by the goal of being rich enough to

1. integrate and accommodate existing systems and data instances
2. satisfy complex as well as simple use cases, also by straightforward extension.

This set of principal design goals and requirements also allowed to work toward the realisation of additional (and important) objectives, such as promoting the opening up of funding data, and the linking and re-using of data.

Special care has been devoted to minimizing the efforts in applying/adopting the model by users. In particular, while the model has been created using Linked Data fundamentals, it is apt to different implementations and integration in non-graph-type data bases, hence it does not address specifically the optimization of graph inference and graph-based queries.

5.2 Approach to Ontology Design

Ontology generation is a complex process that has been scrutinised in the literature and has led to the establishment of a number of engineering best practices, see for example [13, 16, 17, 31, 32, 38]. The design of DINGO has followed such best practices. The main guidelines followed have been:

- a mixture of middle-out and bottom-up approach: starting from actual data (such as funding data from various agencies, see below), several main concepts have been designed and the ontology generation has proceeded by distinguishing a number of commonalities (generalisations) and specificities; the advice of domain experts has also been essential, mostly profiting from the fact that DINGO has been developed at the ERC(EA) [12]
- practical usability of the end results
- interoperability/integration from the inception with other graphs (for instance, Wikidata and Schema.org)
- sufficient granularity to allow for efficient monitoring and evaluation purposes, but also sufficient generality to accommodate potentially all funding data, thus providing the whole benefit of a large Linked Data Graph. DINGO is straightforwardly extensible to provide additional granularity
- coverage of all areas of interest, also for non-academic actors and stakeholders.

For DINGO’s data-based mixed middle-out and bottom-up development we have used various research funding data, in particular looking at data freely provided by several funding agencies. For instance, we have used data from the European Union Funding (Research Framework Programmes), The Australian Research Council (ARC), the Swiss National Science Foundation (SNSF), the Croatian Science Foundation, the US National institute of Health (NIH), the US National Science Foundation (NSF), the various UK agencies coordinated by the Research Councils UK (RCUK).

Finally, we have adopted elements of agile development, not dissimilarly from what proposed in [28], for instance concerning unit testing.

The tools employed in the design and coding process of DINGO have been: UMLet [2] with some custom diagrams elements for graphical representations, while the documentation has been build using a custom software written in Python (unpublished) to automatically generate human-readable HTML documentation from OWL ontologies serialisations (see Sect. 5.4).

5.3 Ontology Description

Here we describe DINGO’s main components and their features, while the ontology full specification is available at <https://w3id.org/dingo>.

DINGO is an OWL-DL ontology comprising 40 classes and 68 properties. Its classes provide an articulated conceptualisation of entities relevant for the characterisation of data in the research, funding and research-related domain. In particular, besides classes for Projects, Grants, FundingAgency and others, there are specific classes for describing funding policies, with several specific subclasses (which can straightforwardly be expanded).

As we said, the variety and diversity of funding realities (which we will also call “realisations”) makes semantic conceptualisation particularly difficult. For example, different funding agencies/funders classify their funding policies in various and discording ways, sometimes using the same word for different things (for instance, the terms funding scheme/programme/action). Also the role and

characterisation of the different actors in projects and grant agreements are quite diverse. Such modeling complexity appears not only at the level of concepts, but also of relations/properties. Notably, the relationships between the funding and the research enterprise can be various and rather complex.

Furthermore, alongside concepts definition, additional complexity is given by the variety of use cases: besides the simple case of one grant funding one project, often multiple fundings are attached to a single project (either in sequence or at the same time), or a single grant funds several (sub)projects.

Therefore, DINGO's properties and classes have been designed to allow high modeling capability to represent such variety of concepts and realisations.

DINGO's main features are as follows:

- it defines a number of principal classes: **Project**, **Grant**, **Funding Agency**, **FundingScheme**, **Role**, **Person**, **Organisation**, **Criterion**, various sub-classes of those and some related specialised classes;
- a **Project** is an organised endeavour (collective or individual) planned to reach a particular aim or achieve a result
- a **Grant** is a disbursed fund paid to a recipient or **beneficiary** and the process for it; DINGO focuses on the main definition of “funding” (which is defined as “money for a particular purpose; the act of providing money for such a purpose” both in the Cambridge, Oxford and Collins dictionaries [6, 8, 27]), but can be extended to other types of funding (non-monetary ones), see Sect. 6.
- a **Project** may be funded by one or more Grants simultaneously or in sequence
- a **Grant** may fund one or several Projects
- **Grants** can be awarded to Person(s) and/or to Organisation(s)
- **Projects** can be participated by **Person(s)** or by **Organisation(s)**, hence a **participant**, characterised by a **Role**, can be a **Person** or an **Organisation**
- the **Role** class can be used to specify the semantics of the participation to a Project or role in a Grant. This class provides instruments to model a large variety of semantic types, to account for the variety of practices found in actual data
- types of organisations can be specified using one of the several sub-classes of **Organisation** or creating new ones
- a **participant (Person or Organisation)** in a **Project** may not actually be **beneficiary** of a specific **Grant** funding the Project; accordingly, DINGO reflects that particular participants of Project and beneficiaries of Grant funding the same Project may be different
- temporal aspects of the various concepts can be fully modeled, and are expressed by specific properties (*start_time*, *end_time*, *inception*, and so on)
- **Funding Agencies** are the organisations materially disbursing and administering the Grant process
- **Funding Schemes** are funding instruments accompanied by specifications of Grant coverage, eligibility, reimbursement rates, specific criteria for funding, grant population targets, and similar features. Such specifications constitute one or more **Criterion** to award funds (Grants);

- **Funding Schemes** may be sub-specifications of other Funding schemes; this recursive relation allows to model existing complicated hierarchies of funding instruments. The word “Scheme” has different meanings for different funding agencies/funders. In fact, there exist other related terms such as funding program and funding action, in particular in case of a hierarchy of funding instruments. DINGO represents the generalisation of such instruments via the class FundingScheme, and expresses the taxonomy and relations among the various instruments via the Criterion class and subclasses and the FundingScheme (recursive) class properties
- **Criteria** can be of different nature, modeled in DINGO via different subclasses; multiple criteria can coexist in a single funding scheme; they provide a conceptualisation (straightforwardly extensible by sub-classing) to characterise funding policies in relation to funding schemes and activities.

We present in Fig. 1 a graphical illustration of the main parts of the ontology, both classes and properties, portrayed respectively by ovals and arrows.

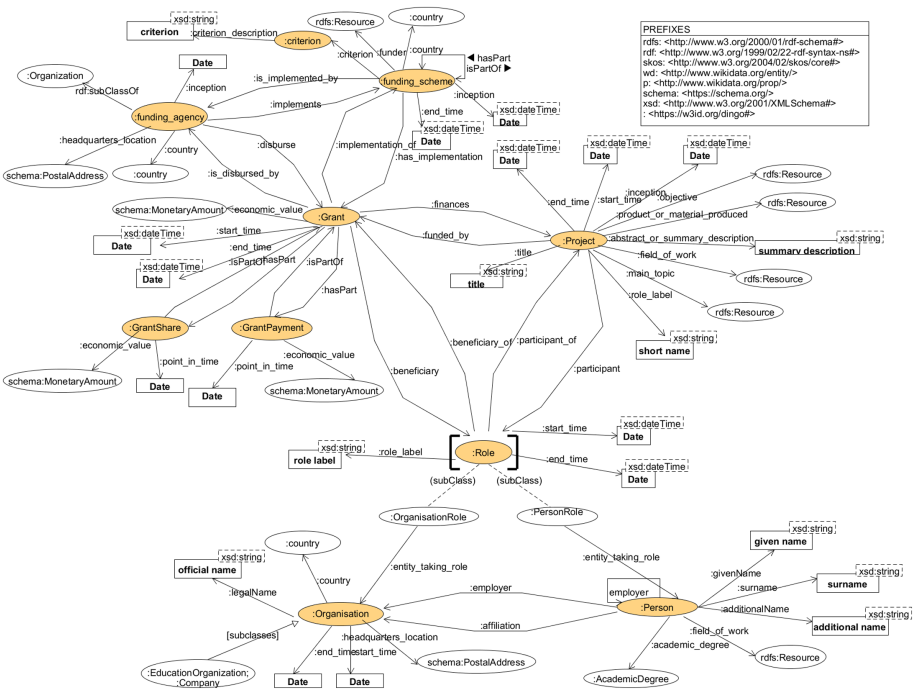


Fig. 1. Graphical representation of DINGO (main parts).

5.4 DINGO Documentation and Machine-Readable Serialisations

DINGO is documented at <https://w3id.org/dingo>. The documentation has been created using custom software written in Python (unpublished) that automatically extracts classes, properties, individuals, annotations, axioms and namespaces from OWL ontologies and produces human-readable HTML.

The machine-readable serialisation of DINGO is provided in RDF-Turtle language, and available at <https://w3id.org/dingo> by redirection when visited with the “text/turtle” header. We also provide, at the same address, a Shape Expression [35] data model for validation of data triples.

6 Maintenance and Evolution of DINGO

DINGO’s maintenance is continuous and evolutive in nature, because DINGO aims at effectively modeling funding and research practices, which continuously evolve by themselves. As mentioned, the evolution and extension of DINGO will be eased by the specific design choices made in creating it, which provide for a high modeling power to cope with the variety of existing funding realities. Hence, in many cases the required evolution/extension will be minimal (just by subclassing for new concepts).

DINGO can however be straightforwardly extended even in more orthogonal directions. For example, as discussed in Sect. 5.3, DINGO focuses on the main definition of “funding” (the monetary one, see the Cambridge, Oxford and Collins dictionaries [6, 8, 27]), but it can be extended to non-monetary funding simply by providing parallel classes as Grant, with properties for the specific resources provisions (and possibly a generalisation class to describe their commonalities).

7 Summary and Outlook

We have presented an OWL-based ontology for research and funding called DINGO and illustrated its main features, uptake and evolutive maintenance.

DINGO has the potential to constitute a key ingredient for a set of orthogonal and interoperable ontologies for the knowledge area of funding, research and their impact. In particular, there is a lack of ontological conceptualisations concerning the domain of impact and impact studies, hence, for instance, we have already planned the development of ontologies for data relative to impact indicators.

Moreover, as we mentioned, DINGO has features that enable it to be both used for domain knowledge graphs specific to research, as well as in graphs for other domains where funding aspects and policies are of interest (such as the arts, cultural conservation, and the like).

DINGO has already been used in a number of projects, as described in Sect. 3. We plan to engage further with relevant communities to create systems that offer information on research funding in distributed manner using DINGO. This should eventually lead to a truly global Open Research Information Graph providing access to data in several interconnected research information systems.

Acknowledgements. We would like to thank the co-organisers and the participants of the workshop “Wikidata for research”, Berlin, 17–18 June 2018 for their feedback and input.







References

1. Alistair, M., Sean, B.: SKOS simple knowledge organization system reference <https://www.w3.org/TR/skos-reference/>
2. Auer, M., Tschurtschenthaler, T., Biffl, S.: A flyweight UML modelling tool for software development in heterogeneous environments. In: Proceedings of EUROMI-CRO 2003 (2003). <https://doi.org/10.5555/942796.943259>
3. Besselaar, P.V.D., Khalili, A., Harmelen, F.V., de Graaf, K.: Improving social research using heterogeneous data. The SMS platform (2017). https://ic2s2.org/2017/elements/accepted_posters.html
4. BIBFRAME. <http://www.loc.gov/bibframe/docs/index.html>
5. Bibliotek-o. <https://bibliotek-o.org/>
6. Cambridge. <https://dictionary.cambridge.org/dictionary/english/funding>
7. CASRAI. <https://dictionary.casrai.org/>
8. Collins. <https://www.collinsdictionary.com/dictionary/english/funding>
9. Conlon, M., et al.: VIVO: a system for research discovery (2019). <https://doi.org/10.21105/joss.01182>
10. Crossref. <https://www.crossref.org/>
11. DCMI. <http://dublincore.org/documents/dcmi-terms/>
12. European Research Council (ERC) and Executive Agency (ERCEA), the European Union funding organisation for frontier research across all fields. <https://erc.europa.eu/>
13. Fernandez-Lopez, M., Gomez-Perez, A., Juristo, N.: METHONTOLOGY: from ontological art towards ontological engineering. In: Proceedings of the AAAI97 Spring Symposium, Stanford, USA, pp. 33–40, March 1997
14. FRAPO. <https://sparontologies.github.io/frapo/current/frapo.html>
15. Groza, T., Handschuh, S., Moller, K., Decker, S.: SALT - semantically annotated LaTeX for scientific publications. In: Proceedings of ESWC 2007, pp. 518–532 (2007). https://doi.org/10.1007/978-3-540-72667-8_37
16. Guarino, N.: Semantic matching: formal ontological distinctions for information organization, extraction, and integration. In: Pazienza, M.T. (ed.) SCIE 1997. LNCS, vol. 1299, pp. 139–170. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-63438-X_8
17. Guarino, N.: Some ontological principles for designing upper level lexical resources. In: Proceedings of the First International Conference on Lexical Resources and Evaluation, Granada, Spain, 28–30 May 1998 (1998)
18. Hammond, T., Pasin, M.: The nature.com ontologies portal. In: Proceedings of LISC 2015 (2015). <http://ceur-ws.org/Vol-1572/paper2.pdf>
19. Hammond, T., Pasin, M., Theodoridis, E.: Data integration and disintegration: managing Springer Nature SciGraph with SHACL and OWL. In: Proceedings of the Posters, Demos & Industry Tracks of ISWC 2017 (2017). <http://ceur-ws.org/Vol-1963/paper493.pdf>
20. Hu, Y., Janowicz, K., McKenzie, G., Sengupta, K., Hitzler, P.: A linked-data-driven and semantically-enabled journal portal for scientometrics. In: Alani, H., et al. (eds.) ISWC 2013. LNCS, vol. 8219, pp. 114–129. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41338-4_8

21. IG, R.: Open Science Graphs for FAIR Data. <https://www.rd-alliance.org/open-science-graphs-fair-data-ig>
22. Jörg, B.: CERIF: the common European research information format model. *Data Sci. J.* 9 (2010). <https://doi.org/10.2481/dsj.CRIS4>
23. Manghi, P., et al.: The OpenAIRE data infrastructure services: on interlinking European institutional repositories, dataset archives, and CRIS systems (2012). <https://www.openaire.eu/>
24. Manghi, P., et al.: The OpenAIRE research graph data model. <https://doi.org/10.5281/zenodo.2643199>
25. Nuzzolese, A.G., Gentile, A.L., Presutti, V., Gangemi, A.: Conference linked data: the scholarlydata project. In: Groth, P., et al. (eds.) ISWC 2016. LNCS, vol. 9982, pp. 150–158. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46547-0_16
26. OEG-UPM. <http://mayor2.dia.fi.upm.es/oeg-upm/index.php/en/ontologies/81-research-proj-ontologies/index.html>
27. Oxford. www.oxfordlearnersdictionaries.com/definition/english/funding
28. Peroni, S.: A simplified agile methodology for ontology development. In: Dragoni, M., Poveda-Villalón, M., Jimenez-Ruiz, E. (eds.) OWLED/ORE -2016. LNCS, vol. 10161, pp. 55–69. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-54627-8_5
29. Peroni, S., Shotton, D.: The SPAR ontologies. In: Vrandečić, D., et al. (eds.) ISWC 2018. LNCS, vol. 11137, pp. 119–136. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00668-6_8
30. Peroni, S., Shotton, D.: OpenCitations, an infrastructure organization for open scholarship. *Quant. Sci. Stud.* 1 (2020). <https://doi.org/10.1162/qss.a.00023>
31. Presutti, V., Gangemi, A.: Content ontology design patterns as practical building blocks for web ontologies. In: Li, Q., Spaccapietra, S., Yu, E., Olivé, A. (eds.) ER 2008. LNCS, vol. 5231, pp. 128–141. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-87877-3_11
32. Reich, J.: Ontological design patterns for the integration of molecular biological information. In: Proceedings of the German Conference on Bioinformatics GCB 1999, pp. 156–166 (1999)
33. Schema.org. <https://schema.org/>
34. SCIGRAPH. <http://scigraph.springernature.com/explorer/datasets/ontology/>
35. Shape-Expressions. <https://shex.io/>
36. Shotton, D.: Semantic publishing: the coming revolution in scientific journal publishing. *Learned Publishing* 22(2), 85–94 (2009)
37. SWC. <http://data.semanticweb.org/ns/swc/ontology>
38. Uschold, M.: Creating, integrating and maintaining local and global ontologies. In: Proceedings of the First Workshop on Ontology Learning (OL-2000) (2000)
39. Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J., Appleton, G., et al.: The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data* 3 (2016). <https://doi.org/10.1038/sdata.2016.18>



Open Science Graphs Must Interoperate!

Amir Aryani¹ , Martin Fenner² , Paolo Manghi³ , Andrea Mannocci³  ,
and Markus Stocker^{4,5} 

¹ Swinburne University of Technology, Hawthorn, VIC, Australia

² DataCite e.V., Hannover, Germany

³ Institute of Information Science and Technology – CNR, Pisa, Italy
andrea.mannocci@isti.cnr.it

⁴ TIB Leibniz Information Centre for Science and Technology, Hannover, Germany

⁵ MARUM Center for Marine Environmental Sciences, PANGAEA Data Publisher
for Earth and Environmental Science, Bremen, Germany

Abstract. Open Science Graphs (OSGs) are Scientific Knowledge Graphs whose intent is to improve the overall FAIRness of science, by enabling open access to graph representations of metadata about people, artefacts, institutions involved in the research lifecycle, as well as the relationships between these entities, in order to support stakeholder needs, such as discovery, reuse, reproducibility, statistics, trends, monitoring, impact, validation, and assessment. The represented information may span across entities such as research artefacts (e.g. publications, data, software, samples, instruments) and items of their content (e.g. statistical hypothesis tests reported in publications), research organisations, researchers, services, projects, and funders. OSGs include relationships between such entities and sometimes formalised (semantic) concepts characterising them, such as machine-readable concept descriptions for advanced discoverability, interoperability, and reuse. OSGs are generally valuable individually, but would greatly benefit from information exchange across their collections, thereby improving their efficacy to serve stakeholder needs. They could, therefore, reuse and exploit the data aggregation and added value that characterise each OSG, decentralising the effort and capitalising on synergies, as no one-size-fits-all solution exists. The RDA IG on Open Science Graphs for FAIR Data is investigating the motivation and challenges underpinning the realisation of an Interoperability Framework for OSGs. This work describes the key motivations for *i*) the definition of a classification for OSGs to compare their features, identify commonalities and differences, and added value and for *ii*) the definition of an Interoperability Framework, specifically an information model and APIs that enable a seamless exchange of information across graphs.

Keywords: Open science · Research · Knowledge graph · Interoperability

1 Introduction

The Open Science movement is urging scientists, communities, institutions, and policymakers to define and adopt methodologies, practices, and tools for open publishing research artefacts beyond the scientific article, including research data, software, and digital experiments. As a consequence of this trend, researchers are increasingly depositing these artefacts and metadata about them, together with relationships among artefacts and other relevant contextual entities such as those described, in metadata registries about authors (e.g. ORCID¹), organisations (e.g. ROR², GRID³), and data repositories (e.g. re3data⁴). De facto, Open Science publishing practices materialise a global and decentralised Open Science Graph.

Naturally, there is great interest to contribute to and/or consume such a graph for discovering and reusing artefacts as well as monitoring Open Science. To address this interest, several initiatives are building specialised Open Science Graphs (OSG), capable of serving specific user needs: Google Scholar, Scopus [3], Web of Science [4], Microsoft Academic Graph [17], FREYA PID Graph [7], Research Graph Foundation [1], OpenAIRE Research Graph [12], Open Research Knowledge Graph [11], Schol Explorer [5], Human Brain Project Knowledge Graph⁵, Open Citations [14], Crossref [9], SciGraph⁶, Semantic Scholar [8], Dimensions [10], as well as the CERIF⁷ graphs built via Current Research Information System (CRIS) are just a few existing OSGs.

The fragmentation of these specialised OSGs motivates our interest to provide OSGs with an Interoperability Framework, whose drivers are manifold. First, interoperability would reduce duplication of effort and capitalise on synergies and complementarity. Second, interoperability enables information to circulate and thus ensures the enrichment and quality of individual OSGs as well as more redundancy to safeguard information availability and persistence. Third, interoperability will elevate OSGs as the backbone of Open Science scholarly communication.

The Research Data Alliance (RDA) Interest Group (IG) on Open Science Graphs for FAIR Data⁸ is currently investigating the motivation and challenges underpinning the realisation of an Interoperability Framework for OSGs. The work presented here describes the motivations and challenges underlying the goal of an OSG Interoperability Framework, identified as:

- Need to define a classification for OSGs that supports assessing their value, compare their features, and identify differences. To this end, the presented

¹ ORCID, <https://orcid.org>.

² ROR, <https://ror.org>.

³ GRID, <https://grid.ac>.

⁴ Re3data, <http://re3data.org>.

⁵ The Human Brain Project, <https://www.humanbrainproject.eu>.

⁶ Springer Nature SciGraph, <https://scigraph.springernature.com>.

⁷ CERIF, <https://www.eurocris.org/cerif/main-features-cerif>.

⁸ RDA Interest Group on Open Science Graphs for FAIR Data, <https://rd-alliance.org/groups/open-science-graphs-fair-data-ig>.

preliminary analysis of the FREYA PID Graph, OpenAIRE Research Graph, Open Knowledge Research Graph, Research Graph, and Scholexplorer paves the way for a classification of OSGs.

- Need to define an agreed-upon information model and APIs that enable the seamless exchange of information across OSGs.

The results of our preliminary investigation suggest that there is a need for a community-driven initiative that ensures common terminology (i.e. classification) and interoperability-enabled added value scholarly communication services that exploit the full potential of OSGs.

2 A Classification for Open Science Graphs

The fabric required to enact Open Science is a digital infrastructure based on an Interoperability Framework that captures research artefacts (in particular articles, datasets, software, services, workflows), metadata about artefacts, people and institutions as well as their relationships, as they evolve over time. This infrastructure relies on the adoption of Persistent Identifiers (PIDs) and metadata standards for the persistent identification and description of such entities across data sources (e.g. repositories, archives), thematic services (e.g. research infrastructures), and research communities.

Open Science Graphs (OSGs) are use case driven specialisations of Scientific Knowledge Graphs that build on the fabric of PIDs, metadata, and relationships. Figure 1 depicts OSGs in their context. Their scope differs according to served stakeholders, whose needs range from discovery, access, and reuse of research artefacts to monitoring and evaluating funding efforts, and identifying research trends. Stakeholder needs also drive the selection of data sources a particular OSG ought to integrate and the required data processing and enrichment capabilities.

The increasingly diverse and complex OSG landscape fuels an urgent demand for a classification to *i)* facilitate service providers in building needed added value services on OSGs, *ii)* assist consumers in selecting the services that meet their needs, and *iii)* facilitate OSG providers in identifying and communicating the characteristics of their service, and in understanding how to benefit from other OSGs.

In a first attempt to develop a classification framework that supports the systematic description of OSG characteristics and identification of their commonalities and differences, in the following, we introduce some existing OSGs that have been developed in recent years, namely: FREYA PID Graph, OpenAIRE Research Graph, Open Knowledge Research Graph, Research Graph, and Scholexplorer. This selection is by no means exhaustive. Indeed, additional initiatives do exist, e.g. Microsoft Academic Graph, SciGraph, Crossref, Dimensions, Semantic Scholar, Open Citations. Still, we argue that the selected OSGs reasonably represent the broader landscape.

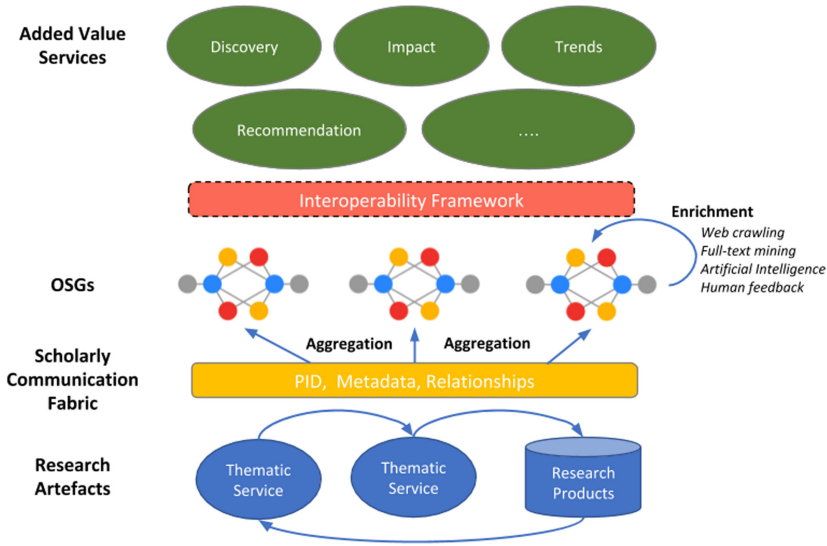


Fig. 1. OSGs and scholarly communication architecture

2.1 Existing OSGs

FREYA PID Graph. The PID Graph is a scholarly infrastructure built by the partners in the EC-funded FREYA project, with the core infrastructure hosted by DataCite⁹. PID Graph identifies all nodes in the graph using persistent identifiers (PIDs) and describes these nodes, as well as the edges between nodes using the metadata associated with these PIDs. The graph is a federated graph, with PIDs and associated metadata provided by a number of PID providers who store this information in their respective services.

The development of the PID Graph is driven by user stories that the FREYA project partners have initially identified and that are continuously expanded. A distinguishing feature of these user stories is that they cannot be easily resolved with existing scholarly infrastructure, as they assume an underlying graph. Many of these user stories are around the discovery of connected resources, and the tracking of reuse.

The research entities supported by the PID Graph currently include publications, datasets, software, physical samples, instruments, services, people, research organisations, funders, and research data repository registries from the PID providers Crossref, DataCite, ORCID and ROR, for a total of currently about 35 million resources.

The PID Graph uses GraphQL¹⁰ to query the PID Graph¹¹, a widely used open source technology that aims to make it easy to build client applications

⁹ DataCite, <https://datacite.org>.

¹⁰ GraphQL, <https://graphql.org>.

¹¹ DataCite API, <https://api.datacite.org/graphql>.

for the PID Graph. The fields that describe resources have been harmonised across resource types to simplify working with the PID Graph and to enhance connections between resources. Many Jupyter notebooks have been written to explore the PID Graph, and they are openly available for reuse. All information in the PID Graph is available for reuse without restrictions, the software stack powering the PID Graph is available as open source software.

OpenAIRE Research Graph. The mission of the OpenAIRE initiative¹², one of the foundations of the European Open Science Cloud (EOSC)¹³, is to provide training, dissemination, and technical services to seed (and support) Open Science publishing practices into the research lifecycle. To this end, one key activity of OpenAIRE aims at the construction of the OpenAIRE Open Research Graph by aggregating and integrating metadata records relative to digital research products (literature, dataset, software, and others) from more than 13,000 scholarly data sources world-wide (scientific repositories, archives, registries, databases, publishers), for a current total of more than 114 million publications, and 10 million research data. The graph is also algorithmically enhanced so to *i*) find and merge metadata records that describe the same entity (literature, and organisations), and *ii*) apply inference techniques on the metadata records and mine full-texts of Open Access publications to add new properties and new semantic relationships. End-user claims provided via the Web portal are also fed in the loop, so to drive the processing of raw metadata.

The OpenAIRE Research Graph data model is described in detail in [13] and its modelled entities are: literature, datasets, software, funders, funding streams, grants, organisations, researchers, data sources. Its content supports a number of analytics and applications such as discovery, research impact assessment, Open Science monitoring, brokering, reporting to funders, and statistics.

The graph is redistributed free of charge for everyone to use¹⁴, both in bulk access mode (snapshot dump [12], OAI-PMH), and in selective access mode via APIs (REST Search API, LOD) under CC-BY licence, due to the fact the graph integrates sources with licences stricter than CC0 (e.g. Microsoft Academic Graph, Springer Nature).

Open Knowledge Research Graph. The Open Research Knowledge Graph¹⁵ (ORKG) [11] is a scholarly infrastructure and open project led by the TIB Leibniz Information Centre for Science and Technology that aims to publish scholarly knowledge communicated in the literature in structured and semantic form.

The entity of primary interest to ORKG is therefore the research article (paper) and, importantly, article content. ORKG models article contents as “research contribution”, an abstract concept that, in general terms, relates the problem addressed by a contribution with the materials and methods used, and the obtained results.

¹² The OpenAIRE project, <https://www.openaire.eu>.

¹³ EOSC, <https://www.eosc-portal.eu>.

¹⁴ Access to the OpenAIRE Research Graph, <http://develop.openaire.eu>.

¹⁵ ORKG, <http://orkg.org>.

ORKG enables a range of new applications, including automated comparisons. As a classic example, it is possible to automatically compare the characteristics of sorting algorithms, e.g. their best and worst-case complexity. Given precision, recall and F1 score of classification algorithms across the literature on a specific problem, say road-vehicle detection, it is possible to create leaderboards automatically, showing the trend of classification performance over time and the currently leading approach.

The primary data sources for ORKG are peer-reviewed research articles. In case data published in the literature (e.g. as a plot) is deposited in a research data repository, such infrastructures are an additional important data source. Furthermore, ORKG relies on third-party terminologies to align resources and thus ensure data interoperability and reusability.

ORKG adds value by making scholarly knowledge published in the literature better accessible to and processable by machines. As a multimodal infrastructure, ORKG integrates diverse data (i.e. scholarly knowledge) acquisition forms, specifically manual crowdsourcing, automated text mining, and scholarly knowledge exchange among research infrastructure, services and tools, e.g. data analysis environments such as Jupyter.

ORKG data export and provision is primarily via its REST API and SPARQL endpoint. Supported data formats are JSON and various RDF serialisations. As an open project and infrastructure, ORKG publishes software and data under open licenses (MIT and CC BY-SA, respectively).

Research Graph. Research Graph is a distributed network of scholarly works including data from data repositories such as NCI in Australia [15], academic and grey literature (e.g. GESIS, ICPSR), grants and funders (e.g. Australian Research Council, NIH) and researchers and research organisation information. Research Graph initially formed as a Graph Database by participants in the DDRI Working Group of Research Data Alliance [2] to connect datasets and metadata about data collections across repositories and data infrastructures. This graph later extended to a distributed network of graphs connecting via graph augmentation functionality running on a hybrid (national, private and commercial) cloud. At the time of writing this article, the graph holds close to 250 million nodes, including metadata about 180 million publications, 51 million datasets, 55 thousand grants, 1.4 million organisations, and 8.6 million researchers.

Research Graph is accessible to the partner organisations via Augment API, that is a cloud-hosted capability which creates graphs from bibliographic records, and extends this graph using information available in Research Graph clusters. The schema used for this transformation is based on the minimum required fields for identifying a research object, a trade-off between completeness and practicality, in favour of practicality. The graph schema [1] supports both XML, RDF XML and JSON-LD [16], and the endpoint supports Cloud Hosted Services, REST API and GraphQL.

Research Graph is mainly used by data infrastructures, repositories, and research systems for discovery of related scholarly works such as related datasets,

and connections between grants and research output. Metadata about Research Graph is available on researchgraph.org, and github.com/researchgraph/schema, the input API supports RDF, DDI, RIF-CS, Dublin Core, Scholix, DataCite, Crossref, and many other metadata formats, and the output includes Research Graph Schema (JSON, XML), JSON-LD and RDF XML. Research Graph includes a subgraph reusable under CC-By licence while some other parts are accessible for limited use only under NC-ND-SA-CreativeCommons.

Scholexplorer. Scholexplorer¹⁶ [5] populates and provides access to a graph of Scholix [6] links between dataset and literature objects, and between dataset and dataset objects. Links (and objects) are provided by data sources managed by publishers, data centres, or other organisations providing services to store and manage links between data sets and publications such as CrossRef, DataCite, PubMed, EMBL-EBI data sources, Pangaea, and OpenAIRE. Scholexplorer aggregates links metadata harvested from these data sources as Scholix records and out of these builds a harmonised and de-duplicated graph of scholarly objects counting today over 21 million publications, 53 million datasets, and over 269 million bi-directional semantic links between them. The graph is openly accessible under CC-BY licence via REST search APIs that return links in Scholix format, and via periodic dumps on Zenodo¹⁷.

2.2 Classification

Based on a comparison of the five initiatives described above (Fig. 2), we propose a first classification across seven main features, regarded as relevant to both OSG consumers and OSG providers.

1. **Research Entities:** Each OSG operates at a specific level of abstraction, and consequently models (including PIDs and metadata descriptions) specific research entities of interest to the scholarly communication domain, such as research artefacts (e.g. publications, data, software, samples, instruments) and concepts therein expressed (e.g. hypothesis, methods, algorithms, protocols), research organisations, researchers, services, projects, and funders.
2. **Applications:** OSGs are designed to serve specific use cases, which may range from the discoverability of research artefacts to the assessment of research impact, from quantitative science studies (e.g. science of science) to the computation of usage statistics, and monitoring, etc.
3. **Data Sources:** OSGs are constructed by collecting and integrating information from different types of data sources (e.g. journals, repositories, archives, registries, other OSGs, etc.), or selections of such sources (e.g. thematic or discipline-driven, from a geographical area, related to organisations), to specific individual sources.
4. **Added Value:** OSGs are often characterised by integration and enrichment techniques, which manipulate the aggregated metadata in several ways; for

¹⁶ Scholexplorer, <http://scholexplorer.openaire.eu>.

¹⁷ Scholexplorer dump, <https://zenodo.org/record/3541646>.

example, by *i*) harmonising metadata to map it onto the OSG information model (e.g. metadata structural and semantic transformations), and *ii*) by enhancing the metadata via web crawling, interlinking, inference, full-text mining, AI, user annotations and feedback, etc.

5. **Data Export and Provision:** OSGs offer access to their content via APIs (e.g. OAI-PMH, SPARQL, GraphQL, ad-hoc REST APIs, etc.) and standard exchange formats (e.g. XML, JSON, RDF) that implement standard metadata formats (e.g. DataCite, Scholix.org, Dublin Core, ORCID profile, CERIF) or proprietary formats.
6. **FAIRness:** FAIRness of OSGs regards their nature as dataset in regard to being Findable, Accessible, Interoperable, and Reusable. Practices vary, but in general, OSGs are available via standard exchange formats (e.g. XML, JSON, RDF) and accessible via standard protocols, from simple download to GraphQL, OAI-PMH, or proprietary search REST APIs. In some cases, accessibility is facilitated by minting a DOI to the OSG collection, and, in some cases, complicated by the fact consumers need to go through toll-gated cloud services to access the graph. OSG schemata give life to the hardest interoperability and reusability challenge, as they follow application-driven interpretations of research entities, which complicate OSG reuse and integration.
7. **Openness:** Different OSGs are released and redistributed under different licences (CC0, CC-BY, CC-SA, etc.). In general, the licence applies to the whole graph, but, in some cases, different parts of the graph can be released under different licences, be accessible only to a limited number of stakeholders, or be behind a paywall. In other cases, for example for Microsoft Academic Graph, the graph is released openly with ODC-BY licence, but a small fee is needed to sustain the provisioning platform (i.e. Azure).

While the table already provides evidence for the value of a classification, it also highlights the need for common agreements on classification criteria. For example, aspects such as coverage of the data sources aggregated by the OSG may be of interest, as a graph may focus on a geographical region, be cross-community or community-specific (e.g. computer science and neuroscience in the early Semantic Scholar), or be able to capture geospatial descriptions (e.g. INSPIRE in Research Graph).

3 A Framework for Open Science Graphs Interoperability

We advocate for the establishment of a community-driven Interoperability Framework in order to mediate the diverse data models and technologies used by existing OSGs. The drivers for conceiving an Interoperability Framework for OSGs are manifold.

Firstly, as we have seen in Sect. 2, the various OSGs differ in scope, extent and technological details as they strive to capture various aspects of scholarly communication from diverse perspectives and different abstraction and granularity. Thus, the information pertaining to different OSGs can be overlapping or

can be complementary. With overlap we gain plurality, e.g. different identifiers for same papers, authors, organisations, etc. while with complementarity we gain completeness and coverage, e.g. integrate information of various granularity as published by various OSGs.

Secondly, despite building on data sources with clear sustainability plans, some OSGs have unclear directions, lack a viable business model, and thus might cease to exist. Given this risk, OSG content should be federated, shared, and possibly fed back to original data sources where it can be managed sustainably for the common good of both the scientific community and, more broadly, society.

Thirdly, OSGs and more generally Scientific Knowledge Graphs should act as the backbone of modern Open Science scholarly communication, embody its core principles, and foster its adoption along several dimensions such as discoverability, monitoring, and FAIRness. This is especially relevant for the non-commercial OSGs and their leading role in open innovation with best-in-class, cutting-edge services, free at the point of use.

It is therefore of paramount importance to exchange OSG content and capitalise on the non-negligible acquisition, integration and enrichment efforts performed by the various OSGs. To facilitate information exchange between OSGs, the Interoperability Framework may rely on an agreed-upon *lingua franca*. This was already achieved with the specification of Scholix [6], an agreed-upon high-level interoperability framework for exchanging information about the links between scholarly literature and data. However, Scholix operated within a much narrower scope. Given the complexity of the modelled information and the ambition of the endeavour, for OSGs a set of “dialects” rather than a single *lingua franca* may be more viable while still efficiently catalyse interoperability.

OSG content exchange has to occur on at least two levels of abstraction: *information model* and *technological*. In regard to information modelling, there is an urge for the various OSGs to define bottom-up a common model that can maximise information exchange and has the flexibility to accommodate unforeseen extensions, use cases, and stakeholders. From a technological standpoint, we need a portfolio of operational frameworks supporting a seamless exchange of information across different OSGs by means of operators/primitives. Doing so implies supporting a plethora of exchange formats (and the relative mappings to the common model) such as CSV, XML, RDF, JSON-LD, Scholix, and OAI-ORE, as well as different APIs enabling the provisioning of OSG information such as REST, SPARQL, and GraphQL.

To this end, we envisage the European Open Science Cloud (EOSC) as one optimal channel through which such an Interoperability Framework for OSGs could be developed via consensus and for the benefit of Open Science, at least at a pan-European level. EOSC is being constructed having a System of Systems paradigm, where local autonomy and differences are fostered as they can be an added value, and where convergence is recommended and facilitated via common interoperability frameworks to optimise cost and maximise the efficiency of science. OSGs would, in such an ecosystem, become the mean for *i*) bridging research infrastructures, i.e. thematic and scholarly communication services, and

	Research Entities	Applications	Data Sources	Added value	Data Export	FAIRness	Openness
<i>PID graph</i>	PIDs: DOI, ORCID, ROR, Crossref Funder ID Entities: publications, datasets, software, funders, research organizations, funders, research data repository registries	Discovery, Research Impact, Open Science Monitor, Brokering, Reporting to funders, Statistics	PID providers	Standard GraphQL query interface with client libraries available in many languages. Strong support for Jupyter notebooks	API: GraphQL	Findable: Graphs are searchable by PIDs and metadata with GraphQL API Accessible: access described entities via PID Interoperable: Input: DataCite XML, Crossref XML, ORCID XML, Output: GraphQL JSON Reusable: Graphs are reusable	Redistributed free of charge
<i>OpenAIRE Research Graph</i>	PIDs: DOI, ORCID, accession numbers, PMCID, URLs, MAG IDs Entities: Literature, datasets, software, funders, grants, organizations, researchers, data sources Relationships: DataCite relationships, fundedBy, similarTo, hasAuthorAffiliatedWith	Discovery, Research impact assessment, Open Science Monitoring, Brokering, Reporting to funders, Statistics	Any data source trusted by scientists, repositories, archives, registries, databases, publishers	Enrichment of metadata and relationships by full-text mining, User-feedback, Inference by context propagation, deduplication, Provenance tracking	Format: OpenAIRE XML format APIs: LOD, ONI-PMIH, Dumps, REST Search APIs	Findable: searchable on Zenodo, accessible by DOI Accessible: every entity is openly accessible via HTTP API and dump Interoperable: OpenAIRE XML, documented online Reusable: Content can be reused with CC-BY licence	Redistributed free of charge under CC-BY licence
<i>ORKG</i>	PIDs: DOI, ORCID, URLs Entities: Literature and items of its content Relationships: addressed problems, utilized materials, employed methods, yielded results.	At the granularity of items of scholarly literature content, discovery, comparison, recommendation, reuse visualization, reuse	Literature, research data repositories, terminologies, LIMS/ELN	Multimodal infrastructure for the acquisition, curation and publishing of machine-actionable scholarly knowledge	Format: JSON and RDF serializations APIs: REST API, SPARQL	Findable: ORKG search, but currently lacking findability through 3rd party systems. Plans to assign DataCite DOIs to elements of ORKG content. All resources are URL-identified. Neo4j dumps or RDF exports can be deposited in a suitable data repository. Accessible: All ORKG resources are URL-retrievable, accessible and have their own descriptive landing page. Content can be accessed programmatically via REST API. Interoperable: Data use a graph-based data model, but currently lacking formal semantics. Alignment with external terminologies is technically possible but not broadly practised. Reusable: Content is reusable under CC BY-SA licence, provenance is tracked. Neo4j dumps or RDF exports can be made available for reuse.	Data and Software are released under CC BY-SA and MIT licences, respectively.
<i>Research Graph</i>	PIDs: DOI, ORCID, PURL, ISNI, GRID, PMCID, Scopus ID Entities: academic articles and grey literature, datasets, funders, grants, organizations, researchers, datasets Relationships: authorship, funding, citation, usage, known_as, employment, custodian, management, etc.	Supporting repositories and research infrastructures	PID providers, data repositories, publishers, funders, discovery services and aggregators	Identity resolution, metadata enhancement, topic modelling, clustering, text mining and GIS mapping	Format: XML and JSON APIs: Cloud Hosted Services, REST API and GraphQL	Findable: Metadata available via researchgraph.org Accessible: Controlled Access available Interoperable: Input: RDF, DDI, RIF-CS, Dublin Core, Scholix, DataCite, Crossref, and many other metadata formats. Output: RDF and Research Graph Schema (JSON, XML) Reusable: includes subgraph reusable under CC-BY licence while some other parts only accessible for limited use under NC-ND-SA-Creative Common	Controlled Access
<i>Scholixplorer</i>	PIDs: DOI, accession numbers, PMCID, URLs Entities: Literature, datasets Relationships: citedBy, supplementedBy, references	Discovery, Statistics, PID Resolution	DataCite, CrossRef Event Data, EMBL-EBI (Scholix), OpenAIRE, SPARC-compliant data sources	Deduplication by PID, provenance tracking	Format: Scholix [?] APIs, Dumps on Zenodo, REST Search APIs	Findable: searchable on Zenodo, accessible by DOI Accessible: every entity is openly accessible via REST API Interoperable: Scholix links Reusable: Content is released under CC-BY licence	Redistributed free of charge under CC-BY licence

Fig. 2. Classification of OSGs initiatives

ii) offer to EOSC users, such as researchers, research communities, policymakers, and SMEs the tools to discover and monitor trends and impact of science. The RDA IG on Open Science Graphs for FAIR Data is and will be contributing to the definition of the EOSC interoperability frameworks to ensure that specific solutions will be sought after. Finally, another channel that is potentially conducive so to bring this discussion onto the global landscape and the long-term perspective of a broader “Global Open Science Cloud” could be the RDA IG on Global Open Research Commons¹⁸.

4 Conclusions

In this paper, we targeted two challenges of working with Open Science Graphs (OSGs). On the one hand, OSGs would benefit from a classification framework that enables their inspection and comparison along key features. On the other hand, we argue that an Interoperability Framework is pivotal to enable a seamless exchange of information among OSGs with the resulting suggested benefits.

We proposed a preliminary classification framework by analysing a selection of representative OSGs, namely: FREYA PID graphs, OpenAIRE Research Graph, Open Knowledge Research Graph, Research Graph, and Scholexplorer. Moreover, we outlined the main drivers and *desiderata* of a possible Interoperability Framework.

Going forward, we see the RDA Interest Group on Open Science Graphs for FAIR Data as an important community to make further progress on aligning the various OSG initiatives, in particular concrete work on interoperability.

Acknowledgements. This work was co-funded by the EU projects OpenAIRE-Advance (Grant agreement ID: 777541), FREYA (Grant agreement ID: 777523), ScienceGRAPH (Grant agreement ID: 819536), and the TIB Leibniz Information Centre for Science and Technology.

References

1. Aryani, A., et al.: A research graph dataset for connecting research data repositories using RD-Switchboard. *Sci. Data* **5** (2018). <https://doi.org/10.1038/sdata.2018.99>
2. Aryani, A., Wang, J.: Research graph: building a distributed graph of scholarly works using research data switchboard. In: Open Repositories Conference (2017). <https://doi.org/10.4225/03/58c696655af8a>
3. Baas, J., Schotten, M., Plume, A., Côté, G., Karimi, R.: Scopus as a curated, high-quality bibliometric data source for academic research in quantitative science studies. *Quantitative Sci. Stud.* **1**(1), 377–386 (2020). <https://doi.org/10.1162/qss.a.00019>
4. Birkle, C., Pendlebury, D.A., Schnell, J., Adams, J.: Web of Science as a data source for research on scientific and scholarly activity. *Quantitative Sci. Stud.* **1**(1), 363–376 (2020). <https://doi.org/10.1162/qss.a.00018>

¹⁸ RDA IG on Global Open Research Commons, <https://www.rd-alliance.org/groups/global-open-research-commons-ig>.

5. Burton, A., et al.: The data-literature interlinking service: towards a common infrastructure for sharing data-article links. *Program* **51**(1), 75–100 (2017). <https://doi.org/10.1108/PROG-06-2016-0048>
6. Burton, A., et al.: The scholix framework for interoperability in data-literature information exchange. *D-Lib Magazine* **23**(1/2) (2017). <https://doi.org/10.1045/january2017-burton>
7. Fenner, M., Aryani, A.: Introducing the pid graph (2019). <https://doi.org/10.5438/jwvf-8a66>
8. Fricke, S.: Semantic scholar. *J. Med. Library Assoc.* **106**(1), 145–147 (2018). <https://doi.org/10.5195/jmla.2018.280>
9. Hendricks, G., Tkaczyk, D., Lin, J., Feeney, P.: Crossref: the sustainable source of community-owned scholarly metadata. *Quantitative Sci. Stud.* **1**(1), 414–427 (2020). <https://doi.org/10.1162/qss.a.00022>
10. Herzog, C., Hook, D., Konkiel, S.: Dimensions: bringing down barriers between scientometricians and data. *Quantitative Sci. Stud.* **1**(1), 387–395 (2020). <https://doi.org/10.1162/qss.a.00020>
11. Jaradeh, M.Y., et al.: Open research knowledge graph: next generation infrastructure for semantic scholarly knowledge. In: *Proceedings of the 10th International Conference on Knowledge Capture*, pp. 243–246. K-CAP 2019, Association for Computing Machinery, New York (2019). <https://doi.org/10.1145/3360901.3364435>
12. Manghi, P., et al.: Openaire research graph dump (2019). <https://doi.org/10.5281/zenodo.3516918>
13. Manghi, P., et al.: The openaire research graph data model (2019). <https://doi.org/10.5281/zenodo.2643199>
14. Peroni, S., Shotton, D.: OpenCitations, an infrastructure organization for open scholarship. *Quantitative Sci. Stud.* **1**(1), 428–444 (2020). <https://doi.org/10.1162/qss.a.00023>
15. Wang, J., Aryani, A., Evans, B., Barlow, M., Wyborn, L.: Graph connections made by RD-switchboard using nci’s metadata. *D-Lib Magazine* **23**(1/2) (2017). <https://doi.org/10.1045/january2017-aryani>
16. Wang, J., Aryani, A., Wyborn, L., Evans, B.: Providing research graph data in JSON-LD using schema.org. In: *Proceedings of the 26th International Conference on World Wide Web Companion*, pp. 1213–1218 (2017). <https://doi.org/10.1145/3041021.3053052>
17. Wang, K., Shen, Z., Huang, C., Wu, C.H., Dong, Y., Kanakia, A.: Microsoft academic graph: when experts are not enough. *Quantitative Sci. Stud.* **1**(1), 396–413 (2020). <https://doi.org/10.1162/qss.a.00021>



WikiCSSH: Extracting Computer Science Subject Headings from Wikipedia

Kanyao Han, Pingjing Yang, Shubhanshu Mishra^(✉), and Jana Diesner

School of Information Sciences, University of Illinois at Urbana-Champaign,
Champaign, IL 61820, USA

{kanyaoh2,py2,jdiesner}@illinois.edu, mishra@shubhanshu.com

Abstract. Domain-specific classification schemas (or subject heading vocabularies) are often used to identify, classify, and disambiguate concepts that occur in scholarly articles. In this work, we develop, apply, and evaluate a human-in-the-loop workflow that first extracts an initial category tree from crowd-sourced Wikipedia data, and then combines community detection, machine learning, and hand-crafted heuristics or rules to prune the initial tree. This work resulted in WikiCSSH; a large-scale, hierarchically-organized subject heading vocabulary for the domain of computer science (CS). Our evaluation suggests that WikiCSSH outperforms alternative CS vocabularies in terms of coverage of CS terms that occur in research articles. WikiCSSH can further distinguish between coarse-grained versus fine-grained CS concepts. The outlined workflow can serve as a template for building hierarchically-organized subject heading vocabularies for other domains that are covered in Wikipedia.

Keywords: Hierarchical vocabulary · Wikipedia · Computer science

1 Introduction

A scholarly publication can be considered as a collection of concepts. Identifying these concepts allows us to build better search interfaces¹, study temporal trends in the evolution of concept usage [13, 15, 17], compute conceptual expertise of authors [11], and study citation patterns in scholarly data [8, 12], among other practical applications. For many domains, e.g., biomedicine, mathematics, and physics, well curated, controlled, and structured vocabularies have been developed, which are commonly referred to as subject heading vocabularies (or simply, subject headings). These subject headings index relevant concepts in a domain, and organize these concepts into a hierarchical structure (e.g., concepts

¹ <https://www.nlm.nih.gov/bsd/pubmed.html>.

This material is based upon work supported by the Korea Institute of Science and Technology Information under Grant No. C17031. We thank Kehan Li for assistance with data annotation, and anonymous reviewers for their feedback.

and sub-concepts), which facilitates coarse-grained and fine-grained knowledge organization that represents the breadth and depth of a field. Examples of prominent subject headings are Medical Subject Headings (MeSH)², Physics subject headings (PhySH)³, and Mathematics Subject Classification (MSC)⁴.

In the domain of computer science (CS), a commonly used classification schema is the ACM Computing Classification System (ACM CCS)⁵. While this vocabulary has been curated by CS domain experts, it is being updated more slowly than the field advances, and is comparatively small-scale: The latest version of ACM CCS was released in 2012 (and its predecessor in 1998) and contains about 2,000 subject headings, while MeSH is updated once a year and contains 25,000 subject headings. Furthermore, the ACM CCS schema contains coarse-grained concepts that are helpful for identifying and categorizing relatively broad research areas of computing, but is not designed to also capture concrete, fine-grained concepts. To remedy these shortcomings, recently, the Computer Science Ontology (CSO) [19] has been introduced as an automatically constructed ontology. CSO was extracted from scholarly papers, contains about 22,000 subject headings and semantic relations between them, and can help to identify both broad and narrow research areas of computing. However, CSO fails to distinguish between core CS concepts versus concepts related to CS that emerge at the nexus of CS and other fields through interdisciplinary work. In this paper, we refer to these related concepts as ancillary CS concepts. Examples of ancillary CS concepts include “gender” and “aircraft”. Another strength of CSO is that it links each concept to multiple knowledge bases, including Wikidata and Freebase. However, CSO does not yet leverage the vast amount of human effort used to organizing knowledge in Wikipedia. To address the outlined limitations, we herein report on the extraction of a large-scale, hierarchical, and semi-curated CS vocabulary that distinguishes between coarse-grained and fine-grained concepts as well as between core and ancillary CS concepts, while being grounded in knowledge provided by many people over time in the form of the Wikipedia Category Tree (WCT)⁶. We refer to our resulting vocabulary as *Wikipedia-based Computer Science subject headings (WikiCSSH)*⁷ [5]. WikiCSSH was created with a mixed methods approach to extracting CS-relevant subject headings, which included breadth first search in the WCT; followed by manual filtering, community detection, embedding-based classification, and human-created rules for removing false positives. Finally, the construction of WikiCSSH benefited from the automatic association of Wikipedia pages with Wikipedia categories, which we used for the automatic expansion of WikiCSSH to include pages affiliated with Wikipedia categories into WikiCSSH.

² https://www.nlm.nih.gov/mesh/concept_structure.html.

³ <https://physh.aps.org/>.

⁴ <https://mathscinet.ams.org/msc/msc2010.html>.

⁵ <https://dl.acm.org/ccs>.

⁶ <https://en.wikipedia.org/wiki/Special:CategoryTree>.

⁷ <https://github.com/uiuc-ischool-scanr/WikiCSSH>.

Our project makes two main contributions. First, we provide a large hierarchical subject headings schema for CS with more than 700,000 CS concepts that are divided into core and ancillary concepts. Second, our work shows how to leverage the Wikipedia Category Tree for this purpose. This methodology might serve as a template for the construction of vocabularies for other domains for which information is available from Wikipedia. This paper illustrates the challenges resulting from using Wikipedia data for this specific task, shows solutions to these challenges, and implements a workflow with human-in-the-loop processes to overcome some of these hurdles.

2 Related Work

Various domains have developed their own hierarchical, domain-specific vocabularies, such as MeSH for biomedicine. MeSH is particularly useful for practical applications due to its hierarchical and non-cyclical nature. Furthermore, MeSH, along with MEDLINE, an annotated biomedical corpus, can be used to track the evolution of biomedical concepts over time and create concept profiles of authors [13, 15, 17]. The fields of mathematics and physics also have developed domain-specific vocabularies, namely, Mathematics Subject Classification (MSC) and Physics subject headings (PhySH). Finally, there exists the Wikipedia Category Tree (WCT), which covers a large number of domains and is used to classify Wikipedia articles. WCT is curated by the Wikipedia community. For CS, ACM CSS [16] and CSO [19] are the two prominent controlled vocabularies. A comparison of various domain-specific and cross-domain controlled vocabularies is shown in Table 1.

Table 1. Comparison of existing controlled vocabularies for various domains.

Name	Type	Size	Curation	Domain
MeSH	Fine grained	25K	National Library of Medicine	Biomedicine
PhySH	Fine grained	3.5K	American Physical Society	Physics
PACS	Subject level	9.1K	American Institute of Physics	Physics
MCS	Subject level	6.1K	Mathematical Reviews and Zentralblatt MATH	Mathematics
CCS	Subject level	2K	Association of Computer Machinery	Computer science
WCT	Fine grained	1M+	Wikipedia contributors	Open domain

While expert-constructed vocabularies often trade off size for quality and accuracy, automatically generated vocabularies often flip this relationship. Constructing vocabularies from structured, crowd-sourced data has become another viable approach [6, 7, 9, 20, 21]. For example, prior research has leveraged Wikipedia as a comprehensive knowledge base [9, 20], e.g., for building multilingual DBpedia [7] and temporal YAGO2 [6, 21]. Since Wikipedia and the referenced related projects are not domain-specific to CS, we herein aim to leverage

Wikipedia to develop a methodology for building a domain-specific, hierarchical, and non-cyclical vocabulary that distinguishes between coarse-grained and fine-grained concepts as well as between core and ancillary CS concepts.

3 Methods

3.1 Wikipedia Category Tree

The Wikipedia Category Tree (WCT) consists of 1.6 M categories with 10.9 M inter-category links and 217.6 M category-page links. Each category in the WCT can have multiple parents as well as multiple children. Links between categories are referred as parent-child links. Each category has multiple affiliated pages. We assume that pages affiliated with a category refer to concrete concepts within that category. In other words, a category is a coarse-grained term that refers to a relatively broad research area or topic, while a page is a fine-grained term that refers to a concrete, fine-grained concept within a category. It is important to note that WCT is not necessarily a tree as it contains circular paths, e.g., *Mathematics* → *Philosophy of mathematics* → *Formalism* → *Formal sciences* → *Mathematics* → *Philosophy of mathematics*. Furthermore, since WCT is crowd-sourced and open-domain, it contains many parent-child relationships which are not relevant for our task of identifying categories relevant to CS research concepts. For example, in the parent-child chain *Computing and society* → *Social media* → *Fiction about social media*, the category *Computing and society* is relevant to CS in our context, but the category *Fiction about social media* is not. Furthermore, *Fiction about social media* leads to additional irrelevant categories (such as *Novels about social media*), and this pattern is recursive.

3.2 Building an Initial CS Domain-Specific Subtree

To construct CS specific subject headings schema, we started by extracting an initial CS subtree (ICS) as described next (see Algorithm 1). The following categories were chosen as starting points because they represent five highest-level domains relevant to CS: *computer science*, *information science*, *computer engineering*, *statistics*, and *mathematics*. These five categories constitute the first level of our initial CS subtree, and determine the overall breadth of our vocabulary. We recursively updated ICS with all children of the categories in the current ICS using a breadth first search over WCT. Redundant categories were removed during this search since we removed all categories based on exact matches of phrases that have occurred before. This resulted in an ICS with more than 1.4 million categories, which were organized in 20 levels (depth of ICS). Overall, the extraction process performed in this first step has resulted in high recall but low precision for CS-relevant categories.

3.3 Removing False Positives from the ICS

Our manual inspection of this ICS revealed a few major issues.

Algorithm 1: Building WikiCSSH

```

input : WCT, ICS  $\leftarrow$  Initial Categories, rules
output: WikiCSSH
1 new_cats  $\leftarrow$  ICS
2 while new_cats  $\neq \emptyset$  do
3   | categories  $\leftarrow$  Children(new_cats)
4   | new_cats  $\leftarrow$  categories - ICS
5   | ICS  $\leftarrow$  ICS  $\cup$  new_cats
6 end
7 ICS  $\leftarrow$  Filter(ICS, manual)
8 communities  $\leftarrow$  FindCommunities(ICS)
9 ICS  $\leftarrow$  Filter(ICS, communities)
10 models  $\leftarrow$  TrainModels(ICS)
11 ICS  $\leftarrow$  Filter(ICS, models)
12 ICS  $\leftarrow$  Filter(ICS, rules)
13 WikiCSSH  $\leftarrow$  ExtractPages(ICS)

```

First, as described above, we identified many categories that were not related to the domain of CS and should therefore be removed from a useful CS subject headings schema. These categories often appear in lower levels of our tree, where the inclusion of even a single irrelevant category can lead to the inclusion of a large number of that category’s irrelevant children. Second, while some categories were related to CS, a few of them were not useful for our intended use. These included names of CS conferences, researchers, and CS research/teaching institutes. We consider the above two issues as cases of false positives and describe our approach for removing those in Algorithm 1. It is important to note that here, false positives and irrelevant categories are meant in reference to our purpose, i.e., building a structured vocabulary of subject headings relevant for indexing research in CS, not noise or irrelevance in Wikipedia itself. We fully acknowledge that any of the instances that we did not include in WikiCSSH might very well be excellent categories for other contexts and applications.

3.3.1 Manual Annotation for First Three Levels

The first three levels of the ICS contained a variety of broad, important sub-domains that are relevant to CS, such as *artificial intelligence* and *algorithms and data structures*. Considering that any false negatives and false positives in these levels that might be caused by automated pruning methods can lead to a lack of significant sub-domains relevant to CS or the inclusion of core research areas from other domains, respectively, we decided to manually annotate a total of 759 categories in the first three levels for relevance for our purpose, and based on that removed 259 (32%) categories from the first three levels. Even though we also removed the children of these 259 categories, there were still around 1.4 million categories remaining in the ICS.

3.3.2 Community Detection

A network with an inherent community structure can be grouped into sets (communities) of nodes such that each set is densely connected within, and weakly connected across communities [3]. In our remaining ICS, categories from the same or similar domains or sub-domains were densely connected through child-parent links, such that we can assume that CS-relevant categories would be clustered together. Considering the large size of the remaining ICS (1.4 million categories),

we leveraged a widely used and fast community detection algorithm, namely, the Louvain algorithm [1]. This algorithm identified a total of 288 clusters in the remaining ICS. The largest and smallest clusters contained 243,597 categories and 1 category, respectively, and the mean and median size of these 288 clusters were 5044 and 41, respectively. To identify and remove CS-irrelevant clusters, we utilized the overlap of categories in those clusters with terms in ACM CSS and CSO. We removed 261 (94.1%) clusters with a total of 0.4 million (28.6%) categories which had no overlap with ACM CCS or CSO. Our inspection of the remaining 1 million categories showed that there were still substantial numbers of false positive categories. To address this issue, we next trained a machine learning model to predict false positive categories.

3.3.3 Embedding-Based Classification

Our next step for reducing false positives was to use a machine learning model to automatically distinguish relevant from irrelevant categories with high accuracy. We utilized embedding based approaches, namely Elmo [18], poincare [14] and node2vec [4] embeddings, to capture the contextual information of our texts, the structured information in our subtree, and the graph information of the child-parent links in our data. While we were able to create features through embeddings, it was difficult to obtain a training set with balanced labeled responses. Since the ratio of positive to negative categories in the remaining ICS was smaller than 1%, we were not able to label enough positive instances for model training through annotating a sample from the remaining ICS. In view of this difficulty, we considered a total of 1756 shared categories between the remaining ICS and ACM CSS or CSO as positive responses. Next, we obtained negative responses by manually annotating a sample from the remaining ICS, and collecting the children of the annotated negative responses. Since we obtained tens of thousands of CS-irrelevant categories (negative responses), we randomly sampled about 1756 categories from them to create a balanced training set by combining them with positive responses. We then utilized a multi-layer perceptron (MLP) to train a model to predict whether a category is CS-relevant or not. The cross validated ($k = 10$) F1 score of the model was around 90%. The Elmo-based model performed best, and the addition of node2vec features improved the performance slightly (1% to 2%). Therefore, we utilized the MLP model based on the features from Elmo and node2vec to predict whether a category is relevant to CS or not. We then applied the trained model to the remaining ICS, and removed all categories labeled as CS-irrelevant from the remaining ICS. Also, if any category was classified as CS-irrelevant, we also removed its child categories. This step removed the majority of categories from the ICS. The remaining ICS only contained about 11,000 (1.1%) categories.

3.3.4 Human-Created Rules

After inspecting the remaining ICS, we still found a substantial number of false positive categories in it. We also saw that there were more false positive categories in the bottom levels. Since manually identifying and removing individual

categories is time-consuming, we developed a set of rules or heuristics to handle patterned cases of false positives that were not captured by any of the above-mentioned steps to prune the ICS. In order to find effective rules, we randomly sampled hundreds of categories from the remaining ICS, and manually annotated whether they were relevant or not. This in-depth work revealed that most false positive categories had common parent categories, and these parent categories often shared common patterns. For example, a commonly shared parent of false positive categories was *Classification system by subject*. This category did not refer to classification methods or systems in CS, but classification schemas in other domains. We also found that the suffix *by subject* in parent categories often led to the inclusion of false positive children categories into the remaining ICS as well. Another example of patterned false positives was *Microsoft software*, which is relevant to CS in general, but irrelevant for our purpose. Therefore, we removed all categories containing the suffix *by subject* and the prefix *Microsoft*. Similarly, through filtering out the false positive categories from the sample and locating their parents by tracing bottom-up parenthood links, we identified around 50 patterns, and created corresponding rules to remove them. Overall, we removed about 4000 (35%) from the remaining ICS, and obtained 7355 categories. At this point, we had used 0.45% of the categories from WCT for WikiCSSH.

3.4 Extracting Fine-Grained Terms

Since a CS subject headings schema should also contain fine-grained concepts within each research area, we utilized all of the pages affiliated with CS-relevant categories identified through the previous steps. Based on our assumption that pages inherit the characteristic of CS-relevance from categories they are affiliated with, we extracted pages were all relevant to CS. This step refined our WikiCSSH with 761,383 pages that were affiliated with the 7355 categories in our remaining ICS.

3.5 Final WikiCSSH

The final WikiCSSH we built consists of 7K Wikipedia Categories organized as a tree, and 761K affiliated Wikipedia pages. Each category in WikiCSSH has on average 104 affiliated pages. Inter-category parent-child links capture the research field hierarchy. Category-page links capture concepts within a research field. Each category in WikiCSSH is assigned a level based on its lowest identified level in the tree. WikiCSSH contains core CS terms (including categories and pages) in levels 1–7, and ancillary CS terms in levels 8–20 (see Fig. 1). Core terms are highly relevant to CS research topics or concepts, while ancillary terms mainly represent interdisciplinary research topics and concepts. Core terms in WikiCSSH account for 63.5% of the terms in WikiCSSH. Users of WikiCSSH can decide which part of our vocabulary they want to use depending on their narrow or broad definition of CS.

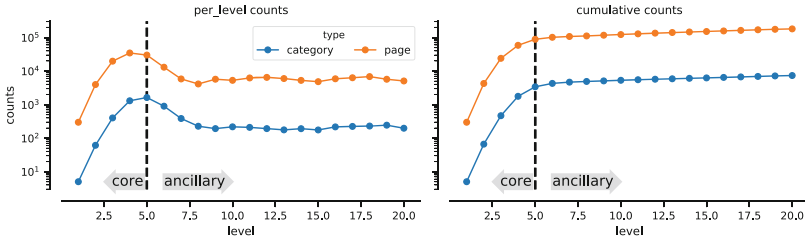


Fig. 1. Distribution of subject-heading counts in each level of WikiCSSH

4 Results and Evaluation

4.1 Comparison with Other CS Subject Headings

Table 2 shows quantitative statistics of our final vocabulary in comparison to ACM CCS and CSO. WikiCSSH contains ~ 7.4 thousand coarse-grained terms (categories) that are associated with ~ 0.75 million fine-grained terms (pages) in 20 levels. Therefore, WikiCSSH is 375 and 33 times larger than ACM CCS (2,000 terms) and CSO (22,000 terms), respectively. Besides that, while both ACM CCS and CSO have a hierarchical structure to represent the relations between the terms they contain, neither of them distinguishes between coarse-grained (categories) and fine-grained (pages) terms as well as between core and ancillary terms.

Table 2. Summary of existing subject headings in Computer Science

Vocabulary	Size	Curation
ACM CCS	2K	Expert labeling
CSO v.3.1	22K	Data mining
WikiCSSH	7.4K categories + 752K pages	Crowdsource + HITL data mining

4.2 Evaluation of Category Extraction Based on Human Annotated Data

In this section, we evaluate the performance of our methods for removing false positives. This evaluation also allows us to test whether our mixed methods approach can outperform any single method approach to pruning a large-scale dataset with a complex structure such as the WCT. We randomly selected a sample of categories from the ICS before our community detection step, and manually annotated whether the sampled categories were CS-relevant or not. Finally, we leveraged this annotated sample to evaluate precision and recall of different category sets extracted through different methods. It is important to note that we only evaluated categories. Pages inherit the characteristic of

relevance to CS from categories they are affiliated with and thus are assumed to share similar results with the evaluation for categories. Table 3 shows the evaluation results. The first three levels are not useful for evaluation as they have been selected manually. From level 4 onward, we find that the embedding based method (ML) achieved a higher precision compared to the community detection (CD) method at the expense of lower recall. Combining ML with rules also increases precision at the expense of lower recall, while combining all of CD, ML, and Rule improves the precision significantly in lower levels (more than 0.4 points for levels 6 and 7). This result provides empirical evidence for our argument that mixed methods can outperform a single method approach or pruning large-scale data with complicated structures.

Table 3. Precision (P) and recall (R) in core levels (recall for levels > 5 cannot be computed as that would require manually annotating all CS-relevant categories.)

Level	CD		ML		ML+Rule		CD+ML+Rule	
	P	R	P	R	P	R	P	R
1-3	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
4	0.36	1.00	0.64	0.88	0.85	0.90	0.85	0.88
5	0.17	1.00	0.66	0.90	0.87	0.79	0.87	0.79
6	0.07	/	0.25	/	0.47	/	0.83	/
7	0.01	/	0.23	/	0.33	/	0.82	/

4.3 Evaluation of WikiCSSH Against an Annotated Scholarly Dataset

A common application of a subject headings vocabulary is to tag scholarly papers with these subject headings. A domain-specific subject headings vocabulary can be considered effective or to provide high coverage if it enables the identification of important key-phrases in a dataset of scholarly papers from that domain. We used KP20k dataset [2, 10] for our evaluation. KP20k contains 20,000 CS research abstracts and human-annotated short key phrases from these abstract, such as *machine learning*, *data mining*, and *clustering*, among others. We matched each keyphrase in KP20k against the terms in WikiCSSH, ACM CSS, and CSO. We basically searched for all exact matches of stemmed words in the keyphrases with stemmed terms from the subject headings vocabularies. For our evaluation, we counted both the number of unique matched phrases and the total number of matched phrases. Since the total number of phrase matches is going to be biased towards frequently occurred concepts that are likely to be present in all vocabularies, we also used unique phrase matches to identify coverage. As reported in Table 4, WikiCSSH extracted 62,635 (8.23%) unique phrases and 1,456,690 (48.3%) total phrases from KP20k, and most of them were contributed by WikiCSSH’s core part. The numbers of extracted unique and total phrases for ACM

were 1,284 (0.17%) and 302,326 (10%), and for CSO 10,985 (1.44%) and 797,447 (26.4%). This evaluation suggests that WikiCSSH supports comparatively high coverage of CS terms occurring in scholarly texts.

Table 4. Comparison of coverage of various vocabularies on phrases in KP20k corpus (percents = phrases extracted by vocabulary/annotated phrases in KP20K)

Vocabulary	Unique phrases	Total phrases
ACM CCS	1,284 (0.17%)	302,326 (10%)
CSO	10,985 (1.44%)	797,477 (26.4%)
WikiCSSH (core)	45,345 (5.96%)	1,207,075 (40%)
WikiCSSH (ancillary)	17,290 (2.27%)	249,515 (8.27%)
WikiCSSH (total)	62,635 (8.23%)	1,456,590 (48.3%)

We also calculated the ratios of total to unique phrases, respectively, for the core and ancillary part of the WikiCSSH, which show WikiCSSH’s ability to extract rare phrases from KP20K. We found that the ratio of total to unique phrases for the core part of WikiCSSH is 26.62, while for the ancillary part, it is only 14.43. Put differently, the core part of WikiCSSH is more likely to capture frequently occurring phrases in CS research articles, while the ancillary part tends to capture rare phrases. Similarly, for ACM CCS and CSO, the ratio of total to unique phrases were 235.5 and 72.6, respectively. This result indicates that WikiCSSH is more likely to extract rare phrases from scholarly articles compared to ACM CCS and CSO. CSO, which was constructed from mining large-scale scholarly data in CS, contains a lower proportion of rare phrases that occur in KP20K compared to WikiCSSH. A possible reason for this lower coverage may be automated data mining methods inability to capture low probability signals.

5 Conclusion, Discussion and Limitations

We have presented WikiCSSH, a large-scale subject headings vocabulary for the CS domain, that we developed using a human-in-the-loop workflow that leverages the crowd-sourced Wikipedia Category Tree. WikiCSSH outperforms two alternative CS vocabularies, namely ACM CCS and CSO, in number of items, coverage of key-phrases in a benchmark dataset of scholarly papers from CS, and categorization of the subject headings into coarse-grained versus fine-grained entries. Users of WikiCSSH can decide which part of WikiCSSH they want to use depending on their needs. For example, users may want to i) use the 7,355 hierarchically structured categories for indexing (research areas and topics in) documents, or ii) use the 0.75 million concrete, fine-grained terms (from pages) within categories for more detailed concept analysis, or iii) select

the core and/or ancillary part of WikiCSSH according to their broad or narrow definition of computer science as needed for their work.

Our work also contributes to methodological work for leveraging existing crowd-sourced data when the main challenge is filtering out false positives to increase precision of some target application. Building a sizeable domain-specific vocabulary like WikiCSSH would be extremely expensive and/ or time consuming if one only relied on manual work by domain experts. However, existing crowd-sourced data with a permissible license opens up an opportunity to build a large-scale, structured vocabulary at low cost in terms of both time and human resources. That being said, our approach is more costly and time-consuming than a fully automated data mining- based approach due to the substantial human interventions we made part of our process. However, we showed that our approach can capture relevant yet rare phrases that might be ignored by fully automated data mining solutions. Our work also illustrates the challenges resulting from using the given structure of Wikipedia data for our specific task and assesses possible solutions to overcome these challenges through the methods described earlier. Our workflow can be extended to construct subject headings for other domains by modifying the rules and training approaches. Code for replicating the construction and refinement of WikiCSSH along with the latest version of WikiCSSH can be found at: <https://github.com/uiuc-ischool-scanr/WikiCSSH> [5].

We acknowledge the limitations of our evaluation of WikiCSSH, for which we aimed to map key-phrases in scholarly papers to entries in WikiCSSH. Even though WikiCSSH outperformed other domain-specific vocabularies in terms of coverage of KP20K, this result only highlights its potential to extract more CS-relevant phrases from scholarly articles than alternative vocabularies. However, precision may be more important if we aim to categorize or index documents based on a controlled vocabulary. In our future work, we plan to test the performance of WikiCSSH for analyzing scholarly data, indexing and categorizing documents, and mining phrases and topics. Additionally, because Wikipedia data and classification methods are being updated over time, we plan to update WikiCSSH based on new data and with new methods as well.

References

1. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *J. Stat. Mech: Theory Exp.* **2008**(10), P10008 (2008). <https://doi.org/10.1088/1742-5468/2008/10/p10008>
2. Gallina, Y., Boudin, F., Daille, B.: Large-scale evaluation of keyphrase extraction models. arXiv preprint [arXiv:2003.04628](https://arxiv.org/abs/2003.04628) (2020)
3. Girvan, M., Newman, M.E.: Community structure in social and biological networks. *Proc. Natl. Acad. Sci.* **99**(12), 7821–7826 (2002)
4. Grover, A., Leskovec, J.: Node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2016, pp. 855–864. Association for Computing Machinery, New York (2016). <https://doi.org/10.1145/2939672.2939754>
5. Han, K., Yang, P., Mishra, S., Diesner, J.: Wikicssh - computer science subject headings from Wikipedia (2020). <https://doi.org/10.13012/B2IDB-0424970.V1>

6. Hoffart, J., Suchanek, F.M., Berberich, K., Weikum, G.: Yago2: a spatially and temporally enhanced knowledge base from Wikipedia. *Artif. Intell.* **194**, 28–61 (2013). <https://doi.org/10.1016/j.artint.2012.06.001>
7. Lehmann, J., et al.: Dbpedia—a large-scale, multilingual knowledge base extracted from Wikipedia. *Semant. Web* **6**(2), 167–195 (2015)
8. Levine, T.R.: Rankings and trends in citation patterns of communication journals. *Commun. Educ.* **59**(1), 41–51 (2010)
9. Medelyan, O., Witten, I.H., Milne, D.: Topic indexing with Wikipedia. In: *Proceedings of the AAAI WikiAI Workshop*, vol. 1, pp. 19–24 (2008)
10. Meng, R., Zhao, S., Han, S., He, D., Brusilovsky, P., Chi, Y.: Deep keyphrase generation. *arXiv preprint arXiv:1704.06879* (2017)
11. Mishra, S., Fegley, B.D., Diesner, J., Torvik, V.I.: Expertise as an aspect of author contributions. In: *Workshop on Informetric and Scientometric Research (SIG/MET)*, Vancouver (2018)
12. Mishra, S., Fegley, B.D., Diesner, J., Torvik, V.I.: Self-citation is the hallmark of productive authors, of any gender. *PLoS ONE* **13**(9), e0195773 (2018). <https://doi.org/10.1371/journal.pone.0195773>
13. Mishra, S., Torvik, V.I.: Quantifying Conceptual Novelty in the Biomedical Literature. *D-Lib Mag.: Mag. Digit. Libr. Forum* **22**(9–10) (2016). <https://doi.org/10.1045/september2016-mishra>
14. Nickel, M., Kiela, D.: Poincaré embeddings for learning hierarchical representations. In: *Advances in Neural Information Processing Systems*, vol. 30, pp. 6338–6347. Curran Associates, Inc. (2017)
15. Nielsen, F.Å., Mitchen, D., Willighagen, E.: Scholia, scientometrics and Wikidata. In: Blomqvist, E., Hose, K., Paulheim, H., Lawrynowicz, A., Ciravegna, F., Hartig, O. (eds.) *ESWC 2017. LNCS*, vol. 10577, pp. 237–259. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70407-4_36
16. Osborne, F., Motta, E.: Klink-2: integrating multiple web sources to generate semantic topic networks. In: Arenas, M., et al. (eds.) *ISWC 2015. LNCS*, vol. 9366, pp. 408–424. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25007-6_24
17. Packalen, M., Bhattacharya, J.: Age and the trying out of new ideas. *J. Hum. Cap.* **13**(2), 341–373 (2019). <https://doi.org/10.1086/703160>
18. Peters, M., et al.: Deep contextualized word representations. In: *Proceedings of the 2018 Conference of the NAACL-HLT*, pp. 2227–2237. Association for Computational Linguistics, Stroudsburg (June 2018). <https://doi.org/10.18653/v1/N18-1202>
19. Salatino, A.A., Thanapalasingam, T., Mannocci, A., Osborne, F., Motta, E.: The computer science ontology: a large-scale taxonomy of research areas. In: Vrandečić, D., et al. (eds.) *ISWC 2018. LNCS*, vol. 11137, pp. 187–205. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00668-6_12
20. Shang, J., Liu, J., Jiang, M., Ren, X., Voss, C.R., Han, J.: Automated phrase mining from massive text corpora. *IEEE Trans. Knowl. Data Eng.* **30**(10), 1825–1837 (2018)
21. Wang, Y., Zhu, M., Qu, L., Spaniol, M., Weikum, G.: Timely YAGO: harvesting, querying, and visualizing temporal knowledge from Wikipedia. In: *Proceedings of the 13th International Conference on Extending Database Technology*, pp. 697–700 (2010)



Integrating Knowledge Graphs for Analysing Academia and Industry Dynamics

Simone Angioni¹, Angelo A. Salatino²(✉), Francesco Osborne²,
Diego Reforgiato Recupero¹, and Enrico Motta²

¹ Department of Mathematics and Computer Science,
University of Cagliari, Cagliari, Italy

{simone.angioni,diego.reforgiato}@unica.it

² Knowledge Media Institute, The Open University, Milton Keynes, UK
{angelo.salatino,francesco.osborne,enrico.motta}@open.ac.uk

Abstract. Academia and industry are constantly engaged in a joint effort for producing scientific knowledge that will shape the society of the future. Analysing the knowledge flow between them and understanding how they influence each other is a critical task for researchers, governments, funding bodies, investors, and companies. However, current corpora are unfit to support large-scale analysis of the knowledge flow between academia and industry since they lack of a good characterization of research topics and industrial sectors. In this short paper, we introduce the Academia/Industry DynAmics (AIDA) Knowledge Graph, which characterizes 14M papers and 8M patents according to the research topics drawn from the Computer Science Ontology. 4M papers and 5M patents are also classified according to the type of the author's affiliations (academy, industry, or collaborative) and 66 industrial sectors (e.g., automotive, financial, energy, electronics) obtained from DBpedia. AIDA was generated by an automatic pipeline that integrates several knowledge graphs and bibliographic corpora, including Microsoft Academic Graph, Dimensions, English DBpedia, the Computer Science Ontology, and the Global Research Identifier Database.

Keywords: Scholarly data · Knowledge graph · Topic detection · Bibliographic data · Scholarly ontologies · Research dynamics

1 Introduction

Academia and industry are constantly engaged in a joint effort for producing scientific knowledge that will shape the society of the future. Analysing the knowledge flow between them and understanding how they influence each other is a critical task for researchers, governments, funding bodies, investors, and companies. Researchers have to be aware of how their effort impacts the industrial sectors; government and funding bodies need to shape research policies and

funding decisions; companies have to constantly monitor the scientific innovation that may be developed in products or services.

The relationship between academia and industry has been analysed from several perspectives, focusing, for instance, on the characteristics of direct collaborations [4], the influence of industrial trends on curricula [16], and the quality of the knowledge transfer [5]. Unfortunately, the lack of a large scale corpus for tracking knowledge flow limited the scope of previous works, which are typically restricted to small-scale datasets or focused on very specific research questions [2, 6].

In order to analyse the knowledge produced by academia and industry, researchers typically exploit corpora of research articles or patents [3, 4]. Today, we have several large-scale knowledge graphs which describe these documents. Some examples include Microsoft Academic Graph¹, Open Research Corpus [1], the OpenCitations Corpus [10], Scopus², AMiner Graph [17], the Open Academic Graph (OAG)³, Core [7], Dimensions Corpus⁴, and the United States Patent and Trademark Office Corpus⁵. However, these resources are unfit to support large-scale analysis about the knowledge flow since they suffer from three main limitations: 1) they do not directly classify a document according to its provenance (e.g., academia, industry), 2) they offer only coarse-grained characterizations of research topics, and 3) they do not characterize companies according to their sectors (e.g., automotive, financial, energy, electronics).

In this short paper, we introduce the Academia/Industry DynAmics (AIDA) Knowledge Graph, describing 14M articles and 8M patents (in English) in the field of Computer Science according to the research topics drawn from the Computer Science Ontology. 4M articles and 5M patents are also classified according to the type of the author's affiliations (academy, industry, or collaborative) and 66 industrial sectors (e.g., automotive, financial, energy, electronics) obtained from DBpedia. AIDA was generated by integrating several knowledge graphs and bibliographic corpora, including Microsoft Academic Graph (MAG), Dimensions, English DBpedia [8], the Computer Science Ontology (CSO) [14], and the Global Research Identifier Database (GRID)⁶. It can be downloaded for free from the AIDA website⁷ under the CC BY 4.0 license.

AIDA was designed to allow researchers, governments, companies and other stakeholders to easily produce a variety of analytics about the evolution of research topics across academy and industry and study the characteristics of several industrial sectors. For instance, it enables detecting what are the research trends most interesting for the automotive sector are or which prevalent industrial topics were recently adopted and investigated by the academia. Furthermore, AIDA can be used to train machine learning systems for predicting the

¹ <https://aka.ms/msracad>.

² <https://www.scopus.com/>.

³ <https://www.openacademic.ai/oag/>.

⁴ <https://www.dimensions.ai/>.

⁵ <https://www.uspto.gov/>.

⁶ <https://www.grid.ac/>.

⁷ <http://aida.kmi.open.ac.uk>.

impact of research dynamics [11]. A preliminary versions of AIDA was used to support a comprehensive analysis of the research trends in the main venues of Human-Computer Interaction [9].

2 Knowledge Graph on Academic and Industrial Dynamics

The Academia/Industry DynAmics Knowledge Graph describes a large collection of publications and patents in Computer Science according to the kind of affiliations of their authors (academia, industry, collaborative), the research topics, and the industrial sectors.

Table 1. Distribution of publications and patents classified as Academia, Industry and Collaboration.

	Academia	Industry	Collaboration	Total classified	Total
Publications	3,043,863	730,332	108,506	3,882,701	14,317,130
Patents	133,604	4,741,695	16,335	4,891,634	7,940,034

Table 1 reports the number of publications and patents from academy, industry, and collaborative efforts. Most scientific publications (78.4%) are written by academic institutions, but industry is also a strong contributor (18.8%). Conversely, 96.9% of the patents are from industry and only 2.7% from academia. Collaborative efforts appears limited, including only 2.8% of the publications and 0.4% of the patents.

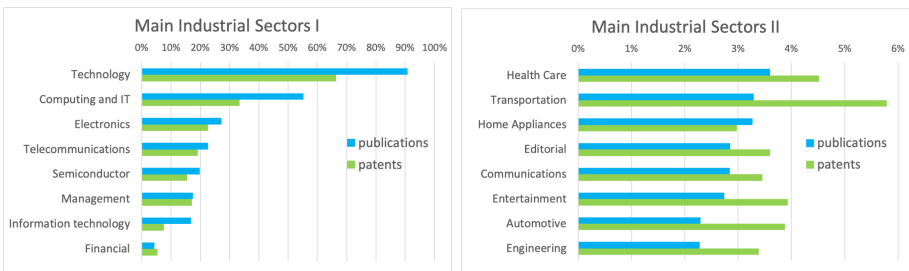


Fig. 1. Distribution of publications and patents in the main 16 industrial sectors.

Figure 1 reports the percentage of publications and patents associated with the most prominent industrial sectors. The most popular sectors in AIDA are directly pertinent to Computer Science (e.g., Technology, Computing and IT, Electronics, and Telecommunications, and Semiconductors), but we can also see

many other sectors which adopt Computer Science technologies such as Financial, Health Care, Transportation, Home Appliance, and Editorial. The first group produces a higher percentage of publications, while the second generates more patents.

The data model of AIDA is available at <http://aida.kmi.open.ac.uk/ontology> and it builds on SKOS and CSO. It focuses on four types of entities: *publications*, *patents*, *topics*, and *industrial sectors*.

The main information about publications and patents are given by mean of the following semantic relations:

- *hasTopic*, which associates to the documents all their relevant topics drawn from CSO;
- *hasAffiliationType* and *hasAssigneeType*, which associates to the documents the three categories (academia, industry, or collaborative) describing the affiliations of their authors (for publications) or assignees (for patents);
- *hasIndustrialSector*, which associates to documents and affiliations the relevant industrial sectors drawn from the Industrial Sectors Ontology (INDUSO) we describe in the next sub-section.

A dump of AIDA in Terse RDF Triple Language (Turtle) is available at <http://aida.kmi.open.ac.uk/downloads>.

2.1 AIDA Generation

AIDA was generated using an automatic pipeline that integrates and enriches data from Microsoft Academic Graph, Dimensions, Global Research Identifier Database, DBpedia, CSO [14], and INDUSO. It consists of three steps: i) topics detection, ii) extraction of affiliation types, and iii) industrial sector classification.

Topic Detection - *hasTopic*. In this phase, we annotated each document with a set of research topics drawn from CSO: the intent is both to obtain a fine-grained representation of topics, with the aim of supporting large-scale analyses of research trends [12], and to have the same representation across the paper and the patents. The latter is critical since it allows to track the behavior of a topic according to different documents from academia and industry and assess its importance for the different industrial sectors.

As first step, we selected all the publications and patents from MAG and Dimensions within the domain of Computer Science. To achieve this, we extracted from MAG all the papers classified as “Computer Science” according to their classification: the Fields of Science (FoS) [15]. Similarly, we extracted from Dimensions all the patents pertinent to Computer Science according to the International Patent Classification (IPC) and the fields of research (FoR) taxonomy. The resulting dataset consists of 14M publications and 8M patents. Next, we run the CSO Classifier [13] on the title and the abstract of all the 22M documents. In addition to extracting the topics relevant to the text, we

also exploited the same tool for including all their super topics according to the CSO. For instance, a paper tagged with *neural networks* was also assigned the topic *artificial intelligence*. This solution enables to monitor more abstracts and high level topics that are not always directly referred in the documents.

Extraction of Affiliation Types - *hasAffiliationType*, *hasAssigneeType*.

In this step, we classified research papers and patents according to the nature of their authors' affiliation in GRID, which is an open database identifying and typing over 90 K organizations involved in research. Specifically, GRID describes research institutions with an identifier, geographical location, date of establishment, alternative labels, external links (including Wikipedia), and type of institution (e.g., Education, Healthcare, Company, Archive, Nonprofit, Government, Facility, Other). MAG and Dimensions map a good number of affiliations to GRID IDs. We classified a document as 'academia' if all the authors have an educational affiliation and as 'industry' if all the authors have an industrial affiliation. Documents whose authors are from both academia and industry were classified as 'collaborative'.

Extraction of Industrial Category - *hasIndustrialSector*.

In this step, we characterised documents from industry according to the Industrial Sectors Ontology (INDUSO)⁸, an ontology that we designed for this specific task. We designed INDUSO by merging and arranging in a taxonomy a large set of industrial sectors that we extracted from the affiliations of the paper authors and the patent assignees. First, we used the mapping between GRID and Wikipedia to retrieve the affiliations on DBpedia by extracting the objects of the properties "About:Purpose" and "About:Industry". This resulted in a noisy and redundant set of 699 sectors. We then manually analysed them with the help of domain experts and merged similar industrial sectors, finally obtaining 66 distinct sectors. For instance, the industrial sector "Computing and IT" in the resulting knowledge graph was derived from categories such as "Networking hardware", "Cloud Computing", and "IT service management". Finally, we designed INDUSO by arranging the 66 sectors in a two level taxonomy using the SKOS schema⁹. INDUSO also links the 66 main industrial sectors to the original 699 sectors using the *derivedFrom* relation from PROV-O¹⁰.

Finally, we associated to each document all the industrial sectors that were derived from the DBpedia representation of its affiliations. For instance, the documents with an author affiliation described in DBpedia as 'natural gas utility' were tagged with the second level sector 'Oil and Gas Industry' and the first level sector 'Energy'.

⁸ INDUSO - <http://aida.kmi.open.ac.uk/downloads/induso.ttl>.

⁹ <https://www.w3.org/TR/2008/WD-skos-reference-20080829/skos.html>.

¹⁰ <https://www.w3.org/TR/prov-o/>.

3 Conclusions and Future Work

In this paper we introduced AIDA, the Academic/Industry DynAMics Knowledge Graph. AIDA includes knowledge on research topics of 14M publications extracted from MAG and 8M patents extracted from Dimensions. Moreover, 4M papers and 5M patents have also been classified according to the types of authors' and assignees' affiliations and 66 industrial sectors.

We are currently working on several next steps: i) we will provide our insights and analysis of research topic trends on academia and industry dynamics; ii) we are setting up a public triplestore to allow everyone to perform SPARQL queries to come up with further analytics and analysis out of the generated data; iii) we are setting up a pipeline that will automatically update AIDA with recent data; and iv) we will provide a rigorous evaluation of each component of the AIDA pipeline.

References

1. Ammar, W., et al.: Construction of the literature graph in semantic scholar. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 84–91. Association for Computational Linguistics (2018)
2. Anderson, M.S.: The complex relations between the academy and industry: views from the literature. *J. High. Educ.* **72**(2), 226–246 (2001)
3. Angioni, S., Osborne, F., Salatino, A.A., Reforgiato, D., Recupero, E.M.: Integrating knowledge graphs for comparing the scientific output of academia and industry. In: International Semantic Web Conference ISWC, vol. 2019, pp. 85–88 (2019)
4. Ankrah, S., Omar, A.T.: Universities-industry collaboration: a systematic review. *Scand. J. Manag.* **31**(3), 387–408 (2015)
5. Ankrah, S.N., Burgess, T.F., Grimshaw, P., Shaw, N.E.: Asking both university and industry actors about their engagement in knowledge transfer: what single-group studies of motives omit. *Technovation* **33**(2–3), 50–65 (2013)
6. Bikard, M., Vakili, K., Teodoridis, F.: When collaboration bridges institutions: the impact of university-industry collaboration on academic productivity. *Org. Sci.* **30**(2), 426–445 (2019)
7. Knoth, P., Zdrahal, Z.: Core: three access levels to underpin open access. *D-Lib Mag.* **18**(11/12), 1–13 (2012)
8. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., et al.: Dbpedia-a large-scale, multilingual knowledge base extracted from Wikipedia. *Seman. Web* **6**(2), 167–195 (2015)
9. Mannocci, A., Osborne, F., Motta, E.: The evolution of IJHCS and CHI: a quantitative analysis. *Int. J. Hum.-Comput. Stud.* **131**, 23–40 (2019)
10. Peroni, S., Shotton, D.: Opencitations, an infrastructure organization for open scholarship. *Quant. Sci. Stud.* **1**(1), 428–444 (2020)
11. Salatino, A., Osborne, F., Motta, E.: Researchflow: understanding the knowledge flow between academia and industry (2020). <http://skm.kmi.open.ac.uk/rf-utkfbaa/>
12. Salatino, A.A., Osborne, F., Motta, E.: How are topics born? Understanding the research dynamics preceding the emergence of new areas. *PeerJ Comput. Sci.* **3**, e119 (2017)

13. Salatino, A.A., Osborne, F., Thanapalasingam, T., Motta, E.: The CSO classifier: ontology-driven detection of research topics in scholarly articles. In: Doucet, A., Isaac, A., Golub, K., Aalberg, T., Jatowt, A. (eds.) TPDFL 2019. LNCS, vol. 11799, pp. 296–311. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30760-8_26
14. Salatino, A.A., Thanapalasingam, T., Mannocci, A., Birukou, A., Osborne, F., Motta, E.: The computer science ontology: a comprehensive automatically-generated taxonomy of research areas. *Data Intell.* 1–38 (2019). <https://doi.org/10.1162/dint.a.00055>
15. Sinha, A., et al.: An overview of Microsoft academic service (MAS) and applications. In: Proceedings of the 24th International Conference on World Wide Web, pp. 243–246 (2015)
16. Weinstein, L.B., Kellar, G.M., Hall, D.C.: Comparing topic importance perceptions of industry and business school faculty: is the tail wagging the dog? *Acad. Educ. Leadersh. J.* **20**(2), 62 (2016)
17. Zhang, Y., Zhang, F., Yao, P., Tang, J.: Name disambiguation in AMiner: clustering, maintenance, and human in the loop. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1002–1011 (2018)



A Philological Perspective on Meta-scientific Knowledge Graphs

Tobias Weber^(✉) 

Ludwig-Maximilians-Universität München, Munich, Germany
weber.tobias@campus.lmu.de

Abstract. This paper discusses knowledge graphs and networks on the scientific process from a philological viewpoint. Relevant themes are: the smallest entities of scientific discourse; the treatment of documents or artefacts as texts and commentaries in discourse; and links between context, (co)text, and data points. As an illustration of the micro-level approach, version control of linguistic examples is discussed as a possible field of application in this discipline. This underlines the claim for data points to be treated like unique entities, which possess metadata of the datum and any text generating knowledge from it.

Keywords: Version control · Meta-documentary linguistics · Identifier · Knowledge graphs · Scientific discourse

1 Why Philology?

At first glance, it might appear strange that a philologist would be involved in meta-scientific discussions among IT specialists and arguing for the philological perspective in the discourse. These topics would be expected to attract sociologists, philosophers, and data scientists - while the stereotypical philologist's place is in a library or archive, covered in the dust of old books and manuscripts. In this paper, principles of philology are presented with a view to ways in which they may be applied to support the creation of sound and thorough databases, ontologies, and knowledge graphs.

Philology comprises a range of definitions, usually including ‘textual curatorship and interpretation’ [12] or ‘the multifaceted study of texts, languages, and the phenomenon of language itself’ [29, p. IX]. Independent of its linguistic or literary orientation, it has a strong history of working with graphs. These graphs and their modern, digital representations help us to understand relationships between languages (like a phylogenetic tree), manuscripts, or versions of writings [8]. They form genealogies to help us understand aspects of our research objects in relation to other artefacts and abstracta, and investigate the history behind these artefacts. As such, the philological approach is characterised by ‘an attitude of respect for the datum’ [32, p. 18]. It provides the basis for investigating intertextuality and the human factor in the creation and reception of

scientific discourse. As such, it can be seen as the ‘fundamental science of human memory’ [17, p. 345], with (textual) artefacts at the core of our endeavour.

If we consider science as a discourse, the exchange of texts and commentaries, facilitated by creating, discussing, and reviewing textual artefacts and constantly developing through a growing body of literature, the need for a philological stance becomes clear. As scholars, we are accessing this ‘memory’ of our disciplines by ‘recalling, iterating, reading, commenting, criticizing, and discussing what was deposited in the remote or recent past’ [1, p. 97]. This holds true for all disciplines, especially those in which discourse forms the basis of knowledge generation. These disciplines cannot rely on a ‘transcription device’ [14] which turns raw data (e.g. a sound recording) into primary data (e.g. a transcription) [15] reliably and independently of the researcher who abstracts data to create a text. Thus, wherever humans are involved in the selection, analysis, and interpretation of the artefacts, we can make a case for philological enquiry [25]. The documents we produce as evidence of our membership in the scientific community [6] are the starting point for charting science [3].

Philologists are increasingly aware and capable of applying computational methods in their research. Those may be applications of computational linguistics, automatic annotations, or the encoding of our artefacts in XML formats [4] such as the TEI standards [9, 22, 27]. These formats do not only enable digital processing but allow for the inclusion of metadata which enrich information as thick metadata [19] for presenting and storing data [28], ideally to keep it usable ‘500 years from now’ [33]. The philological stance is, therefore, independent from any practical application or method and functions as a guiding principle throughout the methodology.

2 Layers of Knowledge

As mentioned earlier, philologists use knowledge graphs to represent relationships between artefacts based on textual and contextual clues. These are contained inside every text, or to an increasing degree in the metadata. It should be noted that commentaries about texts, which are peripheral to a particular artefact inasmuch as they are not contained by default, can form new texts in their own right with a set of new metadata. The transfer from data to text must not lose metadata on the underlying data sets while coming with a set of metadata on the text itself. Likewise, there are no identical copies, as every artefact bears traces of the technology used to (re)create it [26], they originate in different contexts. For example, a digital copy of a document comes with a set of individual metadata different from those of a printout, with its aesthetics and presentational formats as a potential point in a meta-scientific discussion. Similarly, excerpts of text form new entities in an information science interpretation of text [23].

For this paper, I construct a simple typology of scientific artefacts and their relationship to the processes of scholarly writing and publishing. This typology illustrates the different aspects of science which can be included, analysed,

and represented using knowledge graphs (e.g. citation tracking, ontologies like DBpedia). Aside from time as a linear component (e.g. in genealogies or stemmata), there are two basic distinctions (Table 1). Firstly, the difference between the inherent and external attributes of the artefact, or concrete and abstract information about it. Secondly, we can refine the scope from the surroundings or container (macro-level) of the artefact or entity (meso-level) to its smallest components (micro-level). In other words, mirroring layers of philological and linguistic research on a text to the meta-level of the commentary. This mapping can either be structured in directed graphs by linear time (as versions) or symmetrically, requiring every ‘commentary’ to the ‘primary text’ to be linked back to this source in its own textual attributes.

Table 1. Typology of representable aspects of science.

	Concrete	Abstract
macro-level	context	meta-context
meso-level	text or cotext	meta-text or commentary
micro-level	constituent or ‘component’	metadata

On the macro-level, we are dealing with the contexts in which a scientific artefact is embedded. The term context is to be understood primarily as the global extra-linguistic context (i.e. existing outside of the text), while the textual context [20] is particular to each text and relevant for the interpretation of semantics within it (i.e. its relation to the lower level, e.g. decoding of references). Examples of the global context are the inclusion in a particular journal or collection, as well as metadata about the artefact (format, size). The textual context might be exemplified by an article referring to ‘the data’ as a particular, identifiable set of data used for the study. The meta-level can be conceptualised as the position of the artefact among other artefacts, including its references to previous literature, the tracking of citations after publication, and possibly even a comparison of similarity with other documents [11]. The latter two are recorded by libraries and publishers and stored in databases [16].

The meso-level is the level of the actual text, or for its relation to the micro-level the co-text, as the linguistic environment of a text constituent [7]. This layer is charted by numerous scholars aiming to train machine learning algorithms for NLP and creating ontologies and databases of knowledge; it might be represented as n-grams of variable size around a particular word or concept. On its meta-level, we encounter ‘commentaries’, i.e. external texts referencing, discussing, or reviewing the original text. As a central scientific procedure, this has also been represented in digital formats. We might consider possible applications linking knowledge generated in one text to information contained in other scientific texts, as a representation of knowledge generation [21], or with a view to linking scholars, institutions, and datasets [13].

On the micro-level, the focus of our interest is the smallest entities or text constituents of science and the scientific text. These may be concepts or terms

relevant for named entity recognition [18] in the creation of ontological databases [2], or data sets or examples which are discussed in the text. On the meta-level, these smallest entities can create the biggest issues, as they come with large amounts of metadata. To provide an example from linguistics: Imagine an author citing a phrase from a longer narrative which is contained in a corpus, stored in an archive. The metadata should include the relation of the cited phrase to the superset or the narrative and the corpus, as a version with information on its creation. The language of the example, data about the consultant, and information on the documentation project (at minimum place and time of recording) are needed to identify and keep track of the source. Ideally, ‘thick metadata’ [19] cover all necessary information, which, in turn, may consist of full graphs itself (e.g. a genealogy of speakers). Furthermore, the micro-level relates to texts and commentaries across a range of contexts, and these may influence their representation, annotation, or interpretation. In other words, two instances may cite the same source but apply different means of analysis and reach different conclusions - as a result, we can find the same example arguing for *and* against a theory. Since this aspect is underrepresented in the literature, I will be briefly discuss it in the following section.

3 Keeping Track of Data

The ideal knowledge graph to chart scientific knowledge generation thoroughly links all concrete entities to the rest of scientific artefacts, while containing full accounts of (meta)data on all lower levels. That being said, all lower-level constituents shall be identifiable and contain information on their use throughout all (con)texts, at the same time linking the published version to its parent nodes (e.g. collections, full data sets), and allowing for enquiry through the metadata. Consequently, we need to treat data points like abstract entities collected in ontologies (i.e. unique named entities). Using metadata on the constituents and tracing the procedure of knowledge generation based on them, allows us to tell the story of science through its artefacts. This creates the opportunity to write the ‘meta-documentation’ [5] as a narrative of data use in science - the discourse we want to preserve in our (cultural) ‘memory’.

Yet, this preservation and curation of our scientific legacy is not exclusively useful for posterity. Gaining access to actual data use (in citation, analysis, commentary) and the underlying links in the textual matter of the discourse, can shed light on biases [24], trends, and epistemological foundations of science. And, while this endeavour is currently based on linking data from the macro- and meso-level (most visible in citation tracking [30]), we must not ignore the potential of the micro-level. At this point, where particular data points turn into knowledge, data becomes text (see Fig. 1). Yet, both sides are linked by attributes of text or data on any level of description (i.e. abstractness). Each aspect of metadata represented in Fig. 1 has its own attributes, e.g. individual have names, dates of birth, parents, employers, speak languages (all possible attributes in TEI, OWL, or schema.org), which may be included in other parts of

the attributes. For example, different data points by mother and daughter could be linked through their genealogical affiliation; or researchers attributed to their respective institutions with their projects, publications etc. The attributes in the periphery of the non-exhaustive diagram can be at the centre of other diagrams, e.g. different research articles may be at the centre, yet linked through their metadata. Importantly, text and underlying data are closely linked but their respective sets of metadata do not interfere. Even on the extreme abstract of ‘field’ (i.e. macro-level context), there can be different values for the attributes, e.g. this paper using philology on the data plane to generate knowledge on the textual plane of knowledge graphs.

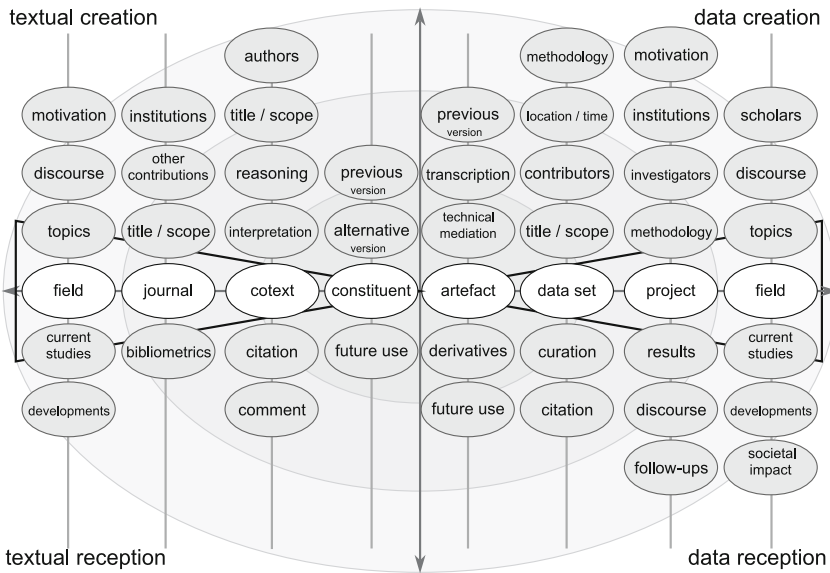


Fig. 1. Exemplary network of scientific text and data artefacts. The textual plane is on the left, the data on the right; the top half refers to attributes preceding the entity, e.g. relevant in its creation, while the bottom half lists aspects in the reception of the artefact, i.e. after its creation. The centre point (origin) is where scientific data is transferred into the realm of scientific knowledge with various degrees of abstractness around this point. The list of attributes for each instance is not exhaustive; other attributes or attributes of the presented attributes may be relevant.

In language documentation, fieldworkers rarely find uncharted land, i.e. to any project there is a precursor, historical interactions between the researcher and the communities, prior research on a language, or existing artefacts in an archive. As a result, it is common to find recurring themes, interacting with the same families as other researchers, and interacting with their research, even by critically assessing their work and righting historical wrongs (e.g. colonialism, missionary efforts). But for a depiction of the progress in science, we do not only

require accounts of controversy and debate - we also need to understand links in data which are not immediately obvious. These are, for example, cases in which a family has supported several different projects through multiple generations, and the knowledge contained in the various collections is linked by the genealogy of the consultants; or where a cited example has later been changed, rejected, or edited for a variety of reasons (e.g. consultant's request, new transcriptions, publishing policies) which are not inherent to the datum but to its contexts of creation and reception. While readers may consider this to be a specific issue of linguistics, there are instances in other fields where such a micro-level approach to the scientific process may yield yet unknown potential. For example - if this was ethically acceptable - cross-linking of participants' data who have been on different medical experiments, provided data to separate socioeconomic enquiries, or answered questionnaires for multiple surveys. Only in certain, specific scenarios is it possible to cross-link such data, e.g. historical medical data [10]. The potential of creating a holistic image does not lie within the macroscopic view of science but within every single data point.

As far as linguistics are concerned, a strong case can be made for the introduction of standardised tracking codes for linguistic examples which grant access to the multiple links behind each utterance. This version control must link an excerpted version to the full version with all relevant metadata on the data plane, while containing information on the role of the example across all layers, in citation, analysis, reception, and recreation on the textual plane [31]. This would expand the networks of researchers' contributions on a paper, as we are reaching conclusions building on the analyses, transcriptions, and interpretations of other scholars. Furthermore, already the selection of an example over others can shape the conclusions, and, even if we copy examples without altering them, we do accept them as valid contributions in the discourse and interact with them. While the precise form and technical implementation of these codes can be debated, their tracking does not differ much from the information already gathered on articles, e.g. providing information on further citations, references cited, bibliographic information, figures and tables. There is no reason why cited examples and (parts of) data sets should not be identified and tracked accordingly, with the philological 'attitude of respect' for the micro-level constituents of knowledge generation. They are identifiable entities in their own right.

4 Conclusion

Philology offers a structure to analyse texts and commentaries on their artefactual level, as well as their contexts and the components constituting it. All of these layers can be presented using knowledge graphs, in themselves and through their relations to other artefacts or other levels of description. Constructing the network of science, as a document- or artefact-based discourse, requires the linking of all layers to uncover the narratives behind science. Independent of the technical details and requirements for this endeavour, the focus on the smallest entities of scientific research, data points and examples, can help to paint a holistic image. Thus, an extended focus on the micro-level, apart from Named Entity

Recognition and linking of concepts as in DBpedia, is required to understand how scientific knowledge is constructed from data points. And with a focus on these individual entities, all researchers should adopt the philological stance.

References

1. Assmann, A.: Canon and archive. In: Erll, A., Nünning, A. (eds.) *Cultural Memory Studies: An International and Interdisciplinary Handbook*, pp. 97–107. Mouton de Gruyter, Berlin (2008)
2. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data. In: Aberer, K., et al. (eds.) *ASWC/ISWC -2007*. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76298-0_52
3. Auer, S., Kovtun, V., Prinz, M., Kasprzik, A., Stocker, M., Vidal, M.E.: Towards a knowledge graph for science. In: *WIMS 2018* (2018)
4. Austin, P.K.: Data and language documentation. In: Gippert, J., Himmelmann, N.P., Mosel, U. (eds.) *Essentials of Language Documentation*, pp. 87–112. Mouton de Gruyter, Berlin (2006)
5. Austin, P.K.: Language documentation and meta-documentation. In: Jones, M., Ogilvie, S. (eds.) *Keeping Languages Alive. Documentation, Pedagogy, and Revitalisation*, pp. 3–15. Cambridge University Press (2013)
6. Bond, G.C.: Fieldnotes: research in past occurrences. In: Sanjek, R. (ed.) *Fieldnotes. The Makings of Anthropology*, pp. 273–289. Cornell University Press, Ithaca (1990)
7. Catford, J.C.: *A Linguistic Theory of Translation: An Essay in Applied Linguistics*. Oxford University Press, Oxford (1965)
8. Crane, G., Bamman, D., Jones, A.: ePhilology: when the books talk to their readers. In: Schreibman, S., Siemens, R. (eds.) *A Companion to Digital Literary Studies*. Blackwell, Oxford (2008)
9. Cummings, J.: The text encoding initiative and the study of literature. In: Schreibman, S., Siemens, R. (eds.) *A Companion to Digital Literary Studies*. Blackwell, Oxford (2008)
10. Dong, L., Ilieva, P., Medeiros, A.: Data dreams: planning for the future of historical medical documents. *J. Med. Libr. Assoc.* **106**(4), 547–551 (2018). <https://doi.org/10.5195/jmla.2018.444>
11. Gipp, B.: *Citation-Based Plagiarism Detection*. Springer, Wiesbaden (2014). <https://doi.org/10.1007/978-3-658-06394-8>
12. Gurd, S.: Philology and greek literature (March 2015). <https://www.oxfordhandbooks.com/view/10.1093/oxfordhb/9780199935390.001.0001/oxfordhb-9780199935390-e-65>
13. Henk, V., Vahdati, S., Nayyeri, M., Ali, M., Yazdi, H.S., Lehmann, J.: Metaresearch recommendations using knowledge graph embeddings. In: *AAAI 2019 Workshop on Recommender Systems and Natural Language Processing* (2019)
14. Latour, B., Woolgar, S.: *Laboratory Life: The Construction of Scientific Facts*. Princeton University Press, Princeton (1986)
15. Lehmann, C.: Data in linguistics. *Linguist. Rev.* **3/4**(21), 275–310 (2004)
16. Leydesdorff, L., Milojević, S.: Scientometrics. In: Wright, J.D. (ed.) *International Encyclopedia of the Social & Behavioral Sciences*, second edn., pp. 322–327. Elsevier, Oxford (2015). <http://www.sciencedirect.com/science/article/pii/B978008097086850308>

17. McGann, J.: Philology in a new key. *Crit. Inq.* **39**(2), 327–346 (2013)
18. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. *Linguisticae Investig.* **30**(1), 3–26 (2007)
19. Nathan, D., Austin, P.K.: Reconcepting metadata: language documentation through thick and thin. *Lang. Doc. Descr.* **2**, 179–187 (2004)
20. Petőfi, J.S.: Text-grammars, text-theory and the theory of literature. *Poetics* **2**(3), 36–76 (1973)
21. Popping, R.: Knowledge graphs and network text analysis. *Soc. Sci. Inf.* **42**(1), 91–106 (2003)
22. Renear, A.H.: Text encoding. In: Schreibman, S., Siemens, R., Unsworth, J. (eds.) *A Companion to Digital Humanities*. Blackwell, Oxford (2004)
23. Renear, A.H., Wickett, K.M.: There are no documents. In: *Proceedings of Balisage: The Markup Conference 2010*. Balisage Series on Markup Technologies, vol. 5 (2010). <https://doi.org/10.4242/BalisageVol5.Renear01>
24. Risam, R.: Telling untold stories: digital textual recovery methods. In: levenberg, Neilson, T., Rheams, D. (eds.) *Research Methods for the Digital Humanities*, pp. 309–318. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96713-4_17
25. Seidel, F.: Documentary linguistics: a language philology of the 21st century. *Lang. Doc. Descr.* **13**, 23–63 (2016)
26. Shils, E.: *Tradition*. The University of Chicago Press, Chicago (1981)
27. The TEI Consortium: TEI P5: guidelines for electronic text encoding and interchange (2020). www.tei-c.org
28. Thieberger, N.: Research methods in recording oral tradition: choosing between the evanescence of the digital or the senescence of the analog. In: levenberg, Neilson, T., Rheams, D. (eds.) *Research Methods for the Digital Humanities*, pp. 233–241. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96713-4_13
29. Turner, J.: *Philology: The Forgotten Origins of the Modern Humanities*. The William G. Bowen Series, vol. 70. Princeton University Press, Princeton (2014)
30. Web of Science. <http://wokinfo.com/>
31. Weber, T.: Can computational meta-documentary linguistics provide for accountability and offer an alternative to “reproducibility” in linguistics. In: Eskevich, M., et al. (eds.) *2nd Conference on Language, Data and Knowledge (LDK 2019)*. OpenAccess Series in Informatics (OASISs), vol. 70, pp. 26:1–26:8. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2019). <https://doi.org/10.4230/OASISs.LDK.2019.26>
32. Wenzel, S.: Reflections on (new) philology. *Speculum* **65**, 11–18 (1990)
33. Woodbury, A.C.: Defining documentary linguistics. In: Austin, P.K. (ed.) *Language Documentation and Description*, vol. 1, pp. 35–51. SOAS, London (2003)

**2nd Workshop of BI and Big Data
Applications (BBIGAP 2020)**



A Scored Semantic Cache Replacement Strategy for Mobile Cloud Database Systems

Zachary Arani¹(✉), Drake Chapman¹, Chenxiao Wang¹, Le Gruenwald¹,
Laurent d’Orazio², and Taras Basiuk¹

¹ The University of Oklahoma, Norman, OK 73019, USA
{myrrhman, christopher.d.chapman, chenxiao, ggruenwald,
taras.basiuk}@ou.edu

² Univ. Rennes, CNRS, IRISA, Lannion, France
laurent.dorazio@univ-rennes1.fr

Abstract. Current mobile cloud database systems are widespread and require special considerations for mobile devices. Although many systems rely on numerous metrics for use and optimization, few systems leverage metrics for decisional cache replacement on the mobile device. In this paper we introduce the Lowest Scored Replacement (LSR) policy—a novel cache replacement policy based on a predefined score which leverages contextual mobile data and user preferences for decisional replacement. We show an implementation of the policy using our previously proposed MOCCAD-Cache as our decisional semantic cache and our Normalized Weighted Sum Algorithm (NWSA) as a score basis. Our score normalization is based on the factors of query response time, energy spent on mobile device, and monetary cost to be paid to a cloud provider. We then demonstrate a relevant scenario for LSR, where it excels in comparison to the Least Recently Used (LRU) and Least Frequently Used (LFU) cache replacement policies.

Keywords: Big data · Cloud computing · Caching

1 Introduction

Since a cache has limited space, it is important to use replacement policies which keep relevant data on a mobile device. In a mobile cloud database system, querying the cloud can often be an expensive operation in regards to time, money paid to a cloud provider, and mobile device energy. For this reason, leveraging a cache grants large boosts in efficiency. The rudimentary Least Recently Used (LRU) policy—which discards the least recently accessed entry when filled—is often implemented in caches. The similar Least Frequently Used (LFU) policy—which replaces the least frequently used entry when full—is also commonly implemented; however, LRU and LFU are not always the most efficient policies

within the context of a relational database system [3, 8, 12, 13]. Instead, many Database Management Systems (DBMS) implement specific replacement policies that cater to the system's needs.

This paper seeks to describe a cache replacement policy for mobile cloud database systems that utilizes decisional semantic caching. We propose the Low-est Scored Replacement policy (LSR), which takes cache relevancy and mobile constraints into account while maintaining a LRU-like overhead. LSR partitions cache events into point scoring categories and utilizes a predefined score based on decisional semantic caching and the Normalized Weighted Sum Algorithm. LSR defines semantic relevancy within the cache while also taking into account user preferences on saving time, energy on mobile device, and cost paid to cloud providers. We then show an implementation of LSR using our existing decisional semantic cache system and demonstrate a relevant scenario for the algorithm. We find from our experiment that there exist scenarios where LSR significantly outperforms the common LRU and LFU cache replacement policies in the mobile cloud database environment.

2 Related Work

Device status and metadata is imperative in mobile cloud database decision making. Metrics such as current battery life, location, and connectivity quality may be leveraged to contextualize computational tasks. Mobile devices are, by their very nature, constrained through their short-term battery life and variable connection to wireless networks. Several replacement policies have been developed to address these issues [1]. These policies make use of metrics such as location, battery life, and on-device data size [2, 7, 15]. Other caching systems, such as semantic caching [11], have also been used to address constraints in a mobile cloud database environment. In our previous work we developed the decisional semantic MOCCAD-Cache [10] as well as the Normalized Weighted Sum Algorithm (NWSA) [4] to better meet constraints on a mobile device. However, our previous work did not propose any solution for a cache replacement policy.

One cache replacement approach that bears some similarities to ours is frequency-based. This method considers each cache entry's access frequency when performing decisional replacement. Examples are found in [6] where three different methods are proposed—each with its own contextual merits. These policies—*The mean scheme*, *The window scheme*, and *The exponentially weighted moving average scheme*—bear some resemblance to ours since they take into account how recently a cache entry was accessed; however, these cache replacement strategies do not take into account semantic information or mobile constraints such as the device's battery life.

There exist other policies which take data size and cloud retrieval cost into account. One example is the SAIU (*Stretch Access-rate Inverse Update-frequency*) replacement policy proposed in [14], as well as a modified version in [5]. The SAIU defines a gain function based on access rate, update frequency, data size, data retrieval time, and (in the latter paper) consistency. The policy

then uses this weight to determine replacement. A version proposed in [9] uses a more generalized cost function. By taking the various costs of each query into account, these methods for cache replacement share some similarity with ours; however, these methods do not factor the benefits of semantic caching into their replacement process.

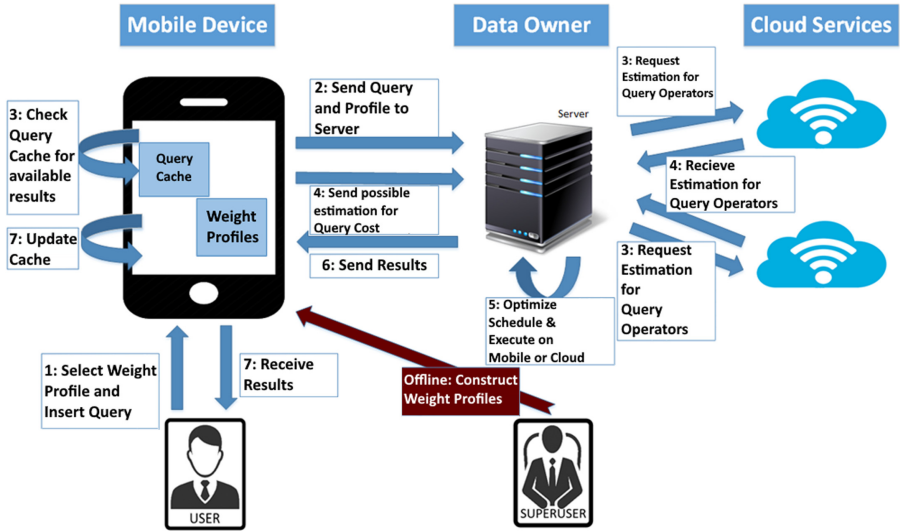


Fig. 1. Our mobile cloud database environment

3 Our Mobile Cloud Database System

Figure 1 details our system architecture and illustrates the possible workflows for query execution. In our mobile cloud database environment, a user sends queries and constraints (in the form of a weight profile) from the mobile device for execution. User constraints are defined by their interest in conserving time, mobile device energy, or monetary cost paid to their cloud provider. Our system decides how to best execute queries with respect to user constraints: either through the cloud or from a local cache if possible. The user’s query and weight profile is sent through some form of infrastructure (the data owner) to cloud services for query execution estimations. The data owner then uses these estimations to decisionally optimize query execution on some combination of the mobile device and the cloud.

3.1 MOCCAD-Cache

Our previously proposed MOCCAD-Cache introduces the concept of decisional semantic caching [10] to solve issues with semantic caching in mobile contexts.

In a semantic caching system, both data and the semantic description of that data are provided to the cache [11]. For instance, the result of a query will be stored along with what relations, attributes, and predicates the query consisted of. This means new queries can be compared against semantic descriptions of stored queries before execution.

Many systems only detect a cache hit event if both the input query and the cache query match exactly. Semantic caching instead allows recognition of overlaps in semantic descriptions. If an input query has overlap in the relations, attributes, and predicates of a cached query, it will be recognized as either a *partial hit* (some data in the cache) or an *extended hit* (all data in cache). If the query semantics match exactly or do not match at all, a cache hit or miss is detected. If the parser detects some semantic overlaps, a query trimmer is used to create two new queries for execution. For example, if only part of the input query is stored in the cache—a partial hit event—the query trimmer will transform the input query into a mobile device query (probe query) and a cloud query (remainder query) for execution. If the input query’s semantic description is fully overlapped with cached semantics, it is recognized as an extended hit event; the query is then executed on the mobile device using cached data. Semantic caching is useful in many cases, especially on a device with limitations such as a mobile phone. However, there are some instances where local query trimming and execution is considerably worse than a simple cloud execution. After trimming the query, the MOCCAD-Cache runs estimations to decide whether to execute the query on the cloud or mobile device—if possible. Despite needing to go through an entirely new estimation phase, this system can improve query processing time (albeit with the general caveat of increased monetary cost).

3.2 Normalized Weighted Sum Algorithm

MOCCAD-Cache may outperform semantic caching in terms of time, but there are many scenarios where time is not the only imperative metric. Our later proposed Normalized Weighted Sum Algorithm (NWSA) addresses this issue by taking into account arbitrary metrics and normalizing them to calculate a score [4]. The NWSA calculates a score based in part on the user’s interest in saving time, energy, or monetary cost¹. These constraints are evaluated against query execution plans (QEPs) to decide a QEP that best respects the user’s desires.

$$\begin{aligned} QEP_1 &: \{M = \$0.080; T = 0.5s; E = 0.012mA\} \\ QEP_2 &: \{M = \$0.050; T = 3.0s; E = 0.300mA\} \\ QEP_3 &: \{M = \$0.055; T = 0.6s; E = 0.013mA\} \end{aligned}$$

Fig. 2. Example query execution plans (QEPs) for a sample query

¹ Each parameter is given on a scale from 0 to 1, where the sum of all parameters must total 1.

$$QEP_{score} = \min_i \sum_{j=1}^n w_j \frac{a_{ij}}{m_j}$$

Fig. 3. Equation for finding the best QEP score using the normalized weighted sum algorithm

$$w_j = \frac{uw_j * ew_j}{\sum uw * ew}$$

Fig. 4. Equation for composite normalized weight factor

The MOCCAD-Cache estimates the efficiency of executing a query on the cloud versus the mobile device. If the query is performed on the cloud, there are several QEPs that can be executed—each with its own costs. Some example QEPs for a given query are shown in Fig. 2. The NWSA takes each QEP and scores how it respects user constraints. The lowest scoring QEP signifies the most efficient execution respecting user constraints. For instance, a user giving priority to time and monetary cost would result in QEP_3 from Fig. 2 being executed, as it respects those two constraints the most.

The methodology of QEP scoring is shown in Fig. 3. The NWSA looks at each QEP and scores it based on three factors. a_{ij} is the i th QEP’s estimated cost for the j th constraint (money, time, energy). m_j is the maximum accepted value for the j th constraint. Any QEP with a constraint value higher than m_j is not considered for best score. Figure 4 describes w_j , a composite normalized weight factor derived from the user constraints (uw) as well as a device’s environmental factors such as battery life or network connectivity (ew).

In summary, the MOCCAD-Cache with the NWSA dictates the structure of cache entries and how to handle new cache events, but it does not detail any methods of cache replacement and may assume an infinite cache size. In Sect. 4 we propose a novel replacement policy leveraging data calculated by the MOCCAD-Cache using NWSA to efficiently replace entries while respecting constraints.

4 The Lowest Scored Replacement (LSR) Policy

This section explores our proposed cache replacement policy and its implementation. The Lowest Scored Replacement Policy (LSR) utilizes the QEP score calculated by the NWSA as well as cache events defined by the MOCCAD-Cache. A modified QEP score along with MOCCAD-Cache cache events score each cache entry for decisional replacement.

4.1 LSR Score

The initial LSR score of a cache entry is based on the QEP score for each query. Higher QEP scores indicate that a given query is more difficult to retrieve from the cloud, and therefore may be more valuable to keep in the cache. For the sake of precision and arithmetic simplicity, we keep baseline LSR scores close to the magnitude of 1. This requires the given QEP score to be multiplied by a scaling factor before being considered as the LSR score.

$$LSR_{Score} = QEP_{Score} * ScaleFactor$$

4.2 Scoring System

After being initialized, an entry’s score is updated by cache events. The most desirable cache event—a cache hit—is rewarded a *FULL-POINT* whereas the least desirable outcome is given a *ZERO-POINT* score. Scores for cache events involving query trimming fall between these two extremes. The extended hit cache event (all relevant data in cache) is rewarded a *HALF-POINT*, as this event generally does not involve accessing the cloud. The partial hit cache event (some relevant data in cache) is rewarded a *QUARTER-POINT*, since some local data is often more desirable than a cache miss.

Since LSR rewards cache entries based on accessed data, it is worth noting its similarity to LFU. Unlike LFU, LSR takes into account the constraints of the mobile device (the QEP score) along with the query’s semantic utility in the cache (MOCCAD-Cache events). This functionality is crucial for mobile cloud computing, where devices are constrained by limited resources and user requirements. In short, LSR inherently respects the constraints of mobile computing, unlike LFU.

4.3 Cache Implementation

The cache is implemented as a minimum priority queue. Entries are initialized with a given score and are updated as cache events occur; the cache entry score dictates the priority within the queue. When an entry needs to be removed, the lowest scoring entry is replaced.

5 Experimentation and Results

In this section we discuss our experiments—and the methodology behind them—as well as their results. We have conducted an experiment to compare the performance of our proposed algorithm, LSR, against the ubiquitous LRU and LFU policies. We compare these replacement policies in terms of the monetary cost, query response time, and energy consumption.

Table 1. Summary of static experiment parameters

Static parameters	Value	Reference
LG V10 memory	2 GB	Kernel
LG V10 SoC CPU active mode frequency	1.728 GHz	Kernel
LG V10 SoC CPU active mode current	157.44 mA	Power profile
LG V10 SoC CPU idle mode frequency	0	Kernel
LG V10 SoC CPU idle mode current	16.4	Power profile
LG V10 SoC Wi-Fi network low current	0.1 mA	Power profile
LG V10 SoC Wi-Fi network high current	60 mA	Power profile
LG V10 battery capacity	3000 mAh	Power profile
LG V10 average bandwidth up	8.47 Mbps	Google speedtest
LG V10 average bandwidth down	12.9 Mbps	Google speedtest
Cloud node memory	16 GB	Kernel
Cloud CPU	Intel i7-8750H @ 2.20 GHz	Kernel
Cloud node disk	512 GB Samsung 970 EVO NVMe PCIe M.2-2280 SSD	Kernel
Cloud node average bandwidth up	3.28 Mbps	Google speedtest
Cloud node average bandwidth down	28.0 Mbps	Google speedtest
Query cache maximum size	100 MB	
Query cache maximum entries	10	
Query set size	20 queries	
Dataset	TPC-H benchmark	TPC
Number of relations	8	TPC
Database size	2 GB	
<i>ScaleFactor</i>	10^{12}	QEP score magnitude

5.1 Experimentation Hardware and Software

In order to simulate a cloud environment, a single node was set up using the Hadoop framework and data warehouse infrastructure. Apache Hive was used as the database system along with MySQL for storing metadata. The node ran the Arch Linux operating system and featured an Intel Core i7-8750H CPU running at 2.20 GHz as well as 16 GB of RAM and a 512 GB Samsung 970 EVO NVMe PCIe M.2-2280 SSD. A RESTful java web servlet running on Apache Tomcat 8.5 was used to access the cloud infrastructure from mobile devices. This servlet is able to retrieve tuples, query cost estimations, and relational metadata from the hive server.² The mobile device ran a development branch of the Java based MOCCAD-Cache Android prototype that features an LSR implementation and other small improvements.³

The mobile device used for testing was a LG V10 Android device⁴, which featured a hexa-core Qualcomm MSM8992 Snapdragon 808, 2 GB of RAM, and a battery capacity of 3000 mAh. Table 1 summarizes the experiment parameters.

² The source code for the cloud web service can be found at <https://github.com/ZachArani/CloudWebService>.

³ The source code of MOCCAD-Cache and the NWSA can be found at <http://cs.ou.edu/~database/MOCCAD/index.php>. This experiment was conducted on the ‘dev’ branch.

⁴ Model *LG-H900*.

5.2 Example Scenario

In this section, we outline a relevant scenario for LSR. We envision a business worker using a mobile device to access company information while away from the office—where productivity may be affected by mobile constraints. The worker spends time focusing on one business context before switching to another. They may execute several queries all related to one product, customer, or region before suddenly switching to a different one. This means that variations of a complex query will be executed several times in sequence before completely unrelated queries are executed. The worker may then return to the original context they were working in later.

In this scenario, there is a full cache with an entry that is semantically useful but is difficult to retrieve from the cloud. If the entry has not been used for a short period (a context switch), LRU and LFU will remove it. These policies remove the useful entry because they do not respect semantic utility or mobile constraints but instead only respect recent accesses. If a query from the original context is then executed, it will be very expensive in terms of time, money, and energy for the LRU and LFU users. Unlike LRU or LFU, LSR will respect the constraints of the mobile cloud database environment and retain the entry for continued use.

5.3 Experimentation Methodology

In order to simulate the database environment of a business worker, we generated a 2 GB database based on the TPC-H model, which is structured to simulate business data.⁵

Before running the experiment, a series of ten *warmup queries* were executed to fill the cache with data prior to the experiment. These queries simulate previous contexts unrelated to the experimental ones in order to encourage cache replacement. We then ran twenty queries for this experiment, starting with a costly query:

```
SELECT DISTINCT l.shipdate FROM lineitem WHERE l.linestatus = 'O';
```

After this, we ran several semantic hits (extended and partial) in the same context before switching to unrelated queries of a different context. After several queries are run in this context, the original query and semantic hits were then run again near the end of the workload. The experiment measured total execution time, energy spent on mobile device, and estimated cost paid to the cloud provider. The experiment was run three times, with the results being averaged. The MOCCAD-Cache prototype's user preference weights for money, energy, and time were all set to an equal one-third amount.

⁵ *hive-testbench* by HortonWorks was used for database creation. It can be accessed at <https://github.com/hortonworks/hive-testbench>.

5.4 Results

Figures 5, 6, and 7 detail the results of our experimentation. As we expected, LSR significantly outperformed LRU and LFU in the business scenario. LSR performed over twice as fast, cheap, and energy efficient when compared to LRU. LFU managed to be somewhat competitive in cost, but still was eclipsed by LSR in speed and energy efficiency. In terms of all three metrics, LSR was clearly cheaper, faster, and more efficient in the mobile cloud database environment. Our policy used valuable metrics to recognize the utility of data as well as respect the constraints of the mobile device. Even though LFU and LRU may have only needed to run a handful of additional queries on the cloud—the re-execution of large or costly queries will not respect the constraints of a mobile device. These costs may be greatly exacerbated depending on the device’s particular context.

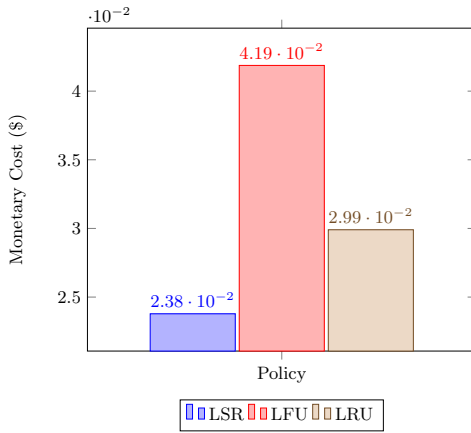


Fig. 5. Scenario monetary cost performance

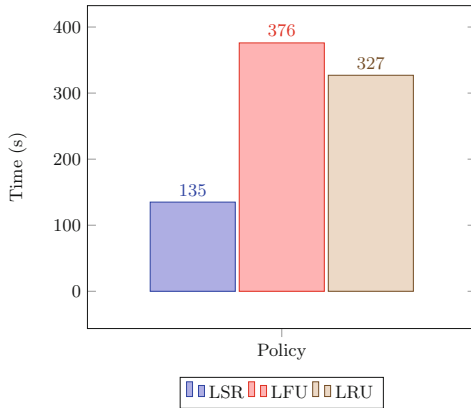


Fig. 6. Scenario time performance

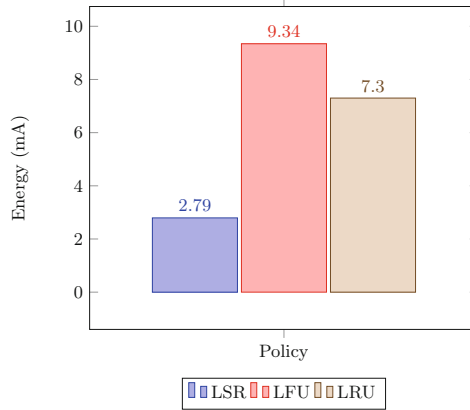


Fig. 7. Scenario energy performance

LSR, by comparison, leveraged user constraints and decisional semantic cache events to intelligently retain valuable data locally.

6 Conclusions

Mobile cloud database systems have become ubiquitous in recent memory. Several advancements have been made in the field, such as our previously proposed decisional semantic MOCCAD-Cache as well as the Normalized Weighted Sum Algorithm. LSR attempts to improve on existing cache replacement strategies in mobile cloud database systems by accounting for mobile device constraints and semantic utility. When combined with the MOCCAD-Cache and the Normalized Weighted Sum Algorithm, there exist scenarios where LSR significantly outperforms LFU and LRU in a mobile cloud database environment.

Although LSR's initial experiments are promising, future work is needed to better determine its use and applicability in the mobile cloud database setting. Experiments could be run to see how LSR compares against other common non-LRU based cache replacement policies. Experiments that vary the size of the cache would prove insightful on how useful LSR would be in other caching environments. On top of this, varying user preference weights for time, money, and energy for NWSA scoring may provide interesting results. Additional workloads and scenarios must be analyzed to investigate where LSR is most applicable to real world applications. Finally, implementing the LSR policy in non-mobile cloud database systems may also yield promising results in testing the policy's utility in other areas.

Acknowledgment. This work is partially supported by the National Science Foundation Award No. 1349285.

References

1. Barbará, D., Imieliński, T.: Sleepers and workaholics: caching strategies in mobile environments (extended version). *VLDB J.* **4**(4), 567–602 (1995). <http://dl.acm.org/citation.cfm?id=615232.615236>
2. Chand, N., Joshi, R.C., Misra, M.: Cooperative caching strategy in mobile ad hoc networks based on clusters. *Wirel. Pers. Commun.* **43**(1), 41–63 (2007). <https://doi.org/10.1007/s11277-006-9238-z>
3. Chou, H.T., DeWitt, D.J.: An evaluation of buffer management strategies for relational database systems. In: Proceedings of the 11th International Conference on Very Large Data Bases, VLDB 1985, vol. 11, pp. 127–141. VLDB Endowment (1985). <http://dl.acm.org/citation.cfm?id=1286760.1286772>
4. Helff, F., Gruenwald, L., d’Orazio, L.: Weighted sum model for multi-objective query optimization for mobile-cloud database environments. In: EDBT/ICDT Workshops (2016)
5. Xu, J., Hu, Q., Lee, W.C., Lee, D.L.: Performance evaluation of an optimal cache replacement policy for wireless data dissemination. *IEEE Trans. Knowl. Data Eng.* **16**(1), 125–139 (2004). <https://doi.org/10.1109/TKDE.2004.1264827>
6. Leong, H.V., Si, A.: On adaptive caching in mobile databases. In: Proceedings of the 1997 ACM Symposium on Applied Computing, SAC 1997, pp. 302–309. ACM, New York (1997). <https://doi.org/10.1145/331697.331760>
7. Li, W., Chan, E., Chen, D.: Energy-efficient cache replacement policies for cooperative caching in mobile ad hoc network. In: 2007 IEEE Wireless Communications and Networking Conference, pp. 3347–3352 (March 2007). <https://doi.org/10.1109/WCNC.2007.616>
8. Li, Z., Jin, P., Su, X., Cui, K., Yue, L.: CCF-LRU: a new buffer replacement algorithm for flash memory. *IEEE Trans. Consum. Electron.* **55**(3), 1351–1359 (2009). <https://doi.org/10.1109/TCE.2009.5277999>
9. Yin, L., Cao, G., Cai, Y.: A generalized target-driven cache replacement policy for mobile environments. In: 2003 Symposium on Applications and the Internet, 2003, Proceedings, pp. 14–21 (January 2003). <https://doi.org/10.1109/SAINT.2003.1183028>
10. Perrin, M., Mullen, J., Helff, F., Gruenwald, L., d’Orazio, L.: Time-, energy-, and monetary cost-aware cache design for a mobile-cloud database system. In: Wang, F., Luo, G., Weng, C., Khan, A., Mitra, P., Yu, C. (eds.) Big-O(Q)/DMAH - 2015. LNCS, vol. 9579, pp. 71–85. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-41576-5_6
11. Ren, Q., Dunham, M.H., Kumar, V.: Semantic caching and query processing. *IEEE Trans. Knowl. Data Eng.* **15**(1), 192–210 (2003). <https://doi.org/10.1109/TKDE.2003.1161590>
12. Sacco, G.M., Schkolnick, M.: Buffer management in relational database systems. *ACM Trans. Database Syst.* **11**(4), 473–498 (1986). <https://doi.org/10.1145/7239.7336>
13. Stonebraker, M.: Operating system support for database management. *Commun. ACM* **24**(7), 412–418 (1981). <https://doi.org/10.1145/358699.358703>

14. Xu, J., Hu, Q., Lee, D., Lee, W.C.: SAIU: an efficient cache replacement policy for wireless on-demand broadcasts. In: Proceedings of Ninth ACM International Conference on Information and Knowledge Management (August 2000). <https://doi.org/10.1145/354756.354785>
15. Yin, L., Cao, G.: Supporting cooperative caching in ad hoc networks. In: IEEE INFOCOM 2004, vol. 4, pp. 2537–2547 (March 2004). <https://doi.org/10.1109/INFCOM.2004.1354674>



Grid-Based Clustering of Waze Data on a Relational Database

Mariana M. G. Duarte^{1(✉)}, Rebeca Schroeder^{2(✉)}, and Carmem S. Hara^{1(✉)}

¹ Universidade Federal do Paraná, Curitiba, Brazil
{mngduarte, carmem}@inf.ufpr.br

² Universidade do Estado de Santa Catarina, Joinville, Brazil
rebeca.schroeder@udesc.br

Abstract. In the urban environment, data collected from traffic events can serve as elements of study for city planning. The challenge is to transform this raw data into knowledge of mobility. Events are usually stored as individual records in a database system, and urban planning involves costly spatial queries. In this paper, we investigate the effect of a grid-based clustering on the performance of such queries, using an off-the-shelf relational database and index structure. We report on the results of this approach using data collected from Waze over a period of one year. We compare the performance of our grid-based approach with a clustered R-tree index over the geometric attribute. The results of this study are of interest to developers of applications that involve spatial data over a specific geographic area, using an existing database management system.

1 Introduction

In the urban environment, data collected from traffic events can be used for planning cities and metropolises. Although a lot of data have been collected, the challenge is to transform this set of spatio-temporal data into knowledge of mobility to assist this planning. Traffic events, such as jams, alerts and irregularities, are continuously produced by applications such as Waze, making the data set increasingly larger. Due to the speed at which this data is reported, these applications generally store events as individual records. For processing spatio-temporal queries, in order to avoid an exhaustive search that scans the entire database, index structures are usually used. The traditional structures adopted for spatial indexing are R-Trees ([5, 9]) and KD-Trees ([2, 4]).

This paper proposes the partitioning of a geographic area of interest, creating a grid composed of juxtaposed Geographic Cells (GCs), as illustrated in Fig. 1. Juxtaposed GCs eliminate the possibility of data belonging to more than one GC since they do not intersect. Events that occurred in the same GC are then stored in a grouped manner, in order to optimize their recovery. Grid-based clustering is not a new idea. It is the basis for some spatial indexing structures proposed in the literature [6, 7]. As opposed to these works, which propose a new

index structure, in this paper we are interested in determining the effect of a grid-based clustering on the performance of spatial queries, using an off-the-shelf relational database and a traditional R-tree indexing. The results of this study is of interest to developers of applications that involve spatial data over a specific geographic area, using an existing database management system (DBMS).

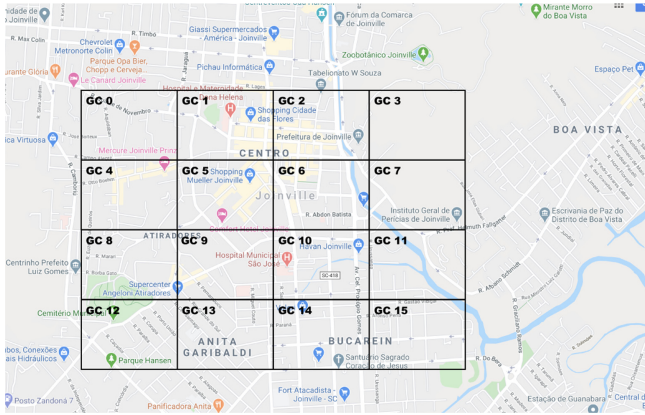


Fig. 1. Grid over a geographic area of interest

As a study case, we apply our approach on data collected by Waze, produced over a period of one year at Joinville, a city in the south of Brazil¹. We thus denote our approach as Waze-GC. We analyze the query performance of Waze-GC on PostGIS, by comparing it with an alternative storage schema, which consists of a clustered R-tree index over the geometric attribute.

The rest of this paper is organized as follows. Section 2 presents and compares approaches related to grid partitioning and indexing of traffic events. Section 3 presents our grid-based clustering approach. Section 4 describes the implementation and the experimental study. Section 5 concludes this paper presenting some future work.

2 Related Work

Grid-based partitioning of data has been applied to several works [6, 7, 10]. Similar to our work, they assume the existence of a delimited geographic area of interest on which a matrix with a predetermined number of cells is created. [6] proposes an index structure for moving objects that combines an R-tree with a grid on the leaves in order to minimize the overlapping among the objects' minimum bounding rectangles (MBR). A similar structure, called GE-Tree [7], is proposed to guarantee constant time to obtain the set of objects within a given

¹ <https://github.com/joinville/Joinville-Smart-Mobility>.

geographic cell. Although the size of the cells is constant and defined beforehand, it proposes the construction of a tree-based index on top of them, with nodes that split on demand. That is, it only creates links to cells that contain data in order to account for the imbalance on the number of objects among cells. GeoSpark [10] applies a grid for a different purpose. It is an in-memory cluster computer framework for processing large-scale spatial data in Apache Spark. A grid is used to partition the data and assign cells to machines for parallel execution. Our work differs on the use of the grid. Here, its purpose is for clustering data stored in a relational database. Waze-CG does not consider new indexing structures such as GE-Tree [7] nor does it consider parallel processing of queries. Waze-GC can be implemented on any traditional relational database, such as the PostGIS extension of Postgres DBMS.

With the rising of urban computing, mobility data management became a very active investigation. Among the works that tackle this problem we mention two: STIG tree [1] and TQ index [3]. Both propose special structures for indexing mobility data that are *not* based on a grid. STIG tree [1] proposes a KD-tree for indexing spatio-temporal data with sets of events on the leaves that can be processed in parallel on GPUs. TQ index [3] was proposed to predict traffic jams based on spatio-temporal traffic data. The model is optimized for analytical queries and maintains two main components: a location index and a time index, which are based on hash tables. Although Waze-GC works with the same type of data as STIG tree and TQ Index, its objective is to determine the impact of clustering traffic events based on a grid using native structures of a database. Waze-GC uses data structures that are already implemented in a relational database, such as a clustered B+ trees and R-trees with no additional structures.

3 Grid-Based Clustering

Our motivation is clustering historic traffic events that occurred in a given area, based on a grid over the area of interest. As a result, it is possible to filter events using spatial information. The grid-based strategy is similar to the matrix representation used by geographic information systems, in which the area of interest is composed of a matrix of cells of regular sizes, each one associated with a set of values that represent geographical characteristics of the region [8].

An *area of interest* has a minimal bounding rectangle, defined by the upper left corner coordinate (lat_{UL} , $long_{UL}$) and the lower right corner coordinate (lat_{LR} , $long_{LR}$). This area can be, for example, a city, a state, or any geographical region. This bounding rectangle is divided into *geographic cells* (*GCs*), which are non-overlapping rectangles of the same size that form a matrix over the area of interest. The limits of each GC are determined by the matrix's number of rows (R) and columns (C). This work considers a linear representation of the matrix to obtain the GC's identifier (*id_{GC}*). That is, GC_n corresponds to the cell in row ($\lfloor n/C \rfloor$) and column ($n \bmod C$) of the matrix. Consider, for example, the area of interest shown in Fig. 1, for which the user defined

that $R = 4$ and $C = 4$. In this example, GC_{11} corresponds to the cell in row $(\lfloor 11/4 \rfloor) = 2$ and column $(11 \bmod 4) = 3$ of the matrix. In our approach, a GC matrix is represented by a GC Table, with $R * C$ records, as shown in the example of Table 1. It consists of GC’s identifier and a rectangle defined by its four limit coordinates, with upper left point, lower left point, lower right point, and upper right point.

Table 1. Waze-GC - GC Table

Id_GC	Geom
1	POLYGON(-49.3 -26.5, -49.2765 -26.5, -49.2765 -26.475, -49.3 -26.475, -49.3 -26.5)
2	POLYGON(-49.3 -26.47, -49.2765 -26.475, -49.2765 -26.45, -49.3 -26.45, -49.3 -26.47)
3	POLYGON(-49.3 -26.45, -49.2765 -26.45, -49.2765 -26.425, 49.3 -26.425, -49.3 -26.45)
...	...

The vertical size of each cell is defined by `SizeLat` and horizontal by `SizeLong` which correspond to $(\text{latUL} - \text{latLR})/R$ and $(\text{longLR} - \text{longUL})/C$, respectively. Each traffic event has a geographical dimension, which may be a point or a set of points defined by their latitude and longitude. In order to identify in which GC the $(\text{lat}, \text{long})$ coordinate of each point is located, we derive its row and column as follows:

$$row = \left\lfloor \frac{\text{latUL} - \text{lat}}{\text{SizeLat}} \right\rfloor \quad column = \left\lfloor \frac{\text{long} - \text{longUL}}{\text{SizeLong}} \right\rfloor$$

Waze-GC represents data collected by Waze as tables for each type of traffic event. Thus, three tables are created in order to store jams, alerts and irregularities. Table 2 shows records of jams, containing the attributes `event ID`, `street`, `pub_utc_date` (date of publication), `id_GC` and `geometry`. The same schema is used to define the tables of alerts and irregularities. Besides, tables are clustered by the attribute `id_GC`. An R-tree is also created on the `geometry` attribute. Therefore, Waze-GC groups records by their spatial proximity, when using the column `id_GC` to define a clustering index for spatial data. The primary key of the table is $(\text{event ID} + \text{id_GC})$. This is because some traffic events, such as jams and irregularities, have a list of points representing its geographical dimension. Since each record should contain only points in a single GC, the list must be split. As an example, consider the event with $\text{ID} = 4$ in Table 2. The points are split into two sequences: the first with points in GC 1 and the second with points in GC 2, generating two records in the table.

The process for generating these records from the original list of points is as follows. Waze-GC analyzes whether two sequential points in the list, say p_n and p_{n+1} , are located in different GCs. If this is the case, it is necessary to divide the list of points. Two records are created, r_1 and r_2 , where r_1 contains the starting point up to p_n , plus an additional point referring to the intersection point of the GC with the line segment formed by p_n and p_{n+1} . This same point is inserted in r_2 , in addition to points starting from p_{n+1} . The original event ID is kept in the table to identify that the points belong to the same event from Waze.

Table 2. Waze-GC - Jams Table

ID	Street	pub_utc_date	Id_GC	Geometry
1	Florianópolis S.	2017-12-15 19:43:43	1	(x -48.833472, y -26.328465), (x -48.837777, y -26.329874)
2	Min. Calógeras S.	2017-12-14 17:35:39	1	(x -48.843751, y -26.30736)
3	BR-101	2017-12-18 18:24:38	1	(x -48.870387, y -26.320411)
4	Min. Calógeras S.	2017-12-14 17:35:39	1	(x -48.843751, y -26.30736)
4	Min. Calógeras S.	2017-12-14 17:35:39	2	(x -49.387950, y -26.30736), (x -49.389090, y -26.30736)
...

Queries executed on Waze-GC can take advantage of the `id_GC` attribute for filtering records occurred in a given area, related to a set of GCs. Besides, query results can be joined from different type of events that occurred in the same GC. For example, events of jams and alerts may be combined if they occurred in the same GC. In the next section, we investigate the effect of Waze-GC on the performance of spatial queries.

4 Experimental Study

The original Waze database is stored in Postgres and was obtained from the *Smart Mobility* project, in development by the city of Joinville-SC². The granting of data for the referred project was provided to the State University of Santa Catarina (UDESC), partner of this work. Such database has 13 Gigabytes (GB), containing data from September 2017 to September 2018. From this database, a new database was created with only the attributes involved in queries of our study case. Besides, a clustered R-tree index was also defined on the *geometry* attribute of Waze. From now on, this new database is called just Waze, while Waze-GC corresponds to the database generated by our approach.

The city of Joinville is the area of interest. We set the grid size to 20 lines and 20 rows, resulting in a grid with 400 geographic cells. The area of interest is a rectangle delimited by the coordinates $(-48.72, -26.39)$ and $(-48.92, -26.2)$ for the upper left corner and lower right corner, respectively. The total area has 625 km^2 , with cells of 1.56 km^2 . We have created datasets *B20*, *B30*, *B40*, *B50* and *B100*, which correspond to 20%, 30%, 40%, 50% and 100% of Waze. These datasets were created incrementally, starting with events that occurred in the *central* region of the city, since it contains a larger concentration of traffic events. Thus, after generating *B20*, dataset *B30* contains all the records in *B20* plus the next 10% of events. This process continues until *B100*, which contains all the events. Table 3 shows the number of records in each dataset, for each *type* of event in Waze and Waze-GC. The number of GCs covered by Waze-GC for each percentage is shown in the last column. The amount of alerts is the same in both databases given that only one point represents their geographical dimension.

² <https://github.com/joinville/Joinville-Smart-Mobility>.

However, Waze-GC has a larger number of records of jams and irregularities. This is because they are represented by a set of points which may be split into different GCs.

Table 3. Number of records for each type of event

Database	Percentage	# Alerts	Waze		Waze-GC		
			# Jams	# Irregularities	# Jams	# Irregularities	# GCs
B20	20%	1024311	587816	22063	588616	22464	81
B30	30%	1536466	887724	33095	888928	33696	120
B40	40%	2048622	1183632	44126	1185230	44926	161
B50	50%	2560777	1479540	55158	1481544	65159	198
B100	100%	5121554	2959080	110316	2963087	112321	400

Figure 2 shows the data loading time in Waze and Waze-GC. Waze-GC time consists of insertions on the database with a stored procedure that checks which GCs the record intersects and splits the list of points when needed. This process doubles the load time if compared to Waze. All experiments were conducted on a machine running Mac OS 10.15.2 on Dual-Core Intel Core m3 with 1.1 GHz and 8 GB of main memory.

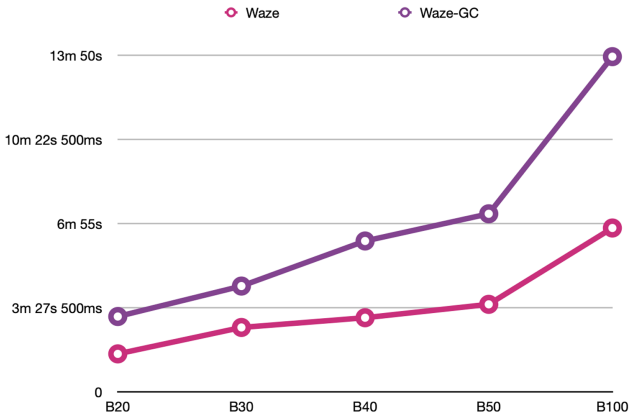


Fig. 2. Data loading time

We compare Waze and Waze-GC with two queries. We have executed each query five times, and the reported values consist of their average. The first query was defined to analyze the impact of clustering events by `id_GC` in Waze-GC. The query statement for Waze-GC is shown below and aims to answer “Which streets had traffic jam and alert events that occurred at exactly the same point on a street in the first seven days of 2018?”. A similar statement was defined for Waze, without the filter applied on `id_GC`.

Query 1 on Waze-GC:

```

SELECT i.street
FROM alerts i, jams j
where i.id_gc=j.id_gc
      and i.pub_utc_date > '2017-12-31 23:59:59'
      and j.pub_utc_date > '2017-12-31 23:59:59'
      and i.pub_utc_date < '2018-01-08 00:00:00'
      and j.pub_utc_date < '2018-01-08 00:00:00'
      and i.name=j.name
      and ST_Intersects(i.geometry,j.geometry);

```

Waze and Waze-GC use the spatial function `ST_Intersects` from PostGIS, which tests whether the geometries of alerts and jams intersect. In both databases, the R-Tree index defined on attribute `geometry` is useful to enhance the performance of `ST_Intersects`. Figure 3 presents the results of query 1 for databases B20-B100. Columns *Waze* and *Waze-GC* show the respective query processing time, and column *Results* shows the number of records returned. It is possible to observe that *Waze-GC* has an advantage in the query processing time compared to *Waze*. In *Waze-GC* the advantage is given by the clustering on `id_GC`. It improves the selectivity of the join condition given that alerts and jams must be in the same GC. It means that, the R-tree index clustering to process `ST_Intersects` in Waze was less effective than in Waze-GC. This is because Waze-GC only compares records in the same GC.

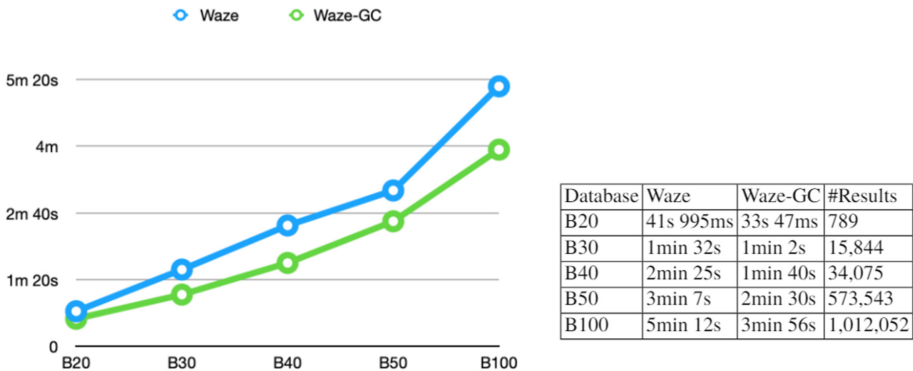


Fig. 3. Query 1 - Response time and # of results

Query 2 was defined to identify the impact of using the GC table and the `id_GC` index by Waze-GC. This query returns “the number of traffic jams in October 2017 in an informed area of interest”. The size of the area of interest is 6,25 km², and the query statements are shown below:

Query 2 on Waze:

```
SELECT COUNT(street)
FROM waze.jams
WHERE pub_utc_date > '2017-09-30 23:59:59'
    and pub_utc_date < '2017-11-01 00:00:00'
    and ST_Intersects(geometry,
        POLYGON(-48.85 -26.35, -48.80 -26.35,
            -48.80 -26.3,-48.85 -26.3,-48.85 -26.35));
```

Query 2 on Waze-GC:

```
SELECT COUNT(street)
FROM jams
WHERE id_gc IN (SELECT i.id_gc FROM gc_table i
    WHERE ST_Intersects(geometry,
        'POLYGON(-48.85 -26.35,-48.80 -26.35,
            -48.80 -26.3,-48.85 -26.3,-48.85 -26.35)')
    and pub_utc_date > '2017-09-30 23:59:59'
    and pub_utc_date < '2017-11-01 00:00:00'
    and ST_Intersects(geometry,
        'POLYGON(-48.85 -26.35,-48.80 -26.35,
            -48.80 -26.3,-48.85 -26.3,-48.85 -26.35)'));
```

Figure 4 presents results for Query 2 on B20-B100 datasets. In Waze-GC statement, the sub-query identifies the GCs in GC Table that intersect the area of interest, which in this case returns only 4 GCs for all percentages (B20–B100). Again the advantage of Waze-GC is given by the filter on `id_gc`, which reduces the search space. However, in this case, GC Table makes the statement more complex than Query 1, where the use of `id_gc` is straightforward.

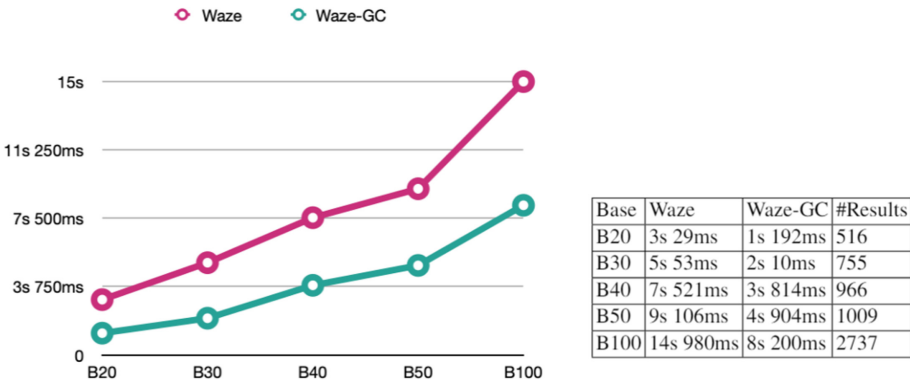


Fig. 4. Query 2 - Response time and # of results

5 Conclusion

We have proposed an approach for partitioning a geographic area of interest, creating a grid composed of juxtaposed geographic cells (GCs). Juxtaposed GCs eliminate the possibility of data belonging to more than one GC since they do not intersect. Events that occurred in the same GC are then stored in a grouped manner, in order to optimize their recovery. As opposed to related work, in this paper we investigated the effect of a grid clustering on the performance of spatial queries, using an off-the-shelf relational database and index structure.

Our study case considered traffic events collected by Waze. Two forms of data storage were tested: (1) a relational database, namely Waze, with a clustered R-tree index on the geometry of events and (2) the same relational database, increased with the respective GC as an additional attribute for each event which is called Waze-GC. In Waze-GC records were clustered by GC, that is, their spatial proximity. We have tested the approaches on data sets of increasing sizes.

The tests performed showed an advantage in query processing time of Waze-GC, compared to Waze. In the queries used in our experimental study, the filter on GCs reduces the search space and is more effective than an R-tree based clustering. The effects on query processing time is more expressive as the database size increases. The shortcomings of Waze-GC is given by the extra time required on database load in order to set the GC for each event. Besides, for query statements where GC geometry must be filtered, Waze-GC makes the query statement more complex than usual. However, we consider the advantage of Waze-GC in query processing overcomes the shortcomings of its implementation costs. Above all, our approach avoids additional data structures on applying grid-clustering in native structures of a relational database.

As future work, we intend to determine the effect of the cell size to better profile this approach, and also investigate its scalability, by considering longer periods of time and larger geographic areas. Future work also includes investigating storage alternatives to reduce the clustering overhead for inserting new records. Structures such as files bring the possibility to group traffic events based on their spatial-temporal proximity. A comprehensive comparison between alternative methods is required.

References

1. Doraiswamy, H., Vo, H.T., Silva, C.T., Freire, J.: A GPU-based index to support interactive spatio-temporal queries over historical data. In: 2016 IEEE 32nd International Conference on Data Engineering (ICDE). Helsinki, Finland, May 2016
2. ExtremeDB: ExtremeDB Documentation. McObject, 8.0 edn (2018)
3. Imawan, A., Putri, F., Kwon, J.: TiQ: a timeline query processing system over road traffic data. In: 2015 IEEE International Conference on Smart City. Chengdu, China, December 2015
4. Oracle: Data Cartridge Developer's Guide. Oracle, 19c edn (2019)

5. PostgreSQL: PostgreSQL 12.2 Documentation. The PostgreSQL Global Development Group, 12 edn (2020)
6. Rslan, E., Hameed, H.A., Ezzat, E.: Spatial R-Tree index based on grid division for query processing. *Int. J. Database Manage. Syst. (IJDMS)* **9**(6) (2017)
7. Shin, J., Mahmood, A., Aref, W.: An investigation of grid-enabled tree indexes for spatial query processing. In: *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 169–178. November 2019. <https://doi.org/10.1145/3347146.3359384>
8. Silva, R.: Banco De Dados Geográficos: Uma Análise Das Arquiteturas Dual (Spring) E Integrada (Oracle Spatial). Master's thesis, Escola Politécnica da Universidade de São Paulo, São Paulo, SP (2002)
9. SQLite: SQLite Documentation. SQLite, 2.1.0 edn. (2018)
10. Yu, J., Wu, J., Sarwat, M.: Geospark: a cluster computing framework for processing large-scale spatial data. In: *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*. Association for Computing Machinery, New York, NY, USA (2015). <https://doi.org/10.1145/2820783.2820860>



Your Age Revealed by Facebook Picture Metadata

Sanaz Eidizadehakhcheloo^{1(✉)}, Bizhan Alipour Pijani², Abdessamad Imine²,
and Michaël Rusinowitch²

¹ Sapienza Università di Roma, 00185 Roma, Italy

eidizadehakhcheloo.1772528@studenti.uniroma1.it

² Lorraine University, Cnrs, Inria, 54506 Vandœuvre-lès-Nancy, France

{bizhan.alipourpijani, Abdessamad.Imine, Michael.Rusinowitch}@loria.fr

Abstract. Facebook users unknowingly reveal personal information that may help attackers to perpetrate malicious actions. In this paper, we show how sensitive age information of a given target user can be predicted from his/her online pictures. More precisely, we perform age inference attacks by leveraging picture metadata such as (i) alt-texts automatically generated by Facebook to describe the picture content, and (ii) picture reactions (comments and emojis) of other Facebook users. We investigate whether the target's age affects other users' reactions to his/her pictures. Our experiments show that age information can be inferred with *AUC* of 62% by using only alt-texts and with *AUC* of 89% by using combination of alt-texts and users' reactions. Additionally, we present a detailed analysis of *spearman correlation* between reactions of Facebook users and age.

Keywords: Social networks · Privacy · Inference attacks · Facebook

1 Introduction

Facebook allows users to control and customize publicly available personal information. In particular, users have the option of hiding profile attributes (e.g., age, gender, relationship status, sexual preference, and political affiliation) and behavioural records (e.g., group, page) from the public and let the profile visible only to the audience of interest. However, users remain vulnerable to attribute inference attacks where an attacker seeks to illegitimately gain private attributes (e.g., age) of target users using collected publicly available information.

Recent attribute inference attacks leverage target user-generated data such as (i) behavioral records (e.g., liked pages and joined groups) [1], (ii) writing style (e.g., word and emoji usage) [18], and (iii) vicinity network (e.g., friends list) [9]. Since standard inference techniques proceed by analyzing data published by the

This work is supported by DIGITRUST (<http://lue.univ-lorraine.fr/fr/article/digitrust/>).

© Springer Nature Switzerland AG 2020

L. Bellatreche et al. (Eds.): ADBIS/TPDL/EDA 2020 Workshops and Doctoral Consortium, CCIS 1260, pp. 259–270, 2020.

https://doi.org/10.1007/978-3-030-55814-7_22

user, one may believe that being cautious in the writing style or hiding attributes (e.g., paged likes and friend list) from the public prevents an attacker to predict sensitive attributes. Unlike previous works, we investigate the feasibility of age inference attacks on Facebook users from non-user generated data.

With an increase in readily available user-generated content, predicting user attributes is an essential technique for targeted advertising [4]. Typically, important attributes (e.g., age, and gender) that are beneficial for providing personalized services are often not accessible. In [8], the authors collected 479K public Facebook users profiles and revealed that age had the highest privacy value since only 3% of users disclose their age. Moreover, the authors of [7] study age privacy on social networks and report that most users consider age as a private attribute. Here we focus on Facebook, the largest social network these days [10].

Motivation. Although Facebook users tend to hide their attributes, they post pictures to expose their personalities, lifestyles, and preferences [20]. Motivated by this observation, we aim to alert users about the privacy breach caused by picture metadata. Our system can also be applied to improve recommender systems. We consider two types of picture metadata. First, alt-text is some freely available textual information describing picture contents and generated by Facebook image processing software. This generated text purpose is to help blind people using a screen reader. Second, reactions are feeling expressions published by other users when looking at a picture (for example words, emojis, and *GIF* usage in comments). Posted pictures receive many types of reactions (see Fig. 1(a) and (b)). We will show that attacks are still possible when the received comments are non-English ones or even non-textual ones (e.g., emoji-based, see Fig. 1). The danger of picture metadata like alt-text is that inference attacks are still possible even when the user hides all public data (e.g., attributes, and writing style) or pictures receive no reaction at all.

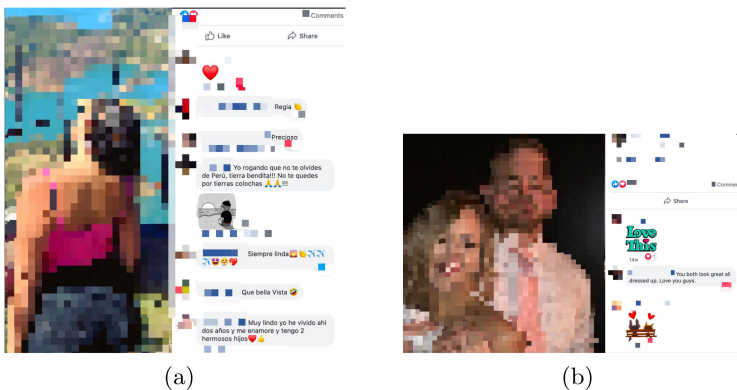


Fig. 1. Picture metadata: (a) Non-English comments (b) English comments.

Problem Statement. The problem is to infer the target user age from published pictures metadata. A challenge is that picture metadata have been added at different moments of the target lifetime and therefore relate to different age categories. The solution is to create different profiles with different age categories. For example, *Alice* can be 31 year old now and has published two pictures. The first one when she was 28, and the second one when she was 31 year old. As a result, she might receive different types of reactions and responses for these two posted pictures based on her picture sharing age. Assigning all target user’s pictures to the same age group may hinder the inference process, as well as the lack of data in some age classes. To circumvent the mentioned problems, we consider the picture publishing time as an important factor to (i) assign pictures to correct age groups, and (ii) increase the number of users in different age groups, when possible. We define this process in more details in Subsect. 4.2.

Contribution. We study the effect of picture owner age on other users’ reactions (e.g., emotional words, emojis usage) when reacting to different age group pictures with the same style and alt-text tags. As posted pictures receive many types of reaction, we construct an age inference attack that depends of the availability of picture metadata (e.g., alt-text or comments). For our attack to be non-trivial we assume: 1) commenters age are hidden; 2) relationship (e.g., friendship, group-ship, etc) between target and commenters are hidden. Unlike previous works:

- 1) We infer user age by leveraging non-user-generated data (picture metadata). This type of attacks show that standard precautionary measures, like hiding friends, attributes or adopting a neutral style in comments are not sufficient for a user to hide age information. As far as we know, our work is the first to design age inference attacks from non-user generated data.
- 2) We analyze the correlation of other users reactions, and alt-text by considering picture owner age. The recognized correlation then helps us to devise attacks with reasonable inference accuracy.

In the following we call *Picture Owner* the Facebook user who published the picture and *Commenter* a Facebook user who comments or reacts to the owner pictures. A commenter can be an owner friend, a friend of an owner friend, or an ordinary user. We call a *Reaction* an emotional response (e.g, posted comment) issued by a commenter.

Outline. The paper is organized as follows: we review related work in Sect. 2. In Sect. 3, we overview attack steps. In Sect. 4, we describe our data set preparation. Section 5 presents our selected features. Section 6 shows experimental results, and we conclude in Sect. 7.

2 Related Work

Online behaviour is representative of many aspects of a user’s demographics [14, 18]. Automatic age prediction of social network users from their public data

has obtained great attention in the past decade since age is an important factor for recommendation systems. Many studies have applied language analysis on the text generated by the target user (users' messages, posts, and status updates) to estimate the user age by implementing machine learning approaches [12, 17]. It has been shown that easily accessible digital records of behavior (such as likes) allow one to accurately predict sensitive personal attributes including age [11]. Both content and stylistic features (such as part-of-speech and the amount of slang words) have been found valuable for predicting the age of users [13]. The effect of age on writing style has also been addressed. Authors of [15] have found that when people get older, they tend to write more positive and fewer negative words, focus more on future and less on the past and make fewer self-references. Unlike the above works, we consider the non-user generated data as a potentially sensitive information that can be used to infer target user attributes. These data are easily available as there is no need to explore the target user vicinity network. Indeed, they reduce the complexity of inference attacks to the point that they can be launched online. Moreover, we handle emoji and *GIF* which makes the attack possible if the picture received different types of responses (Fig. 1).

Picture analysis has been applied in attributes inference attack. [21] propose gender identification through user shared images in Fotolog, Flickr, and Pinterest, three image-oriented social networks. Even though they received good results, they perform image processing and computer vision techniques and analyze the content of individual images (in offline mode), which is not feasible in an online attack. Besides, [2, 16] represent gender inference attacks by using picture metadata. They consider the different reactions of other Facebook users while commenting on female and male-owned pictures. Our work is related to their work. However, our objective is to find the effect of owner age in a positive and negative reduction of emotional responses.

3 Steps Overview

We sketch attack steps, and the learning regression model that we use. Figure 2 shows the system components used to identify the owner age. *Data collection* allows one to create a ground truth to train the classifier. *Data Pre-processing* cleans comments by reformatting inflected words, abbreviations, intentional/unintentional misspelling words, flooded characters, and by removing stop words. This step aims to neat the collected raw data and get the data ready to be analyzed. In *Feature Selection and Extraction* step, we first select a handful number of features from the pruned data and next extract the best features, that contribute the most to the final result. This step intends to (i) reduce the classifier training complexity, and (ii) improve the performance of the machine learning classifier. In *Machine Learning* step, we train logistic regression with, and without best features. Finally, we apply the classifier to identify target user age. For age group, [3] shows that decades play an outsized role in human psychology. Therefore the following classes are considered: *18 to 20*, *20 to 30*, *30 to 40*, and *40 to 50*, as these categories comprise the majority of users [5] (Fig. 2). We explain the components in detail from Sect. 4.

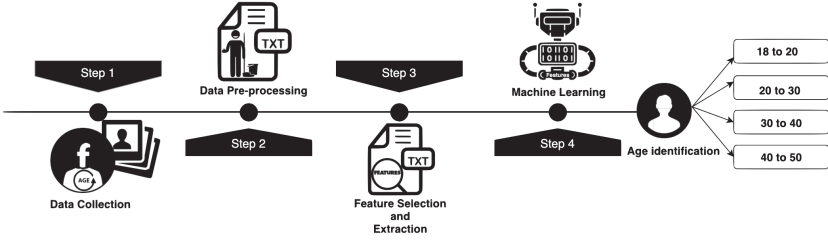


Fig. 2. Machine learning process in our attack

The output variable y in a regression problem is constructed as a linear combination of input variables $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$, transformed by the logistic function. We intend to infer the owner age from a set of pictures PIC . Each published picture $pic \in PIC$ contains metadata (a set of comments c_{pic} , and generated alt-text a_{pic}). We employ logistic regression algorithm [6] for our task. For classification in two classes ($y \in \{0, 1\}$) the model estimates a conditional distribution

$$P(y|\mathbf{x}, \theta) = \frac{1}{1 + exp^{-y(\theta_0 + \mathbf{x}^T \theta)}} \tag{1}$$

where θ_0 and θ are the parameters to be learned. We operate a *one-vs-all* method to handle multi-class classification. We turn the problem of age inference into 4 separate binary classification problems (where we predict only $y \in \{0, 1\}$), as we have 4 age groups. Then, logistic regression assigns value of 1 to one class (for example, 18 to 20) and 0 to the other classes (20 to 30, 30 to 40, and 40 to 50). As a result, we train a logistic regression classifier for each class (age group) i , where $i \in \{1, 2, 3, 4\}$ as follows:

$$h_{\theta}^i(\mathbf{x}) = P(y = i|\mathbf{x}; \theta) \tag{2}$$

that estimates the probability of $y = i$, given \mathbf{x} , and parameterised by θ . Having calculated the result for $h_{\theta}^i(\mathbf{x})$, the class that gives the maximum number of the score will be selected as follows:

$$\arg \max_i (h_{\theta}^i(\mathbf{x})) \tag{3}$$

4 Data Collection and Pre-processing

We present here the dataset and the pre-processing steps.

4.1 User Picture Metadata

To launch an age inference attack, we have to collect the ground truth data for the training process. Hence, for every single picture, we collect picture metadata, alt-text and comments from the HTML part of the picture. Let U be the set of all users, we can associate with each user $u \in U$ a triple (age_u, A_u, C_u) where age_u is the age group of u and A_u (resp. C_u) the alt text (resp. comments from commenters) received by the pictures that u has shared.

4.2 Data Pre-processing

We introduce our two pre-preprocessing steps as follows:

Picture Distribution. We consider the picture publishing time to avoid assigning a picture to a wrong age group. We can compute a user age by checking the published birth year which is available in user profile, and assign the user to an age group. On the other hand, each posted picture has a publication year. As mentioned in Sect. 1, *Alice* is 31 when our crawler encounters her profile (in 2020) and she has published two pictures when she was 28 and 31 respectively. We might wrongly assign *Alice* (with her both pictures) to the age group of 30 to 40 if we ignore the picture sharing date. She might have a different picture sharing style and other users might react differently to those two pictures (as they have been posted at two different ages). These differences can affect the analysis process. To solve the problem, we crawl each user picture and generate different users if the posted pictures belong to different age groups. As for the above example, we generate two *Alice* profiles postfixed by age groups (*Alice2030*, and *Alice3040*). This step aims to approximate user distribution in each category. The teenager category receives half of the new users, with 19 the most represented age. On the other hand, the second category (the most dominant one) contains most of the incorrectly categorized pictures. Lastly, the fourth category receives less than 20% of newly generated users. The dataset has unbalanced users even after applying this step. To make the dataset representative, we train the model on a balanced users distribution (see Sect. 6).

Cleansing Comments. We clean the extracted comments as follows:

Applying Lemmatization. This is a text normalization technique from the field of NLP that diminish the morphological variations of words to their root form, called *lemma*. For example, *loves*, *loving*, and *lovely* are all forms of the word *love*, which is the lemma of all these words.

Changing Shape of Words. Abbreviations and words with flooded characters (e.g. Heeeelloooo) in posted comments make the comments short and emphasize the emotion behind the words, respectively. We apply *NLTK*¹, a natural language processing package in Python to handle these cases. However, some abbreviations cannot be handled by *NLTK* package. For example *love u* is an abbreviated form of *love you*, where *u* can be interpreted as a misspelled letter *a* by *NLTK*. Therefore we change these cases to their corresponding proper format.

Eliminating stop words. Stop words are common words (e.g., *the*, *a*, *an*, *in*) in a language. As users write in an unstructured way, they tend to utilize more stop words in their comments. Hence, we remove stop words from the extracted comments.

5 Feature Selection and Extraction

We are interested in the following hypothesis: does picture owner age reduces the commenter emotional responses? Is it possible to find a correlation between

¹ <https://www.nltk.org/>.

owner age and commenters emotional responses and reactions while facing those pictures? To answer these questions, we apply *spearman rank-order correlation coefficient* that is a statistical test computed as follows:

$$r_s = 1 - \frac{6 \sum d_i^2}{n^3 - n} \quad (-1 \leq r_s \leq 1) \tag{4}$$

where d_i is the difference between two ranked variables (x_i and y_i), and n is the age interval size (in our case 32 as the age interval is [18, 50]). As an example, x_i is the picture owner age $x_i = \{18, 19, \dots, 50\}$, and y_i is the number of times 🥰 emoji is used by commenter while commenting on x_i 's pictures.

If the coefficient is 0 then it means there is no correlation between picture owner age and emotional reaction used by commenters, while a coefficient close to +1 or -1 corresponds to a perfect increase or decrease in emotional responses. To be significant, the observed *spearman*² value must be greater than the critical value (two-side $r_s \ll 0.001$, degree of freedom is 32).

Despite its simplicity this technique has three advantages: (i) it is a non-parametric technique which is unaffected by the distribution of the population, (ii) it is insensitive to outliers as it operates on ranked data, and (iii) it can be applied to any sample size even a very small one. We apply *spearman rank-order correlation coefficient* to effectively measure (i) picture sharing style and age correlation, (ii) age effect on commenters emotional responses, and (iii) age effect on commenters' responses when they respond to the same picture style, content and tags. Table 1 shows correlation/relationship.

Table 1. Spearman rank of: (a) commenters responses, (b) owner picture sharing style, (c) commenters words/emojis usage, (d) correlation of alt-text, words, and emojis.

(a)		(b)	
Responses	spearman's rank correlation coefficient	Alt-text	spearman's rank correlation coefficient
Wow reaction 🤩	-0.806	closeup	-0.791
Received comments	-0.756	1 person	-0.750
Emoji usage in comments	-0.746	2 people	-0.740
Love reaction ❤️	-0.686	smiling	-0.723
		2 people smiling	-0.720
		selfie	-0.706
		outdoor	-0.693
		ocean	-0.640
		child	0.625

(c)		(d)	
Comments	spearman's rank correlation coefficient	Comments	spearman's rank correlation coefficient
pretty	-0.795	selfie ❤️	-0.829
👉	-0.791	1 person 🥰	-0.819
❤️	-0.789	selfie 🥰	-0.817
🥰	-0.786	closeup 🥰	-0.792
🥰	-0.780	selfie 💖	-0.794
🥰	-0.773	1person 💖	-0.780
gorgeous	-0.729	1 person cute	-0.780
love	-0.712	1 person beautiful	-0.770
nice	-0.678	selfie beautiful	-0.779
beautiful	-0.636	1 person 🥰	-0.769

² <https://www.york.ac.uk/depts/maths/tables/spearman.pdf>.

As illustrated in Table 1(a), younger owners received (i) more emoji-based comments, (ii) more comments in general, and (iii) more emotional (❤️), or amazement (😱) reactions from commenters for their posted pictures. Commenters' enthusiasm expressed by these responses extinguished for older users' pictures as their coefficient values are close to -1 . Table 1(b) shows that younger owners share pictures in *selfie*, *closeup*, and *1 person* style, while these tags get less popular as they get older. As also shown, family related tag (e.g., *child*) is more often generated for older users as the *spearman coefficient* value of this tag is positive (0.625). Table 1(c) illustrates the effect of owner age on the hesitation of commenters in using emotional words and emojis while commenting on older owner pictures. Younger owners are likely to receive more emotional words and emojis (e.g., pretty, and ❤️) than older owners. Table 1(d) shows the commenters responses when they comment younger and older owners pictures with the same generated alt-text. For example, commenters use more ❤️ emoji when commenting younger users selfie pictures, than older users selfie pictures.

As a result, we show that commenters' emotional responses to pictures is strongly correlated (decreasing) with owner age. Note that the *spearman rank-order correlation coefficient* value can be applied to reduce the number of selected features by considering those features which are above the significant level. Feature selection in machine learning, is the process of selecting a subset of relevant features for the model construction. Based on Table 1 results, we select features in three different ways as follows:

By Patterns. Similarly to [16], we build our patterns based features for each age group separately. Basically, we consider patterns as types of structure used by commenters when commenting on a picture. As an example of pattern, we check how many times commenters utilize repeated emojis in commenting for each age group. For example, Fig. 1(a) forth and sixth comments from above.

By Linguistic Feature Selection. As illustrated in Tables 1(b) and (c), commenters react with less emotion to older owners pictures. We compute *n-grams* to capture word, emoji and phrase occurrences in posted comments and tags in alt-text in a window of size $n \in 1, 2, 3$.

By Correlational Analysis. Table 1(d) confirms the significant reduction of commenters words and emojis usage for younger and older owner, while they encounter the same generated alt-text. These differences lead us to consider pairs of alt-text and comments (containing emoji, word, *GIF*) as features. We constructed a co-occurrence network to explore these relationship.

By applying these feature selection techniques, we collect 198 features. Note that we keep those features that have significant negative or positive correlation with age according to their *spearman's rank correlation coefficient* values ($r_s \ll 0.001$). After selecting features, we have to prune them to improve the performance and speed of the classifier. As for feature extraction, we used *LIME* [19], a technique that explains a classification by ranking features according to their contribution to predict the final result. Figure 3 shows the output of *LIME* for users in age group 30 to 40. We keep the features that have a positive contribution to the final result. In that way we reduce the number of extracted features

to 115. *LIME* attempts to understand the contribution of each feature to the prediction result. It provides local interpretability and allows us to determine which feature has the most impact on the prediction. It can be used as a first step to prune useless features.

Contribution*	Feature	Value
+1.111	<BIAS>	1.000
+0.248	ulook	1.000
+0.213	u0001f602u0001f602	1.000
+0.169	youre	1.000
+0.161	loveit	0.000
+0.153	u0001f495	1.000
+0.073	sopretly	0.000
+0.070	lovetha	0.000
+0.069	u2764	0.000
+0.057	hahaha	1.000
+0.048	goodpicture	0.000
+0.045	greatpicture	0.000
+0.026	sohappy	0.000
+0.024	u0001f923u0001f923	0.000
+0.019	youhave	1.000
+0.018	socute	0.000

Fig. 3. LIME output

Although *LIME* gives the contribution score of each feature to the final result, we want to be sure that the selected features are the most significant ones. To that end we use *Univariate feature selection* on top of the pruned features. It selects features that have a statistically significant relationship to the final result. Finally, we end with 89 features, called *best feature set*.

6 Experimental Results

We evaluate our classifier performance with and without the best feature set on a dataset that comprises 8,922 random collected pictures, where 6184 pictures received complete metadata (commenters responses and alt-text) and where 1762 pictures received no comments. Moreover, Facebook system was unable to generate alt-text for 976 pictures. Below, we represent our age inference attack based on picture metadata availability (As mentioned in Sect. 1). We implement the logistic regression classifier by using Python library *scikit-learn*. We first divide the dataset into two part, assigning 80% of it to train the classifier and 20% to testing. To make sure if the dataset is representative, we select close distribution of each category for training dataset. We apply 5-fold cross-validation on our training dataset to see the performance of classifier on trained dataset. Finally, we evaluate our classifier on the test dataset (unseen dataset to classifier) to prevent bias. In our experiment, we apply *AUC-ROC*³ curve for evaluating the learning model. *ROC* is a probability curve and created by plotting the *True Positive Rate (TPR)* against the *False Positive Rate (FPR)* at various threshold settings. The *TPR* is the probability of detection and *FPR* known as the probability of false alarm. *AUC* measures the performance of classifier across all possible classification thresholds. We represent the *AUC-ROC* result of logistic

³ https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html.

regression in different age categories in Fig. 4. We labeled these age categories from 0 to 3, where 0 represent 18 to 20, 1 shows 20 to 30, 2 is labelled for 30 to 40, and 3 is for 40 to 50. To evaluate the classifier performance, we train the logistic regression on two different input features. First, we train logistic regression on the best feature set that we extracted from *Univariate feature selection* algorithm. Second, we remove best feature set and train the classifier without these features. Figure 4 shows the result of these two input data with four curves, one curve for each class.

Figure 4(a) shows the result of logistic regression classifier in different age categories when Facebook generated alt-text is the only picture metadata (976 pictures). In this case, logistic regression infers the first class (18 to 20) better than the other three classes with 70% AUC. Figure 4(b) represents the result where commenters responses are the only picture metadata (1762 pictures).

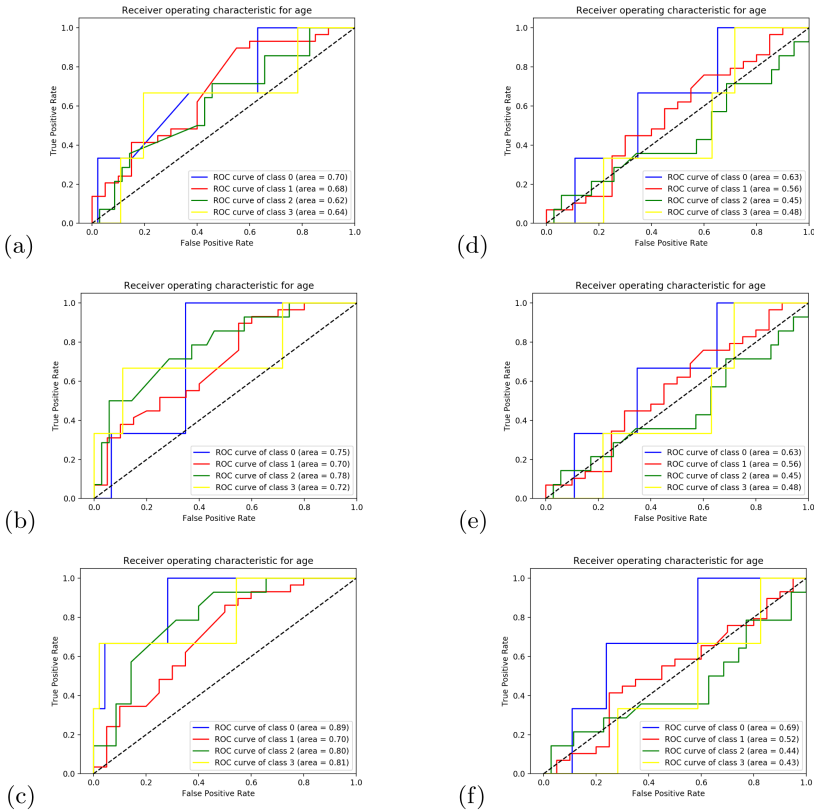


Fig. 4. AUC result of logistic regression trained on: (a) only alt-text (b) only commenters reactions (c) both alt-text and commenters reactions (d) removing alt-text features (e) removing commenters reactions features (f) removing alt-text and commenters reactions features.

Logistic regression infers third class (40 to 50) with 78% AUC. The 3 other classes are identified fairly similarly. Finally, Fig. 4(c) displays the result where both comments and alt-text are the available data to attacker (6184 pictures). As illustrated, class 0 (18 to 20) can be predicted very well with AUC of 89%. Following the previous result, the result of logistic regression after removing best feature set drops dramatically. Figure 4 (d,e, and f) proves that the age obfuscation is possible although it is out of the scope of this paper. To sum up, our preliminary experiments on Facebook show that age prediction is possible even when the user is careful and hide all profile attributes.

7 Conclusion

In this study, we have shown that commenters react differently to younger and older owner pictures. Such a disparity can be used to implement an age inference attack. The results show the possibility of age inference attack on Facebook users that have only published their pictures, by leveraging non-user generated data (picture metadata). This attack has practical advantages: (i) it exploits data that are easily available (crawling user neighborhood is needless), (ii) thanks to alt-text image processing is needless, (iii) words, emojis, and GIF handle all possible received comments (as presented in Fig. 1). These advantages make the attack suitable for online mode. As a future work, we plan to (i) automatically extract features by using a *Random Walk* approach, (ii), adjust the age groups to infer the owner exact age, and (iii) enrich the dataset to compare the result of deep learning approaches with our current algorithm.

References

1. Abdelberic, C., Ács, G., Kâafar, M.A.: You are what you like! information leakage through users' interests. In: 19th Annual Network and Distributed System Security Symposium, NDSS 2012, San Diego, California, USA, 5–8 February 2012. The Internet Society (2012)
2. Alipour, B., Imine, A., Rusinowitch, M.: Gender inference for Facebook picture owners. In: Gritzalis, S., Weippl, E.R., Katsikas, S.K., Anderst-Kotsis, G., Tjoa, A.M., Khalil, I. (eds.) TrustBus 2019. LNCS, vol. 11711, pp. 145–160. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-27813-7_10
3. Alter, A.L., Hershfield, H.E.: People search for meaning when they approach a new decade in chronological age. *Proc. Nat. Acad. Sci.* **111**(48), 17066–17070 (2014)
4. Belinic, T.: Personality profile of social media users how to get maximum from it. <https://medium.com/krakensystems-blog/personality-profile-of-social-media-users-how-to-get-maximum-from-it-5e8b803efb30> April 2009
5. Clement, J.: Distribution of facebook users worldwide as of January 2020, by age and gender, February 2020
6. Cox, D.R.: The regression analysis of binary sequences. *J. Royal Stat. Soc.: Ser. B (Methodological)* **20**(2), 215–232 (1958)
7. Dey, R., Tang, C., Ross, K.W., Saxena, N.: Estimating age privacy leakage in online social networks. In: Proceedings of the IEEE INFOCOM 2012, Orlando, FL, USA, 25–30 March 2012, pp. 2836–2840. IEEE (2012)

8. Farahbakhsh, R., Han, X., Cuevas, A., Crespi, N.: Analysis of publicly disclosed information in facebook profiles. In: *Advances in Social Networks Analysis and Mining 2013*, ASONAM 2013, Niagara, ON, Canada - 25–29 August 2013, pp. 699–705. ACM (2013)
9. Gong, N.Z., Bin L.: You are who you know and how you behave: Attribute inference attacks via users' social friends and behaviors. *CoRR*, abs/1606.05893:979–995 (2016)
10. Kellogg, K.: The 7 biggest social media sites in 2020. <https://www.searchenginejournal.com/social-media/biggest-social-media-sites/> February 2020
11. Kosinski, M., Stillwell, D., Graepel, T.: Private traits and attributes are predictable from digital records of human behavior. *Proc. Nat. Acad. Sci.* **110**(15), 5802–5805 (2013)
12. Nguyen, D., Gravel, R., Trieschnigg, D., Meder, T.: How old do you think I am? a study of language and age in twitter. In: *Proceedings of the Seventh International Conference on Weblogs and Social Media, ICWSM 2013*, Cambridge, Massachusetts, USA, 8–11 July 2013. The AAAI Press (2013)
13. Nguyen, D., Smith, N.A., Rosé, C.P.: Author age prediction from text using linear regression. In *Proceedings of the 5th ACL Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities, LaTeCH@ACL 2011*, 24 June, 2011, Portland, Oregon, USA, pp. 115–123. The Association for Computer Linguistics (2011)
14. Pennacchiotti, M., Popescu, A.-M.: A machine learning approach to twitter user classification. In: *Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, 17–21 July 2011*. The AAAI Press (2011)
15. Pennebaker, J.W., Stone, L.D.: Words of wisdom: language use over the life span. *J. Personality Soc. Psychol.* **85**(2), 291 (2003)
16. Pijani, B.A., Imine, A., Rusinowitch, M.: You are what emojis say about your pictures: language-independent gender inference attack on facebook. In *SAC 2020: The 35th ACM/SIGAPP Symposium on Applied Computing*, online event, [Brno, Czech Republic], March 30 - April 3, 2020, pp. 1826–1834. ACM (2020)
17. Rangel, F., Rosso, P.: Use of language and author profiling: identification of gender and age. *Nat. Lang. Process. Cognitive Sci.* **1**, 177 (2013)
18. Rao, D., Yarowsky, D., Shreevats, A., Gupta, M.: Classifying latent user attributes in twitter. In: *Proceedings of the 2nd International Workshop on Search and Mining User-generated Contents, SMUC@CIKM 2010*, Toronto, ON, Canada, October 30, 2010, pp. 37–44. ACM (2010)
19. Ribeiro, M.T., Singh, S., Guestrin, C.: Why should I trust you?: explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 13–17 August 2016, pp. 1135–1144. ACM (2016)
20. Sung, Y., Lee, J.-A., Kim, E., Choi, S.M.: Understanding motivations for posting pictures of oneself: why we post selfies. *Personal. Individual Diff.* **97**, 260–265 (2016)
21. You, Q., Bhatia, S., Luo, J.: A picture tells a thousand words - about you! user interest profiling from user generated visual content. *Signal Process.* **124**, 45–53 (2016)



Enacting Data Science Pipelines for Exploring Graphs: From Libraries to Studios

Genoveva Vargas-Solar¹(✉), José-Luis Zechinelli-Martini²,
and Javier A. Espinosa-Oviedo³

¹ Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG-LAFMIA,
38000 Grenoble, France
genoveva.vargas@imag.fr

² Fundación Universidad de las Américas Puebla, 72810 San Andrés Cholula, Mexico
joseluis.zechinelli@udlap.mx

³ Univ. of Lyon, LAFMIA, 69008 Lyon, France
javier.espinosa@imag.fr

Abstract. This paper proposes a study of existing environments used to enact data science pipelines applied to graphs. Data science pipelines are a new form of queries combining classic graph operations with artificial intelligence graph analytics operations. A pipeline defines a data flow consisting of tasks for querying, exploring and analysing graphs. Different environments and systems can be used for enacting pipelines. They range from graph NoSQL stores, programming languages extended with libraries providing graph processing and analytics functions, to full machine learning and artificial intelligence studios. The paper describes these environments and the design principles that they promote for enacting data science pipelines intended to query, process and explore data collections and particularly graphs.

Keywords: Data science · Data flow · Graph analytics · Machine learning studio

1 Introduction

Huge collections of heterogeneous data containing observations of phenomena have become the backbone of scientific, analytic and forecasting processes. Combining artificial vision and machine learning with data science techniques, it is possible to compute mathematical models to understand and predict phenomena. To achieve this, data must go through complex and repetitive processing and analysis pipelines, namely “data science pipelines”. Observations can be

Partially supported by the Auvergne-Rhône-Alpes Pack-Ambition project SUMMIT (<http://summit.imag.fr>). The work is part of the activities of the working group DOING-MADICS <https://www.madics.fr/ateliers/doing>.

structured as networks that have their own interconnection rules determined by the variables (i.e., attributes) characterising each observation. Yet, relations among observations and interconnection rules are most of the times not explicit and it is the role of analytics process to deduce, discover and eventually predict them.

Four challenges must be considered for performing data science pipelines on graphs:

1. How to reuse and integrate existing solutions (libraries, environments, services) into complex recurrent pipelines that can explore and process (big) graphs at scale?
2. How to compare and analyse results of graph analytics processes by keeping track of the tasks that lead to results?
3. How to deploy graph processing pipelines on target architectures like the cloud and HPC environments?
4. How to maintain pipelines' adjusting parameters so that they can lead to new data, better results quality with better performance?

Different environments and systems can be used for designing and enacting pipelines. They range from graph NoSQL stores and programming languages extended with libraries providing graph processing and analytics functions, to full machine learning and artificial intelligence studios. This paper proposes a study of existing environments used to enact data science pipelines that can be applied to graphs. The paper describes these environments and their design principles for enacting pipelines intended to query, process and explore data, particularly graphs.

The remainder of the paper is organised as follows. Section 2 gives synthesises approaches and systems addressing graph modelling, processing, querying and exploration. Section 3 describes data labs and machine learning/artificial intelligence studios intended to provide full-fledged environments for defining and enacting data science pipelines. Section 4 concludes the paper and discusses research perspectives.

2 Processing, Querying and Exploring Graphs

Different data modelling approaches have used graphs to model data or represent knowledge [5–7, 13, 18, 21, 22]. Ontologies have been used in databases and Semantic Web. They have been used for solving heterogeneous data integration problems, modelling complex interconnected data in the Web and social networks [12, 23], and data-centric information systems integration [19]. Linked data [4] and knowledge graphs [20] have been used to propose abstract and schematic representations of knowledge domains (art, history, biology classifications, retail). Knowledge associated with concrete content can be then explored interactively or through queries expressing navigational operations or subgraph extraction or clustering [15].

A key aspect is to provide well-adapted storage solutions for maintaining, managing and exploring persistent graphs. There are two families of systems: (i) those adopting a database strategy with graph stores and associated query evaluation engines based on declarative or semi-declarative languages, and (ii) those centred on libraries with algorithms implementing several graph operations that are then used within programs intended to process graphs. The next sections synthesise these systems' families.

2.1 Graph Stores

A graph database is used to deal with dynamic and complex relationships in connected data. To handle graph databases, academic and industrial works have introduced the concept of graph and graph storage, where graphs model huge data sets. In every graph, there are nodes, properties, and edges as the relationship among nodes. The graph database exhibits a single data structure known as a graph (similar to a database schema). Graph database systems use square or adjacency matrices, and index free square matrix for representing finite graphs. Then each node manages the direct relationship with the adjacent nodes. These data structures are used both to represent the graph and ensure high-performance graph traversal operations for solving queries.

Academic and industrial systems provide solutions for processing and analysing graphs [5–7, 13, 18, 21, 22]. Such solutions stem from the database domain including graph databases that emerged in the early 2000 and later on [11, 17], and then during the NoSQL boom where the most prominent graph stores are GraphDB lite¹, Neo4J², OrientDB³, Graph Engine⁴, HyperGraphDB, MapGraph⁵, ArangoDB⁶, Titan⁷, BrightStarDB⁸, Cayley⁹, WhiteDB¹⁰, Orly¹¹ and CosmosDB¹². In general these graph stores provide functions for creating graphs similar to data definition languages in classic relational systems. They also provide built-in functions for manipulating graphs in more or less declarative query expressions.

For developing a graph data science pipeline, it is necessary to embed queries within programs. For example, discover the social relations and communities of people within a social network or compute the most influential people represented in a social network graph. The pipeline is implicit in the program control flow

¹ GraphDB Lite, <https://www.ontotext.com/products/graphdb/graphdb-free/>.

² Neo4J, <https://neo4j.com>.

³ OrientDB, <https://orientdb.com/why-orientdb/>.

⁴ GraphEngine, <https://www.graphengine.io>.

⁵ Mapgraph, <http://mapgraph.io>.

⁶ ArangoDB, <https://www.arangodb.com>.

⁷ Titan, <http://titan.thinkaurelius.com>.

⁸ BrightStarDB, <http://brightstardb.com>.

⁹ CayLayGraph, <https://github.com/cayleygraph/cayley>.

¹⁰ WhiteDB, <http://whitedb.org>.

¹¹ Orly, <https://github.com/orlyatomics/orly>.

¹² <https://docs.microsoft.com/en-us/azure/cosmos-db/introduction>.

that coordinates the order in which the operations are executed. For example, a pipeline for answering these queries would include tasks like: creating a graph from a social network, profiling it through the number of nodes and edges, and applying a machine learning algorithm to determine possible connections among people for the first query, and applying a page rank operation for the second one.

2.2 Graph Analytics

Deep learning models on graphs have achieved remarkable performance in various graph analysis tasks (e.g., node classification, link prediction and graph clustering). Works have shown that graph querying can be also based on data mining techniques as an alternative to the definition of operators as in graph stores. This technique is known as mining-based graph indexing [9] that runs a graph mining algorithm on graph database, and then indexing patterns are implemented on these patterns after mining.

Mining algorithms are based on frequent graphs which are considered the base indexing unit. The objective is then to find frequent subgraphs within a graph database [24] and consider them as possible candidate answers to a given query. Other methods consider frequent trees as indexing units [25]. The objective here is to find frequent trees in a graph database and selecting some of them to build an index and evaluate certain properties.

Graph analytics by definition considers the whole graph (i.e., not necessarily every property but all vertices and edges). Example queries include resolving global pattern matching, shortest paths, max-flow or min-cut, minimum spanning trees, diameter, eccentricity, connected components, PageRank and some traversals (e.g., finding all shortest paths of unrestricted length). Graph processing systems such as Pregel [14] or Giraph¹³ focus specifically on resolving these graph analytics tasks [10].

Network science, social networks analytics and data science rely also on the representation of complex systems as (multi-dimensional and dynamic) graphs [1,3]. In these domains given a raw dataset, the objective is to “discover” the graph that can best model the phenomena and their evolution and then operate on the graph to provide insight and foresight analysis [3,16].

Graphs and associated querying navigational and analytics operations are mathematically and computationally complex and costly. Complexity and cost become critical when the number of nodes and edges becomes massive and the deployment of processing, querying, exploration and analytics operations on graphs call for flexible and efficient platforms that can run these operations at scale.

Regarding graph processing at scale [7,18], solutions like Spark with GraphX, Graph processing with Python and programming languages like CUDA running on top of HPC or GPU farms [13], and mathematical analytical platforms like

¹³ <https://giraph.apache.org/>.

Matlab or R, have addressed these issues from different perspectives (e.g., interactive graph visual exploration). The majority of solutions reason about static graphs, and even if they are able to analyse dynamic situations, analytics and processing are run in sequential batch processes [8]. Systems and environments allowing to process graphs must provide strategies to keep track of the querying and analytics operations, so that they can be reused and also explored for understanding the way results were computed (i.e., reproducibility).

3 Data Science Environments

Two families of environments have emerged to provide tools for data scientists to explore, engineer and analyse data collections that can be applied to graphs: (i) data labs externalised on the cloud that combine notebook environments with data libraries for defining and executing notebooks, and (ii) studios that combine libraries, data labs and large scale execution environments based on the notion of project or experiment.

Data-oriented problems are solved on top of programming environments based on a programming language. Python is one of the most popular programming languages that can be seen as a multiparadigm language adapted for exploring data. Python is extended with a full ecosystem of libraries prepared for performing data processing and analytics tasks. It provides ad hoc libraries for dealing with graphs (`networkx`¹⁴) and facilitate the definition of programs that can be used for exploring them (e.g., `shortest_path()`, `shortest_path_length()`, `all_pairs_dijkstra_path()` provided by `networkx`).

Programming environments can be delivered by integrated development environments that can be installed locally or externalised in clouds. This environments are based on the notion of notebook that is a JSON document, following a versioned schema, and containing an ordered list of input/output cells which can contain code, text (using Markdown¹⁵), mathematics, plots and rich media. Thereby, experiments can become completely and absolutely replicable.

Data Labs with Runtime Environments allow data scientists to find and publish data sets, explore and build models in a web-based data-science environment, and work with other data scientists and machine learning engineers. Data labs offer a public data platform, a cloud-based workbench for data science, and Artificial Intelligence experiments. Examples of existing data labs are Kaggle¹⁶ and CoLab¹⁷ from Google, and Azure Notebooks from Azure¹⁸.

The principle of data labs is first provide tools for managing data collections and automatically generating associated qualitative and quantitative descriptions. These descriptions provide insight to data scientists about the size of the

¹⁴ <https://networkx.github.io>.

¹⁵ <https://guides.github.com/features/mastering-markdown/>.

¹⁶ <http://www.kaggle.com>.

¹⁷ <https://colab.research.google.com>.

¹⁸ <https://notebooks.azure.com>.

data collections, licences, provenance as well as data structure and content distributions that can be visually observed. Associated to search engine facilities data collections can be shared to be used as input to target data science projects. Data labs offer storage space often provided by a cloud vendor (e.g., users of CoLab use their google drive storage space for data collections, notebooks and results). Other open and private spaces can be coupled and used as persistence support for decoupling the enactment environment from the data persistence support (e.g., data can be initially stored in github and then uploaded to the data lab for analytics purposes). In both cases available space depends on the type of subscription to the data lab.

3.1 Platforms for Custom Modelling

Platforms for custom modelling provide a suite of machine learning products that allows developers with little experience in this field to train high quality models that meet the specific needs of their companies. They are provided as services by commercial cloud providers that include storage, computing support and particularly environments for training and enacting greedy artificial intelligence models. The main vendors providing this kind of platforms are Amazon Sage Maker¹⁹, Azure ML Services²⁰, Google ML Engine²¹ and IBM Watson ML Studio²². All except Azure ML Service provide built-in machine learning and artificial intelligence algorithms, prediction tools like linear regression, and operation for processing data structures such as tabular data representations. These tools are used for implementing data science pipelines based on graphs.

Table 1 gives a comparative view of the main vendors providing this kind of platforms considering whether they provide built-in algorithms and the associated frameworks or libraries that can be used for running the AI models.

Table 1. Platforms for custom modelling

	Amazon Sage Maker	Azure ML Services	Google ML Engine	IBM Watson ML Studio
Built-in algorithms	Y	N	Y	Y
Supported frameworks	TensorFlow, ML.Net, Keras, Gluon, Pytorch, Keras, Caffe2, Chainer Torch	Tensorflow, Scikit-Learn, MS Cognitive , Toolkit, SparkML	Tensorflow, Scikit-Learn, XGBoost, Keras	Tensorflow, SparkML lib, Scikit-Learn, XGBoost, PyTorch, IBM SPSS, PMML

¹⁹ <https://aws.amazon.com/fr/sagemaker/>.

²⁰ <https://azure.microsoft.com/en-us/services/machine-learning/>.

²¹ <https://cloud.google.com/ai-platform>.

²² <https://www.ibm.com/cloud/machine-learning>.

They also support the most commonly used machine learning and artificial intelligence libraries and frameworks for executing tasks that can be wrapped as pipelines. Supported libraries include: TensorFlow²³, ML.Net [2], Keras²⁴, Scikit-learn²⁵, ML lib Spark's machine learning library, Gluon²⁶, PyTorch²⁷, Microsoft Cognitive Services²⁸, XGBoost²⁹ and Statistical Package for the Social Sciences (SPSS). The data scientist can then choose and even combine different libraries and frameworks for developing a pipeline. These platforms can also support execution environments like Caffe2 (Convolutional Architecture for Fast Feature Embedding)³⁰ and Chainer³¹.

3.2 Data Science Studios

Machine Learning and Artificial Intelligence Studios give an interactive, visual workspace to easily build, test, and iterate on analytics models and develop experiments. An experiment consists of data sets that provide data to analytical modules, connected together to construct an analysis model. An experiment has at least associated one data set and one module. Data sets may be connected only to modules and modules may be connected to either data sets or other modules. All input ports for modules must have some connection to the data flow. All required parameters for each module must be set. An experiment can be created from scratch or derived from an existing sample experiment as a template.

Data sets and analysis modules can be drag-and-dropped onto an interactive canvas, connecting them together to form an experiment, which can be executed on a machine learning runtime (cloud) environment. Machine learning runtime environments provide the tools needed for executing machine learning workflows including data stores, interpreters and runtime services like Spark, Tensorflow and Caffe for executing analytics operations and models.

The most prominent studios are provided by major cloud and Big Data processing vendors. For example Amazon Machine Learning, Microsoft Artificial Intelligence and Machine Learning Studio, Cloud Auto ML, Data Bricks ML Flow and IBM Watson ML Builder. Each provide different families of built-in models such as classification, regression, clustering, anomaly detection, recommendation and ranking.

²³ <https://www.tensorflow.org>.

²⁴ <https://keras.io> capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, R, Theano, or PlaidML.

²⁵ Scikit-learn is a machine learning library for Python.

²⁶ Gluon is a deep learning interface by AWS and Microsoft.

²⁷ <https://pytorch.org>.

²⁸ MS cognitive services is a set of machine learning algorithms.

²⁹ <https://github.com/dmlc/xgboost>.

³⁰ Caffe2 includes new features such as Recurrent Neural Networks and it is merged into PyTorch, see Caffe2 Merges With PyTorch". Medium. May 16, 2018, <https://medium.com/@Synced/caffe2-merges-with-pytorch-a89c70ad9eb7>.

³¹ <https://chainer.org>.

3.3 Discussion

Current data science solutions provide analytics and artificial intelligence libraries. They focus on libraries of functions that implement almost all known models proposed in data mining, probability, statistics, machine learning and deep learning. The functions implement also assessment measurement and they can be combined into code, often in notebooks, that implicitly defines data science pipelines. Yet they externalise data management and create a divide since they do not provide well adapted solutions for dealing with: data loading, in memory/cache/disk indexing, data persistence, query optimisation, concurrent access, consistency and access control. These functions must be revisited under less strong hypothesis to support the enactment of data science pipelines.

Deep learning frameworks such as Caffe2, Cognitive Toolkit, TensorFlow, and Apache MXNet³² are, in part, an answer to the question “how can we speed this process up?”. Just like query optimisers in databases, the more a training engine knows about the network and the algorithm, the more optimisations it can make to the training process. For example, it can infer what needs to be re-computed on the graph based on what else has changed, and skip the unaffected weights to speed things up. These frameworks also provide parallelisation to distribute the computation process, and reduce the overall training time.

However, to achieve these optimisations, most frameworks require the developer to do some extra work. Specifically, by providing a formal definition of the network graph, up-front, and then ‘freezing’ the graph and just adjusting the weights.

The network definition, which can be large and complex with millions of connections, usually has to be constructed by hand. Not only are deep learning networks unwieldy, but they can be difficult to debug and it is hard to re-use the code between projects.

Studios provide all-in-all tools to deal with the complete data science pipelines life cycle including design, testing, training, assessment, export and monitoring to decide when it is necessary to re-train models. They provide necessary data fetching, transformation, engineering and preparation, modelling, visualisation, training, assessment and deployment tools. The tools are adapted for non-structured and structured data sets, including graphs. The intelligence of a data scientist is used to best combine the tools to develop specific solutions. These studios act also as social networks where know-how and data are shared publicly among data scientists. Thus it is possible to reuse existing data science pipelines and modify them to address other similar problems. Existing solutions are searched through a search engine with keyword queries.

4 Conclusion and Future Work

For the time being graph analytics (processing, querying and exploration) through pipelines is done in an artisanal manner according to the type of target experiment. This practice does not encourage factorising and reusing the

³² <https://mxnet.apache.org>.

pipelines that are implemented for addressing a problem. Having pipelines as first class citizens can enable to manipulate them, and particularly optimizing them by applying well-known classic query optimisation techniques.

Coupling together data analytics methods and models with data management strategies and execution environments services for addressing data science query processing, is an important challenge in the database community. The next data science querying environments providing services at scale are still to come, and there is room to contribute and propose leading solutions to the opportunities opened by this new data vision. Given the challenges introduced by emerging data centric sciences and advances on data science stacks and environments, it will be possible to integrate their forces to encourage the design, maintenance and reuse of data science pipelines. This concerns our current and future work.

References




1. Abdelhamid, E., Canim, M., Sadoghi, M., Bhattacharjee, B., Chang, Y.C., Kalnis, P.: Incremental frequent subgraph mining on large evolving graphs. *IEEE Trans. Knowl. Data Eng.* **29**(12), 2710–2723 (2017)
2. Ahmed, Z., et al.: Machine learning at microsoft with ml. net. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2448–2458 (2019)
3. Barabási, A.L., et al.: *Network Science*. Cambridge University Press, Cambridge (2016)
4. Bizer, C., Heath, T., Berners-Lee, T.: Linked data: the story so far. In: *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, pp. 205–227. IGI Global (2011)
5. Bonifati, A., Fletcher, G., Voigt, H., Yakovets, N.: Querying graphs. *Synthesis Lect. Data Manage.* **10**(3), 1–184 (2018)
6. Bonifati, A., Holubová, I., Prat-Pérez, A., Sakr, S.: Graph generators: state of the art and open challenges. *arXiv preprint arXiv:2001.07906* (2020)
7. Dayarathna, M., Suzumura, T.: Towards scalable distributed graph database engine for hybrid clouds. In: *2014 5th International Workshop on Data-Intensive Computing in the Clouds*, pp. 1–8 (2014)
8. Desikan, P., Srivastava, J.: Mining temporally evolving graphs. In: *Proceedings of the the Sixth WEBKDD Workshop in Conjunction with the 10th ACM SIGKDD Conference*, vol. 22. Citeseer (2004)
9. Dinari, H.: A survey on graph queries processing: techniques and methods. *Int. J. Comput. Netw. Inf. Secur.* **9**(4), 48 (2017)
10. Han, M., Daudjee, K., Ammar, K., Özsu, M.T., Wang, X., Jin, T.: An experimental comparison of pregel-like graph processing systems. *Proc. VLDB Endowment* **7**(12), 1047–1058 (2014)
11. He, H., Singh, A.K.: Graphs-at-a-time: query language and access methods for graph databases. In: Wang, J.T. (ed.) *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10–12, 2008*, pp. 405–418. ACM (2008). <https://doi.org/10.1145/1376616.1376660>

12. Heath, T., Bizer, C.: *Linked Data: Evolving the Web into a Global Data Space, Synthesis Lectures on the Semantic Web: Theory and Technology*, vol. 1. Morgan & Claypool, 1st ed., html version edn, February 2011. <https://doi.org/10.2200/S00334ED1V01Y201102WBE001>
13. Kalmegh, P., Navathe, S.B.: Graph database design challenges using HPC platforms. In: *2012 SC Companion: High Performance Computing, Networking Storage and Analysis*, pp. 1306–1309 (2012)
14. Malewicz, G., et al.: Pregel: a system for large-scale graph processing. In: *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pp. 135–146 (2010)
15. Mattson, T.G., et al.: Standards for graph algorithm primitives. In: *2013 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–2 (2013)
16. Mayer, R., Mayer, C., Tariq, M.A., Rothermel, K.: Graphcep: real-time data analytics using parallel complex event and graph processing. In: *Proceedings of the 10th ACM International Conference on Distributed and Event-based Systems*, pp. 309–316 (2016)
17. Muntés-Mulero, V., Martínez-Bazán, N., Larriba-Pey, J.-L., Pacitti, E., Valduriez, P.: Graph partitioning strategies for efficient bfs in shared-nothing parallel systems. In: Shen, H.T., et al. (eds.) *WAIM 2010. LNCS*, vol. 6185, pp. 13–24. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-16720-1_2
18. Patil, N.S., Kiran, P., Kiran, N.P., Patel, K.N.: *A survey on graph database management techniques for huge unstructured data* (2018)
19. Paulheim, H.: *Ontology-based Application Integration*. Springer, New York (2011). <https://doi.org/10.1007/978-1-4614-1430-8>
20. Paulheim, H.: Knowledge graph refinement: a survey of approaches and evaluation methods. *Semantic Web* **8**(3), 489–508 (2017)
21. Rawat, D.S., Kashyap, N.K.: Graph database: a complete gdbms survey. *Int. J.* **3**, 217–226 (2017)
22. Robinson, I., Webber, J., Eifrem, E.: *Graph Databases*. O’Reilly, Beijing, 2 edn (2015). <https://www.safaribooksonline.com/library/view/graph-databases-2nd/9781491930885/>
23. Segaran, T., Evans, C., Taylor, J., Toby, S., Colin, E., Jamie, T.: *Programming the Semantic Web*. O’Reilly Media Inc., 1st edn. (2009)
24. Yan, X., Yu, P.S., Han, J.: Graph indexing: a frequent structure-based approach. In: *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, pp. 335–346 (2004)
25. Zhang, S., Hu, M., Yang, J.: Treepi: a novel graph indexing method. In: *2007 IEEE 23rd International Conference on Data Engineering*, pp. 966–975. IEEE (2007)

**International Symposium
on Data-Driven Process Discovery
and Analysis (SIMPDA 2020)**



Towards the Detection of Promising Processes by Analysing the Relational Data

Belén Ramos-Gutiérrez¹(✉) , Luisa Parody² ,
and María Teresa Gómez-López¹ 

¹ Universidad de Sevilla, Sevilla, Spain
{brgutierrez,maytegomez}@us.es

² Universidad Loyola Andalucía, Sevilla, Spain
mlparody@uloyola.es
<http://www.idea.us.es/>

Abstract. Business process discovery provides mechanisms to extract the general process behaviour from event observations. However, not always the logs are available and must be extracted from repositories, such as relational databases. Derived from the references that exist between the relational tables, several are the possible combinations of traces of events that can be extracted from a relational database. Different traces can be extracted depending on which attribute represents the *case_id*, what are the attributes that represent the execution of an activity, or how to obtain the timestamp to define the order of the events. This paper proposes a method to analyse a wide range of possible traces that could be extracted from a relational database, based on measuring the level of interest of extracting a trace log, later used for a discovery process. The analysis is done by means of a set of proposed metrics before the traces are generated and the process is discovered. This analysis helps to reduce the computational cost of process discovery. For a possible *case_id* every possible traces are analysed and measured. To validate our proposal, we have used a real relational database, where the detection of processes (most and least promising) are compared to rely on our proposal.

Keywords: Process discovery · Promising process · Measures · Relational databases

1 Introduction

Business Process Management (BPM) facilitates the modelling and deployment of the process with a high level of automation [15]. BPM is centred on the optimization of the processes, based on a described model and the observation of the real actions executed in the organizational daily activities. The irruption of

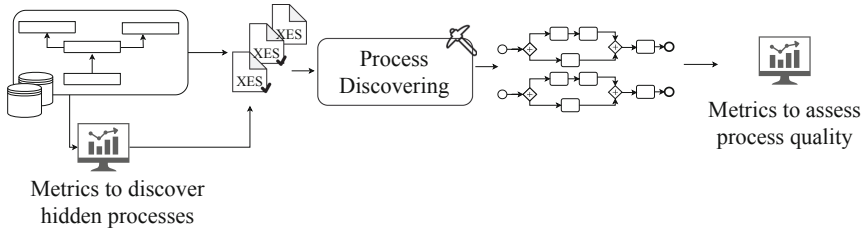


Fig. 1. Measuring the promising processes

BPM in industrial scenarios [20] has provoked to tackled cases where processes are not modelled or executed by Business Process Management Systems but implemented as specific applications for each organization. These applications generate an important quantity of data that is stored as evidence of the processes executed. This data could be used in Process Mining [5] to discover the processes and to ascertain if the organization is working as expected. However, one of the challenges to applying process mining techniques is to generate the event logs used as input of the discovering algorithms. An event log can be defined as an ordered list of activities executed in accordance with a *case_id* that differentiates each trace. There are several papers and tools that facilitate the extraction of the traces from relational databases for later discovery, as detailed in Related Work section (Sect. 2). The problem is that derived from the relationships that exist between the data of the relational tables, a huge number of different traces can be obtained from a relational database. The knowledge of the expert is a key aspect to ascertain which attributes are relevant to extract promising business model in a discovery process. This implies to ascertain which attributes will represent: the *case_id*, the *activity name* and the *timestamp*.

In order to guide the expert about the feasibility of detecting a promising process from a possible trace extracted from a relational database, we propose a method to measure the potential traces that can be extracted. As shown in Fig. 1, several are the potential traces (represented by the standard XES [2]) that can be extracted from a relational database. To propose a ranking for the most and least promising traces for later discovery, we propose a set of metrics for detecting hiding processes. To validate our proposal, we have included another metric to measure the quality of the discovered processes, to verify if the metrics for detecting hiding processes are valid.

The paper is organized as follows: Sect. 2 summarises the most relevant related proposals; Sect. 3 presents the real example where our proposal has been applied; Sect. 4 details the method proposed to extract the most promising traces; in Sect. 5 the evaluation of our proposal in a real example is shown; and finally the paper is concluded and future work is explained.

2 Related Work

The relevance of the extraction of event data from databases is widely known [4, 10], and it is an important mechanism to enrich the process mining [22]. For a

general point of view, the analysis of the evolution of the data also represents the activities executed [21]. Previous proposals have analysed the possible relations between the stored data and the business processes [11, 16, 17, 25]. Especially, relational databases have been used as a source of analysis to extract log traces for a later process discovery [22]. In [12], Dijkman et al. apply relational algebra to query the database and extract the log traces. In [8], Berti and van der Aalst include the discovery of Multiple Viewpoint models annotated with performance and frequently information from relational databases. However, their proposal is based on the analysis of the attributes but not in the values of the attributes. In addition, different tools have been implemented to support the trace generation [18, 27] and to retrieve event data from databases, such as OpenSLEX [23].

Not only relational databases have been the analysed source of traces, but also the problem of log extraction from semi-structured sources has been addressed [26]. There are also some studies to ascertain the *case_id* from unstructured data. Bayomie et al. in [7] infer the *case_id* for the unlabeled event logs, and Helal et al. in [19] establish a ranking of possible *case_id* from unlabeled event logs. Nevertheless, none of them infers the *case_id* ranking the information from a relational database.

The necessity to measure the quality of the process is a known problem, and the application of discovery techniques to incorrect or inaccurate data log will generate incorrect or inaccurate business process models [28]. There exist several criteria to assess the data quality in general [6, 24], but centred on data log quality, the Process Mining Manifesto [3] develops a deeper analysis, including safety, completeness, correctness or trustworthiness. Nevertheless, this analysis only includes the quality of the log, but not the evaluation of the quality of the knowledge acquired from the log. In [3] the quality is measured quantitatively. These maturity levels assign the lowest quality when the recorded events do not correspond with the reality, for example, when they are recorded manually. Whereas high quality describes an automatic and complete recovery, reaching the highest quality when every event recorded, and its attributes, have a known semantic meaning about the Business Process Model (BPM). However, this way to measure the quality is not related to the type of processes that could be discovered after the application of a discovery process, the reason why we have defined different metrics to guide in this issue.

To the best of our knowledge, this is the first proposal that tackles the problem of detecting the most promising traces for a later discovery process.

3 Case Study

As outlined in Sect. 1, one of the biggest challenges of the discovery task in process mining is the automatic creation of an event log from a data source. The extraction of these logs from relational databases is an area in which numerous works have been developed as it is analysed in Sect. 2.

However, all these solutions have a common factor: they all require expert knowledge about the domain and the data in order to perform an extraction that

has the potential to be used in subsequent discovery tasks. This fact represents a major problem when we are facing with a large volume of data, which is very common in digitized organizations nowadays. Moreover, this problem seems even more interesting to address when is taken into account that the organization's data expert does not have to be a business process management expert, which means that the data extracted by the data expert could be of little use for process discovery. In the case of the business process management expert, it may take a long time to understand the distribution and semantics of information in order to extract the most relevant in the correct way.

As an example of this scenario, we present our case study: a relational database with more than 300 tables containing the daily operation of assembly and testing processes on aircraft in one of the *Airbus Space & Defence* factories¹, that will serve us to evaluate our proposal to automate the detection of promising processes. More specifically, for our proof of concept, we have focused the experiments on the four fundamental database tables shown in Fig. 2:

- **Aircraft** represents the different aircraft that are assembled and tested. The table contains, among other things, information such as unique identification of the aircraft, its type and version, a serial number and registration, and modification dates and registrars.
- **Functional tests** stores every data related to ground test instructions, which are a representation of the functional tests carried out on aircraft during their assembly process. This table has more than 45 attributes, containing the information as relevant as the unique identifier of the test, its title, code and version, the reasons why it exists, the type of test, various dates and user identifiers related to its creation and modification, subsystem in which it is carried out, different natural text fields describing test specifications, etc.
- **Test execution** keeps the information concerning the execution of the tests described above. So that each record in this table will be associated with a particular functional test and aircraft, as well as a workstation where the test has been executed.
- **Incidents** saves the incidents that occur during each running test, using more than 30 attributes, which indicate the type of incident and its severity, dates and users associated with its registration, modification or cancellation, and various attributes related to observations made in natural text.

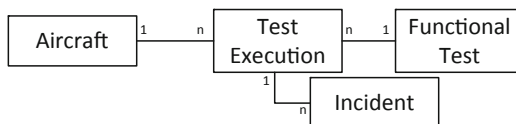


Fig. 2. Case study class diagram.

¹ The data cannot be published due to a confidentiality agreement.

With only these four tables, it is possible to obtain several interpretations of the assembly and testing process, and understand the high number of possible traces that can be created, since it provides us with enough information to be able to discover a wide variety of processes using different *case_id* and *events*. For example, the evolution in the assembly of the aircraft can be analysed according to the workstations in which the tests are executed, the tests can be studied for the type of the incidents that occur during its execution, or the incidents can be analysed based on the functional tests with which they are associated.

To be more precise, with only this small sample of four tables of the database, at least 240 different processes could be discovered. Moreover, the combinatorial nature of the problem has been reduced with: (1) only the primary keys of the tables can be selected as *case_id* since the *case_id* must be a unique value, and; (2) float numbers, dates and Boolean attributes cannot be taken as events since this type of data cannot help to determine the execution of activity of a change of state in a system. This fact highlights the need to find out a way to automate the search for information that can be extracted to create promising processes.

4 Proposal: Method to Detect Promising Processes

The main objective of the proposal consists of analysing the data of the relational database in order to detect hiding business processes that can be promising to know how our system works. The method comprises the following steps:

1. **Analysis of Promising Traces** is the first step to decide which attributes of the database are more suitable to be the *case_id* and to represent the *events* of the promising business processes. In this step, various metrics are necessary to evaluate the traces and select which ones will compose the event logs, before the traces are generated or the processes are discovered. Each part of this analysis is detailed in Subsect. 4.1.
2. **Generation of traces** is executed once the *case_id* and the *events* are selected according to the previous step. The traces are extracted from the relational database by using some of the existing solutions, such as XESame [18] plugin of ProM.
3. **Discovery of Business Process and Assessment of Process Quality** algorithms are applied to the selected traces to discover the promising processes. These business processes should be analyzed to ascertain if they are promising or not. This step is analysis in deep in Subsect. 4.2.

4.1 Analysis of Promising Traces

Many times, the information generated during the execution of a process is stored in a scattered manner in the databases. For example, a test run of an aircraft includes new tuples in a table but can also update values from existing tuples in other tables. This implies that to know how an aircraft evolves can be relevant to analyse the attributes of Table Aircraft but also their tests, execution of

incidents. But hundred are the possible combinations according to the different attributes of each table.

The first step is the analysis of a possible *case_id*, that must be a unique value that comes together with a set of events that represents the execution of an instance of a process. Thus, the primary key of the tables is a unique identifier, as well. In our approach, we consider the primary keys as a good candidate to represent the *case_id*. Besides, since the relationship between two tables in a database is established by using a pair of primary-foreign keys, *case_id* will allow the information of related tables to be included in the case.

On the other hand, an event in a log trace represents the execution of a task in a process instance. In our proposal, the possible events are the values of the attributes that are related to a specific *case_id*. It is worthwhile prioritizing among the attributes that give the most relevant information related to the selected *case_id*. Therefore, those attributes whose values are Float numbers, Dates, or Boolean, can be discarded at the outset since they cannot be understood as a type of task involved in a business model. For example, which activity would represent the *true* value or the date 05/05/2020?. To know what are the possible attributes, the relational database can be represented as a graph (as shown in Fig. 3) where each node represents an attribute, and the edges relate the attributes with its primary key or the relation primary-foreign key. From a primary key attribute, every reachable node can represent the execution of a task (representation of an event), whose timestamp is extracted from the redo log files of the database. To carry out the analysis of which attribute is the most appropriated to represent an event, we analyse the specific values stored in each attribute and its relation with the *case_id*. For the example, if the primary key of Aircraft Table is the *case_id*, and the events are the types of incidents of the Table Incidents, we could know (i) the number of traces (different values of the primary key in aircraft); (ii) the events of each trace (the types of incidences of every text related to each aircraft); and (iii) the different number of events (the different values of the types of incidents). Without creating the trace, we obtain important information about the potential traces, and the possible process discovered later. To measure this information, we present the following set of metrics to know the complexity, diversity, and noise of the different attributes as hypothetical events. The metrics are:

Complexity: the complexity (C) of a process model is understood as the average number of events per trace. A very high number of event per trace will generate too complex processes. However, a very low number of events will produce too simple processes. In order to obtain values between 0 and 1 that allows us to compare all the candidates, it is necessary to normalise (C_n) the values computed in the previous arithmetic operation. To do so, the formula below has been used to penalise both excessively simple and extremely difficult processes and favour those in-between. In the formula, C_i represents the complexity value of a candidate (attribute), C_{q1} and C_{q3} the values of the first and third quartile

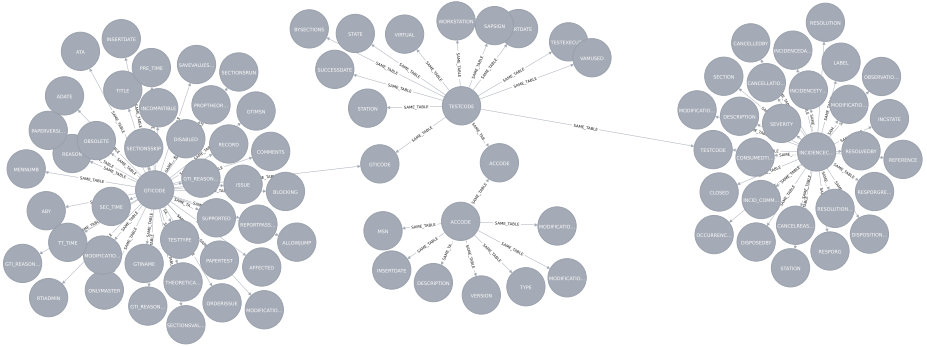


Fig. 3. Relational Database as a graph.

respectively of all the complexities of the chosen sample and C_{max} the maximum of the complexities of the selected sample.

$$C_n(C_i) = \begin{cases} C_i < C_{q1} & (\frac{1}{C_{q1}}) \cdot C_i \\ C_{q1} < C_i < C_{q3} & 1 \\ C_i > C_{q3} & (\frac{-C_i + C_{q3}}{C_{max} - C_{q3}}) + 1 \end{cases} \quad (1)$$

Diversity: the diversity (D) of a process model is the value that represents the density of different events that occur in all the traces among all the events that are presented in the log. This value should also be standardized (D_n) to allow comparison of the quality of different candidates. There is also necessary to penalise proportionally both processes, those in which there is little variety of events and those where is so much variety that is difficult to extract frequent behaviour. The normalisation function is as follows, where D_i represents the diversity of a candidate and D_{mean} , the mean of the diversities of the candidates of the selected sample:

$$D_n(D_i) = \begin{cases} D_i < D_{mean} & (\frac{1}{D_{mean}}) \cdot D_i \\ D_i > D_{mean} & (\frac{-D_i + D_{mean}}{1 - D_{mean}}) + 1 \end{cases} \quad (2)$$

Noise: the noise (N) of a process model means the average of events that only occur once in the whole log among all the events inside of it. The normalised (N_n) value should be measured between $[0..1]$ according to the following function, which rewards candidates in which the presence of noise is minimal:

$$N_n(N_i) = -N_i + 1 \quad (3)$$

4.2 Discovery of Business Processes and Assessment of Process Quality

Once the XES traces are created, different algorithms can be used to discover the process model. In our case, we have applied Inductive Miner techniques using ProM [1, 14]. The noise threshold used in the process discovery is 0.2.

In order to know if the selected traces are promising, we propose to (i) measure the quality of the business processes, (ii) verify the proper functioning of the metrics, and (iii) corroborate that those are promising indeed. To this end, we have defined the **Will Level** metric, as the mean of possible tasks that can be selected in each step according to the process model, divided into the number of total tasks of the process. It is applied to the discovered process model to know how the general it is. For example, in a flowering process, in each step of the instance, any task can be executed, then the metric will level informs about the low use of a process that does not restrict what activity can be executed. The range of the metric is between $[0..1]$, where 0 represents a very restricted process (a sequence of tasks) and 1 a process with XOR-gates that include every task among their branches. The calculation of the Will Level is based on the analysis of the process as a graph, to analyse the possible next activity to execute analysing the possible paths. In order to do that, to obtain the value of the metric the following phases are required:

1. Translation of the process model (i.e. a BPM modelled using the standard BPMN [9] in our case) to a directed labelled graph, where events, gateways and tasks are nodes. The weight of the edges will be 1, if the target of the edge is a task since it means that we have an option, 0 otherwise.
2. Execution of the Dijkstra algorithm [13] to find out all the shortest paths in the graph from the start event and a task to the others.
3. Calculate the will level of each task and the start event as, the number of possible next activities, that are those paths of length equal to 1, divided into the total number of tasks.
4. The calculation of the total Will Level is the arithmetic mean of the Will Level value of each activity node.

5 Evaluation

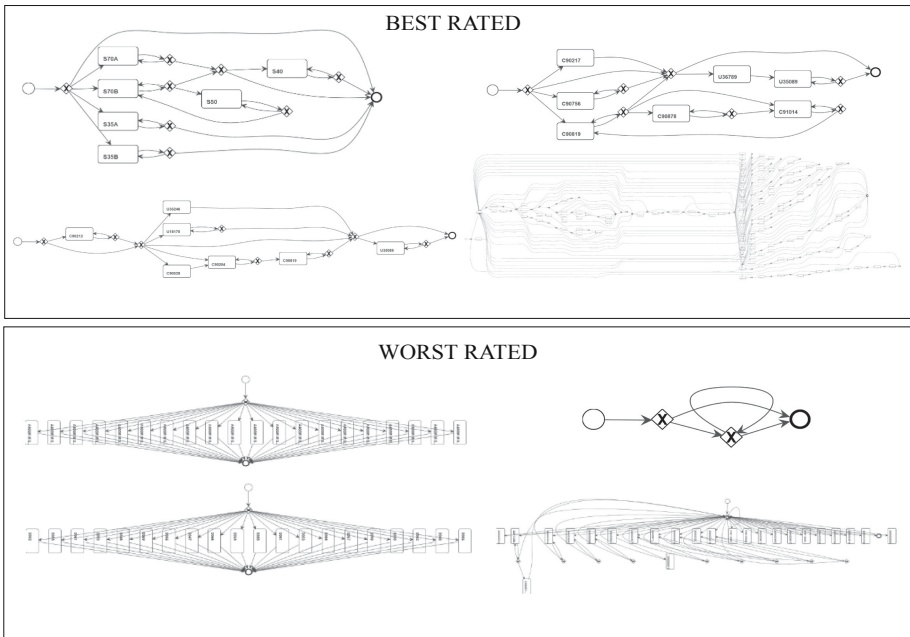
In order to evaluate our proposal, we utilize the data described in Sect. 3 applying the metrics presented in Sect. 4. More specifically, we have made all the possible combinations in which the unique identifier of Table Aircraft acts as *case_id*, while its attributes and those belonging to the other three tables represent the events. Thus, we analysed 58 possible attributes as potential events, before creating the traces used for discovering the processes. The use of our proposal reduces the analysis of those 58 traces, focused on discovering only the most promising according to the metrics.

Table 1 shows the results of the normalised metrics for a selection of 13 for the primary key of the aircraft table as the *case_id*. These 13 represent the 9 best traces (good values for the three metrics), and the 4 worst (bad values for the three metrics).

Some of the discovered processes are shown in Fig. 4. Although the details of the processes cannot be seen, it is possible to observe which are the best rated since

Table 1. Results of the metrics for the detection of hiding processes.

Table	Attribute	Complexity	Diversity	Noise
Incident	Station	0.971	0.777	0.999
Functional test	Name	1	0.970	0.986
Functional test	Reason	1	0.860	0.986
Functional test	Affected	1	0.916	0.999
Functional test	Title	1	0.816	0.998
Functional test	Comments	0.970	0.973	0.985
Functional test	Aby	1	0.893	0.999
Functional test	Modification user	1	0.875	0.999
Functional test	Supported	1	0.988	0.996
Aircraft	Serial Number	0.005	0.094	0
Aircraft	Description	0.005	0.094	0
Incident	Comment	0.176	0.131	0.221
Functional test	Reason reference	0.015	0.142	0.666

**Fig. 4.** Examples of discovered processes.

they are more understandable and relevant processes than the others. The worst-rated are processes with some XOR-gateways with several branches, that represent that any task can be executed.

By measuring the quality of these processes through the will level metric defined previously, we obtain Table 2 that shows the results.

Table 2. Sample of Will Level metric results.

Table	Attribute	Will Level
Incident	Station	0.28
Functional test	Name	0.02
Functional test	Reason	0.06
Functional test	Affected	0.76
Functional test	Title	0.06
Functional test	Comments	0.21
Functional test	Aby	0.21
Functional test	Modification user	0.28
Functional test	Supported	0.05
Aircraft	Serial Number	0.95
Aircraft	Description	0.95
Incident	Comment	-
Functional test	Reason reference	0.81

To assess the results obtained with the proposed methodology, we will rely on the values of the metrics depicted in Tables 1 and 2. In both, the candidates with the best results are placed above the double horizontal line, while those with the worst results are placed below. As can be seen, the candidates rated as best or worst in Table 1 have continued to be classified in the same way with the metrics to discover hidden processes.

In Table 1, it can be seen that all the best-rated attributes have a complexity close to 1, which is positive, as it implies that they have a well-balanced number of events per trace, meaning that they are not excessively simple or too large. Concerning diversity, we found that the best candidates present values greater than 0.75, while the worst are always below 0.2. For the last ones, we can observe that most of the worst-rated candidates have a diversity close to 0, meaning that either they are excessively repetitive traces, or there is so much variety in the distribution of events (infrequent behaviour). Regarding the noise, event logs that have a very low noise level are benefited, and this evidences that the best-rated candidates, that have a noise level very close to 1, have hardly any instances that represent an outlier and can alter the results in the discovery.

The results must be interpreted oppositely in Table 2, the candidates will be better, the lower their value in this metric, since this will indicate high levels of

sequentiality. Special cases, such as ‘*Affected*’ or ‘*Reason reference*’, whose values are slightly close to 1, might indicate that this type of event log would offer better results if declarative rather than imperative models are used. This conclusion comes since their traces reflect that they are very permissive processes but have certain restrictions. The attribute *Comment* does not have will level defined since the level of noise is so high that no activities are discovered by using Inductive Miner. Once the most promising processes have been obtained, the next step would be to ask the business experts which of them could really be used because they are really useful for the business. This step is outside the scope of this article but will be a future work.

6 Conclusion and Future Work

This paper presents a method to guide in the detection of hiding processes by analysing the information of the relational database that contains the data produced by a process. To extract the most promising processes, hiding into the data, some metrics have been proposed based on the number of traces, events, and frequency of them, aligned with the metrics of complexity, diversity, and noise. The analysis of these metrics provides a ranking to ascertain, for each *case_id*, what are the possible event logs that could be interesting to participate in a discovery process. The validation of our proposal is focused on the analysis of the relevance of the obtained processes, using the evaluation of an expert and measuring the level of will that represents the discovered process. According to these metrics, our proposal has been ratified. For the future, we consider extending the types of metrics both before and after the processes discovery. Moreover, analysing the database structure to infer new possible indicators that help to infer the most promising processes.

Acknowledgement. This research was partially supported by Ministry of Science and Technology of Spain with projects ECLIPSE (RTI2018-094283-B-C33) and by Junta de Andalucía with METAMORFOSIS projects; and by European Regional Development Fund (ERDF/FEDER).

References

1. Prom tool. <http://www.promtools.org/doku.php>
2. IEEE standard for extensible event stream (XES) for achieving interoperability in event logs and event streams. IEEE Std 1849–2016, pp. 1–50 (2016)
3. van der Aalst, W., et al.: Process mining manifesto. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM 2011. LNBIP, vol. 99, pp. 169–194. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28108-2_19
4. van der Aalst, W.M.P.: Extracting event data from databases to unleash process mining. In: BPM - Driving Innovation in a Digital World, pp. 105–128 (2015)
5. Aalst, W.: Data science in action. Process Mining, pp. 3–23. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49851-4_1

6. Batini, C.: Data quality assessment. In: Liu, L., Özsu, M.T. (eds.) *Encyclopedia of Database Systems*, 2nd edn. Springer, New York (2018). https://doi.org/10.1007/978-0-387-39940-9_107
7. Bayomie, D., Helal, I.M.A., Awad, A., Ezat, E., ElBastawissi, A.: Deducing case ids for unlabeled event logs. In: Reichert, M., Reijers, H.A. (eds.) *Business Process Management Workshops*, pp. 242–254. Springer, Cham (2016)
8. Berti, A., van der Aalst, W.M.P.: Extracting multiple viewpoint models from relational databases. *CoRR abs/2001.02562* (2020)
9. Business process model and notation (BPMN) version 2.0.2. Standard, Object Management Group Standard (2014)
10. Calvanese, D., Kalayci, T.E., Montali, M., Santoso, A.: OBDA for log extraction in process mining. In: Reasoning Web. Semantic Interoperability on the Web - 13th International Summer School 2017, London, UK, 7–11 July 2017, Tutorial Lectures, pp. 292–345 (2017)
11. Calvanese, D., Montali, M., Syamsiyah, A., van der Aalst, W.M.P.: Ontology-driven extraction of event logs from relational databases. In: Reichert, M., Reijers, H.A. (eds.) *BPM 2015. LNBIP*, vol. 256, pp. 140–153. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42887-1_12
12. Dijkman, R., Gao, J., Syamsiyah, A., van Dongen, B., Grefen, P., ter Hofstede, A.: Enabling efficient process mining on large data sets: realizing an in-database process mining operator. *Distributed and Parallel Databases* **38**(1), 227–253 (2019). <https://doi.org/10.1007/s10619-019-07270-1>
13. Dijkstra, E.W., et al.: A note on two problems in connexion with graphs. *Numerische mathematik* **1**(1), 269–271 (1959)
14. van Dongen, B.F., de Medeiros, A.K.A., Verbeek, H.M.W., Weijters, A.J.M.M., van der Aalst, W.M.P.: The ProM framework: a new era in process mining tool support. In: Ciardo, G., Darondeau, P. (eds.) *ICATPN 2005. LNCS*, vol. 3536, pp. 444–454. Springer, Heidelberg (2005). https://doi.org/10.1007/11494744_25
15. Dumas, M., Rosa, M.L., Mendling, J., Reijers, H.A.: *Fundamentals of Business Process Management*. Springer, Heidelberg (2013). <https://doi.org/10.1007/978-3-642-33143-5>
16. Gómez-López, M.T., Borrego, D., Gasca, R.M.: Data state description for the migration to activity-centric business process model maintaining legacy databases. In: *BIS*, pp. 86–97 (2014)
17. Gómez-López, M.T., Reina Quintero, A.M., Parody, L., Pérez Álvarez, J.M., Reichert, M.: An architecture for querying business process, business process instances, and business data models. In: Teniente, E., Weidlich, M. (eds.) *BPM 2017. LNBIP*, vol. 308, pp. 757–769. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-74030-0_60
18. Günther, C.W., van der Aalst, W.M.P.: A generic import framework for process event logs. In: Eder, J., Dustdar, S. (eds.) *BPM 2006. LNCS*, vol. 4103, pp. 81–92. Springer, Heidelberg (2006). https://doi.org/10.1007/11837862_10
19. Helal, I.M.A., Awad, A., El Bastawissi, A.: Runtime deduction of case id for unlabeled business process execution events. In: *2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*, pp. 1–8 (2015)
20. Kalpic, B., Bernus, P.: Business process modelling in industry - the powerful tool in enterprise management. *Comput. Ind.* **47**(3), 299–318 (2002)
21. Li, G., de Murillas, E.G.L., de Carvalho, R.M., van der Aalst, W.M.P.: Extracting object-centric event logs to support process mining on databases. In: Mendling, J., Mouratidis, H. (eds.) *CAiSE 2018. LNBIP*, vol. 317, pp. 182–199. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-92901-9_16

22. de Murillas, E.G.L., Reijers, H.A., van der Aalst, W.M.P.: Connecting databases with process mining: a meta model and toolset. *Software & Systems Modeling* (2018)
23. González López de Murillas, E., Reijers, H.A., van der Aalst, W.M.P.: Connecting databases with process mining: a meta model and toolset. In: Schmidt, R., Guédria, W., Bider, I., Guerreiro, S. (eds.) *BPMDs/EMMSAD -2016*. LNBIP, vol. 248, pp. 231–249. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39429-9_15
24. Otto, B., Lee, Y.W., Caballero, I.: Information and data quality in networked business. *Electron. Markets* **21**(2), 79–81 (2011). <https://doi.org/10.1007/s12525-011-0062-2>
25. Pérez-Alvarez, J., Gómez-López, M., Eshuis, R., Montali, M., Gasca, R.: Verifying the manipulation of data objects according to business process and data models, January 2020
26. Valencia-Parra, Á., Ramos-Gutiérrez, B., Varela-Vaca, A.J., Gómez-López, M.T., Bernal, A.G.: Enabling process mining in aircraft manufactures: extracting event logs and discovering processes from complex data. In: *Proceedings of the Industry Forum at BPM 2019*, Vienna, Austria, September 1–6, 2019, pp. 166–177 (2019)
27. Verbeek, H.M.W., Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: XES, XESame, and ProM 6. In: *Information Systems Evolution - CAiSE Forum 2010*, Hammamet, Tunisia, June 7–9, 2010, Selected Extended Papers, pp. 60–75 (2010)
28. Wynn, M.T., Sadiq, S.: Responsible process mining - a data quality perspective. In: Hildebrandt, T., van Dongen, B.F., Röglinger, M., Mendling, J. (eds.) *Business Process Management*, pp. 10–15. Springer, Cham (2019)



Analysis of Language Inspired Trace Representation for Anomaly Detection

Gabriel Marques Tavares¹(✉)  and Sylvio Barbon Jr.² 

¹ Università degli Studi di Milano (UNIMI), Milan, Italy
`gabriel.tavares@unimi.it`

² Londrina State University (UEL), Londrina, Brazil
`barbon@uel.br`

Abstract. A great concern for organizations is to detect anomalous process instances within their business processes. For that, conformance checking performs model-aware analysis by comparing process logs to business models for the detection of anomalous process executions. However, in several scenarios, a model is either unavailable or its generation is costly, which requires the employment of alternative methods to allow a confident representation of traces. This work supports the analysis of language inspired process analysis grounded in the word2vec encoding algorithm. We argue that natural language encodings correctly model the behavior of business processes, supporting a proper distinction between common and anomalous behavior. In the experiments, we compared accuracy and time cost among different word2vec setups and classic encoding methods (token-based replay and alignment features), addressing seven different anomaly scenarios. Feature importance values and the impact of different anomalies in seven event logs were also evaluated to bring insights on the trace representation subject. Results show the proposed encoding overcomes representational capability of traditional conformance metrics for the anomaly detection task.

Keywords: Anomaly detection · Encoding · Business process · Natural language processing

1 Introduction

The assessment of anomalous business process executions is a real concern for organizations. Naturally, by detecting and mitigating wrongly executed processes, enterprises avoid frauds, save resources, and refine their methods. For that, companies rely on process-aware information systems, which is the group of software systems that support and control business processes [2]. Counting with a process notion, the operation of such software generates a log of events, i.e., the recording of activities within a process. From that, process analysis can be performed on such logs. Process Mining (PM) is the area aimed at extracting information and producing analysis starting from process data. In the PM realm,

the *event log* is the set of events executed under a business process. Furthermore, an event records the execution of an *activity* at a given *time*. Finally, each event relates to a single process instance, referred to as a *case*.

Traditionally, conformance checking is the PM task aimed at evaluating the behavior quantitatively. Conformance methods compare the process model to the event log, checking for deviations and further identifying anomalous instances [19]. Contrasting a process model with event log data is a common way of detecting anomalies. However, this is not feasible in scenarios without a process model. Thus, there is a need for algorithms that infer data patterns without any prior domain knowledge.

Given the necessity of detecting anomalous traces and the limitations of traditional approaches, we propose the usage of a language inspired trace representation. The method's core is based on Natural Language Processing (NLP) encoding of textual data. For that, we map activities and traces as words and sentences, respectively, before applying the word2vec encoding algorithm [13]. Word2vec captures contextual information, i.e., it models activities surroundings, such that each activity is represented by a vector. Traces with uncommon encoding are potential anomalies. We took advantage of the Random Forest machine learning algorithm to induce models for trace classification. Then, Random Forest's importance of features was used to support the discussion about encoding descriptive patterns.

The following sections are organized as follows. Section 2 presents anomaly detection works and encoding attempts in PM research. Section 3 reports (i) event log generation and anomaly injection, (ii) word2vec encoding and classical conformance metrics extraction, and (iii) the experimental setup for anomaly detection in business processes. Section 4 shows the obtained results, compares the encoding strategies, and evaluates how different anomaly types are perceived. The analysis is supported by accuracy, time and features importance metrics. Section 5 concludes the paper and leaves the final remarks.

2 Related Work

Anomaly detection in business process data has been extensively explored in recent years [3, 7, 14–17]. From the traditional PM pillars, conformance checking methods are the most used for anomaly detection. Conformance techniques rely on the comparison between a process model and an event log [2]. It follows that non-complying business cases can be interpreted as anomalous. This compliance is measured by the use of constraint satisfaction or transition marks, being employed either to control-flow or data-flow perspectives.

One of the earliest works for anomaly detection in PM uses a conformance checking pipeline [6]. The method filters the log based on domain-dependent knowledge and applies process discovery techniques to the filtered log. The most appropriate model is chosen as the process model. Thus, traces are classified depending on model fitting; that is, a non-fitting trace is classified as anomalous. However, this approach depends on a clean event log for model creation and

assumes that process discovery techniques might generate an ideal model, which is not necessarily true. Moreover, domain knowledge costs resources and is not always available.

More recent approaches explore the use of likelihood graphs to model process behavior and detect anomalous instances [7]. The encoding models both control and data-flow perspectives, and cases deviating from observed probability are classified as anomalous. However, the method introduces bias by connecting attributes to the graph, while the probabilities for activities execution are not connected to their attributes, which is inconsistent with real scenarios. The efficiency of this approach highly depends on the discovered process model quality, which is subjective in most cases. There is no consensus on whether the discovered model best represents log behavior, as the process of discovering a model comes from a trade-off between precision and generality.

Another family of approaches emerges from Machine Learning (ML) methods applied in business process contexts. In [14, 16], the authors use an autoencoder to model process behavior. The technique encodes the event log using one-hot encoding and trains the autoencoder using the log as both the input and the output. The mean squared error between the input and output is measured and given a threshold, anomalous instances are highlighted. The main drawback of the approach is that vector sizes increase linearly with the number of activities, which is costly resource-wise. Moreover, the one-hot encoding technique produces very sparse vectors, further increasing computational overhead. To overcome this issue, in [15, 17], the authors proposed a deep learning method considering both control and data-flow perspectives. The technique uses a deep neural network trained to predict the next event. Given the network probability score, an activity or attribute with a low execution probability is interpreted as an anomaly. However, the computational cost of deep learning methods is very high, which hinders its application in many scenarios.

Several techniques explore encoding, given that trace context is a determinant factor for anomaly detection. However, advanced deep learning methods demand high resource consumption and have limited interpretability. At the same time, traditional conformance approaches depend on model discovery, which is a challenging task. Our work bridges the gap between these two types of approaches by using a light-weight encoding technique based on NLP, taking advantage of activities context within a trace. The computational burden is considerably inferior when compared to deep learning methods, and without the need of a process model. Thus, combining the best aspects from conformance checking and deep learning approaches.

3 Methodology

3.1 Event Logs

One of the main goals of the experiments is to compare traditional trace metrics with trace encodings based on natural language models. Thus, a controlled scenario with known labels is the best way to evaluate the different modeling

approaches. Using the proposed framework for event log generation in [15], we created several business process logs based on the provided guidelines.

The PLG2 tool [9] was used to generate six random process models exploring various complexities, such as the number of activities, breadth and width. Moreover, a handmade procurement process model (P2P [17]) was added to the model pool. A likelihood graph [7] was adopted to introduce long-term control-flow dependencies, a common characteristic of real-world process logs. Such dependencies regard event to event transitions but also include event to attribute relations. This way, the probability distributions are constrained. For instance, an activity A has a determined probability of being followed by activity B . This can be extended to event attributes, such as an attribute C has a probability of being logged when activity A is executed in specific conditions. The combinations within a likelihood graph are extensive, thus providing a complex graph, mimicking real-world conditions and scenarios.

This way, models created in PLG2 are extended to likelihood graphs. From that, random walks in the graph generate the event log. Note that the random walks comply with transition probabilities, both for control-flow and data perspectives. The next step is to inject anomalous traces in the event log, a traditional practice in the literature [6, 7]. As in the reference work [15], six elaborate anomaly types were applied: 1) Skip: a sequence of 3 or less necessary events is skipped; 2) Insert: 3 or less random activities inserted in the case; 3) Rework: a sequence of 3 or less necessary events is executed twice; 4) Early: a sequence of 2 or fewer events executed too early, which is then skipped later in the case; 5) Late: a sequence of 2 or fewer events executed too late, which is then skipped later in the case; 6) Attribute: an incorrect attribute value is set in 3 or fewer events.

The artificial anomalies are applied to 30% of the cases from previously generated event logs. The ground truth label is on the event level, however, it can be easily converted to the case level. Whenever a case has an anomalous event, the respective case is labeled as an anomaly. Table 1 reports the detailed event log statistics. Finally, note that the recreation of these event logs and their anomalies is replicable by following the steps reported in the original work [15].

Table 1. Event log statistics: each log contains different levels of complexity

Name	#Logs	#Activities	#Cases	#Events	#Attributes	#Attribute values
P2P	4	27	5k	48k–53k	1–4	13–386
Small	4	41	5k	53k–57k	1–4	13–360
Medium	4	65	5k	39k–42k	1–4	13–398
Large	4	85	5k	61k–68k	1–4	13–398
Huge	4	109	5k	47k–53k	1–4	13–420
Gigantic	4	154–157	5k	38k–42k	1–4	13–409
Wide	4	68–69	5k	39k–42k	1–4	13–382

ML techniques operate at the instance level, i.e., an event is a complete instance representation, while PM methods operate at the business case level, i.e., the group of events from the same case composes an instance. Due to this mismatch at the representation level, traditional ML algorithms can not be directly applied to business process logs [20,21]. This way, an encoding layer extracting case features is necessary to overcome this issue, thus, merging the gap between process science and ML methods.

Log encoding was already explored in the literature [14,16]. In [14], the authors use one-hot encoding whereas in [15,17], the authors use integer encoding. Both approaches transform the event log, and consequently traces, into a numerical representation, which is then fed to ML algorithms. However, one-hot and integer encodings generate sparse data. In [10], the authors represent a trace using the NLP *doc2vec* encoding [11]. Though the technique is innovative, its activity and trace representation lacks context as the *doc2vec* encoding is designed for paragraphs, which generally contain more data than a trace.

Inspired by NLP research, which has solid literature in representational learning, we propose the use of the *word2vec* encoding method to capture business process behavior [13]. Word2vec produces word encodings using a two-layer neural network aimed at reconstructing the linguistic context for each word in the corpus. The produced vectors model semantic and syntactic characteristics using the weights produced by the neural networks. Namely, words with similar contexts have analogous vectors. This way, our approach interprets each activity as being a word. By consequence, the set of unique activities is the corpus used by the word2vec model. Within a model, each activity is represented by a numerical vector describing its context. To retrieve trace-level encoding, we aggregate the word vectors that compose a respective trace. For that, we use element-wise mean, i.e., the trace representation is the mean of its activities representations.

3.2 Traditional Feature Engineering

A classical approach to evaluate business cases uses conformance checking techniques. Conformance aims to compare a process model to the event log and measure their differences [2]. This process must account for the alignment between trace and model elements. Here, we employ two of the most traditional conformance approaches: token-based replay and alignment. Token replay matches a trace to a process model in the Petri net format. The resulting value is usually a fitness score produced by firing tokens and accounting the mismatch between trace and model-allowed transitions. Further, following a comparison approach, trace alignment links trace activities into Petri net transitions [1]. The alignment is measured by comparing a trace to log moves, accounting moves that can or not be mimicked in the log, e.g., skipped and silent activities.

Both conformance techniques were implemented using the PM4Py Python package [5], following standard hyperparameters¹. Table 2 shows the extracted

¹ <https://pm4py.fit.fraunhofer.de/documentation>.

features for each trace. These features are used for model creation and posterior anomaly detection. Moreover, both techniques require a Petri net model to compute conformance measures. This way, before applying feature extraction, we induct a process discovery algorithm using the Inductive Miner Directly Follows (IMDF) algorithm [12] (employing PM4Py library). IMDF was chosen since its goal is to construct a sound model with good fitness values, leveraging the quality of extracted features.

Table 2. Extracted trace features to describe trace behavior. There are two types of features: token replay and alignment

Feature type	Feature	Meaning
Token replay	<i>trace_is_fit</i>	Indicates if the trace fits the model
Token replay	<i>trace_fitness</i>	Trace fitness value
Token replay	<i>missing_tokens</i>	Number of missing tokens
Token replay	<i>consumed_tokens</i>	Number of consumed tokens
Token replay	<i>remaining_tokens</i>	Number of remaining tokens
Token replay	<i>produced_tokens</i>	Total number of tokens produced
Alignment	<i>cost</i>	Cost of the alignment
Alignment	<i>visited_states</i>	Number of visited states
Alignment	<i>queued_states</i>	Number of queued states
Alignment	<i>traversed_arcs</i>	Number of traversed arcs
Alignment	<i>fitness</i>	Trace fitness value

3.3 Experimental Setup

The experiments aim at measuring two main aspects: (i) how different anomalies behave and how much they affect classification performance, and (ii) to which extent traditional and natural language inspired encodings can represent log behavior, including anomalies. For that, we designed two sets of experiments. The first is a binary classification using only the normal class and one anomaly (this is replicated for all anomalies). The second experiment is a multi-class detection task, thus, using all the available classes. The latter experiment is more challenging as the encodings need to represent all different behaviors at the same time, considering that anomalies tend to harm encoding quality.

For the word2vec encoding, we explored several vector sizes (25, 50, 100, 200, 400), this way, evaluating if more complex vectors capture better trace behavior. The Random Forest (RF) [8] algorithm was used for the classification task following the *scikit-learn* Python package [18]. RF was selected due to its extensive use in ML literature. RF is very robust and controls overfitting with its ensemble nature. Moreover, RF requires less computational resources compared to deep

learning methods. To provide a common testbed for the different encodings, we implemented a grid search method for hyperparameter tuning, a standard technique in literature [4]. Grid search trials are formed by assembling all possible hyperparameter values combination. Table 3 lists the explored hyperparameters, their implication and employed values.

Table 3. Random Forest hyperparameters. A grid search method was used to combine all possible hyperparameter values, yielding an optimal performance

Hyperparameter	Meaning	Values
<i>n_estimators</i>	Number of forest trees	50, 100, 250, 500, 750, 1000
<i>max_features</i>	Number of features to consider for the best split	auto, log2
<i>max_depth</i>	Maximum tree depth	4, 8, 16, 32, 64, default
<i>criterion</i>	Function to measure split quality	gini, entropy

4 Results and Discussion

All the discussion in this section was made using a RF model induced using *n_estimators* of 50, *max_features* with log2, *max_depth* with default value and entropy as *criterion*. These values were found after the tuning procedure. The time presented was computed during the model induction period.

4.1 Overall Performance

An overview of predictive performance is exposed in Fig. 1. The accuracy to detect anomalies is sorted from left to right. The lowest performance was reported in the logs with all anomalies concurrently. Late, attribute and early anomalies followed with similar performance. The most accurate classifications were made over the insert, rework and skip anomalies.

It is easy to comprehend the low performance of all anomalies scenario since it is a multi-class problem in which the model needs to deal with seven different outcomes (common behavior and six different anomalies). Aggregating the performance of all models by each encoding, word2vec methods obtained superior performance (average of 84.7%) in comparison to the classic method (76.3%). No specific word2vec length outperformed the others, the obtained accuracies were 84.6%, 84.6%, 84.7%, 84.7% and 84.9% by 100, 200, 25, 50 and 400, respectively.

Late, attribute, and early anomalies were classified with a similar predictive performance by the RF models. Word2vec methods obtained an average accuracy of 93.3% while the classic method achieved a slightly superior performance of 93.5%. The most predictable anomalies were insert, rework and skip, where word2vec encoding obtained 99.8% accuracy. The standard deviation of $1e^{-3}$ within word2vec models shows that different lengths did not affect performance. Contrarily, classic encoding obtained inferior results, an average of

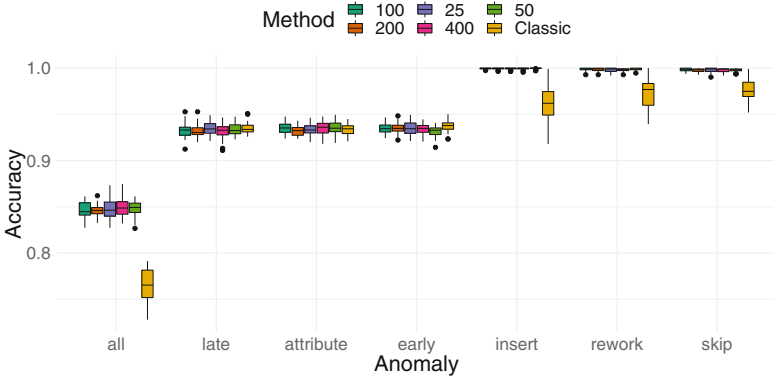


Fig. 1. Comparison of accuracy among word2vec using different lengths (25, 50, 100, 200 and 400) and classic encoding methods when representing seven different anomaly scenarios (all anomalies concurrently, late, attribute, early, insert, rework and skip)

96.9% ($\pm 1e^{-4}$), with insert as the less accurate classification (96.1%). Rework and skip obtained (97.1%) of accuracy. Overall, word2vec encoding had a better performance when compared to classical encoding.

Beyond the predictive performance, we performed a time analysis to deepen our discussion about the suitable method. Figure 2 presents violin plots built using the time to induce a RF model using different encoding methods. We choose this visualization to illustrate the average performance and to clarify some possible distortions of time due to different datasets compromised with the same anomaly.



Fig. 2. Comparison of time among word2vec using different lengths (25, 50, 100, 200 and 400) and classic encoding method when representing seven different anomaly scenarios (all anomalies concurrently, late, attribute, early, insert, rework and skip)

Classic encoding takes a similar time for all anomalies, an average of 0.23 s ($\pm 6e^{-3}$), to induce a RF model, being the fastest encoding technique. Word2vec encodings required more time for particular anomalies (e.g., all concurrently, late, attribute and early), an average of 0.35 s (± 0.07). When modeling a classifier to the all anomalies scenario, this difference becomes more evident since classic encoding obtained 0.28 s and word2vecs 0.39 s, 0.40 s, 0.43 s, 0.48 s and 0.55 s for 25, 50, 100, 200 and 400, respectively. Regarding word2vec time performance, it is possible to observe a straightforward relation between time and feature vector length, as expected. An analogous relation was observed between feature vector length and stability of the model. Using the standard deviation of induction time for each word2vec range, we found the same behavior, the larger feature vector, the higher time variability. The most stable encoding was the smallest (25 features).

4.2 Encoding Strategy and Feature Importance

One of the key-points of an encoding method is its capability to support class disjunctions. Taking advantage of the RF models, we observed the importance of each variable grounded on RF importance. Figure 3 presents an overview of RF feature importance for classic encoding. The alignment family of features (*traversed_arcs*, *cost*, *visited_states*, *queued_states*) was the most important group, except by *fitness*. Token replay features did not contribute to the model.



Fig. 3. RF feature importance from alignment and token replay encodings

The limited performance of token replay features happens as the approach tend to consistently produce high fitness values. Moreover, a path through the model is not created for non-fitting cases. Therefore, the same values for token replay features are produced to the majority of traces, adding no benefit in the trace encoding. Consequently, their significance for classification is downgraded,

which is represented by the low RF importance. On the other hand, the alignment family of features overcomes these limitations by introducing more robust rules [1], thus, being decisive in anomaly detection.

Figure 4 displays RF feature importance of word2vec encodings. No optimal subset of features can be found. Further, the importance of features spreads as vector length grows, i.e., the spike of importance had its value reduced as vector length enlarged.

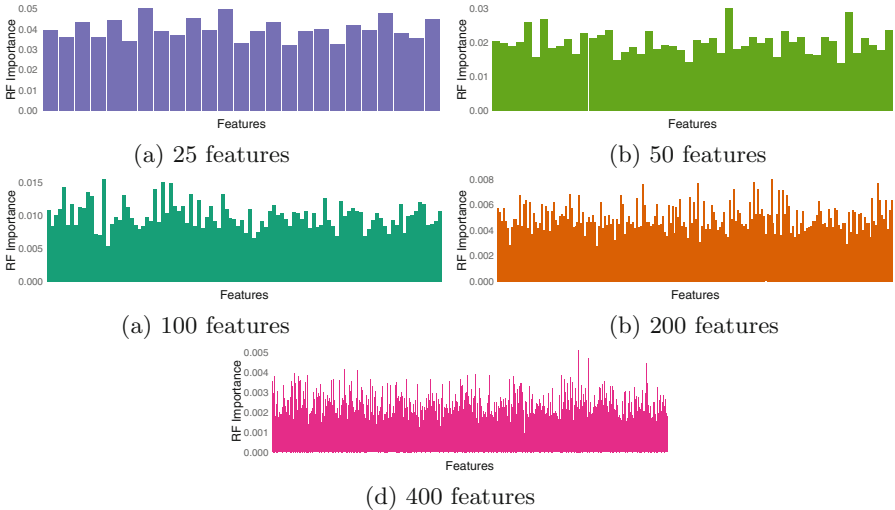


Fig. 4. RF feature importance from five different word2vec lengths

The distribution of importance values over the feature vector exposed an intrinsic characteristic of word2vec encoding, the capacity of exploring the problem using an inputted granularity. In other words, this encoding method dispersed throughout all features the capacity of trace representation. Complex problems that demand a more precise description can take advantage of large feature vectors, at the cost of time and stability.

4.3 Anomalies Analysis

Figure 5 presents the average accuracy from all word2vec models over each event log. From this, *Gigantic* appears as the most complex event log with lower accuracies, obtaining its lowest classification performance when compromised by all anomalies. This is explained by the log complexity, which can be observed by the high number of activities (Table 1). Higher performances were observed over *Large*, *Small* and *Wide* event logs. Although there exists an accuracy variation on the datasets classification performance, we need to mention the most difference was presented in the anomaly perspective. When comparing the

average accuracy per dataset, a variation of 1% was observed. In contrast, the average accuracy difference within anomalies stood at 16%.



Fig. 5. Average accuracy obtained with word2vec encoding for each combination of dataset and anomaly

According to Fig. 5, late, attribute, and early anomalies are the most difficult of being recognized. Late and early anomalies affect an activity being executed before or after, respectively, its expected execution. This way, the trace context is affected more subtly because the activity is being executed, even if in the wrong position. The attribute anomaly is challenging as it affects the data-flow perspective, so methods that do not consider this flow tend to present lower performances. The skip, rework and insert anomalies affect more profoundly the control-flow perspective. This aggressive behavior is easily detected by the encoding methods, e.g., even the classical method was able to perform better in these anomalies (Fig. 1).

5 Conclusion

Conformance checking is one of the most important tasks of PM in real-life business applications, mainly for anomaly detection. This paper compared classic conformance features (token replay and alignment) with language inspired trace representation (word2vec) over different event logs compromised in seven different scenarios. A RF classification model was combined with the encoding techniques for the anomaly detection task. Word2vec correctly captured trace context and demonstrated a better performance than classic encodings. The most challenging scenario is dealing with all anomalies concomitantly, where lower accuracies were achieved. On the other hand, detecting *insert*, *rework*, and *skip* anomalies leveraged accuracy performance. In both extreme scenarios, i.e., higher and lower accuracies, word2vec overcame classic encoding. The performance results of encoding methods were similar just for medium-range accuracy scenarios (*late*, *attribute*, and *early* anomalies). Regarding different event

logs, *Gigantic* was the most complex. When comparing the importance of the classic encoding techniques, alignment features prevailed token replay features. As future work, we plan to investigate anomaly detection in online settings.

References

1. van der Aalst, W., Adriansyah, A., van Dongen, B.: Replaying history on process models for conformance checking and performance analysis. *WIREs Data Min. Knowl. Disc.* **2**(2), 182–192 (2012)
2. van der Aalst, W.M.P.: *Process Mining: Data Science in Action*, 2nd edn. Springer, Heidelberg (2016). <https://doi.org/10.1007/978-3-662-49851-4>
3. Barbon Junior, S., Tavares, G.M., da Costa, V.G.T., Ceravolo, P., Damiani, E.: A framework for human-in-the-loop monitoring of concept-drift detection in event log stream. In: *Companion Proceedings of the The Web Conference 2018*, pp. 319–326. WWW 2018, International World Wide Web Conferences Steering Committee (2018)
4. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **13**(10), 281–305 (2012)
5. Berti, A., van Zelst, S.J., van der Aalst, W.: *Process mining for python (pm4py): Bridging the gap between process- and data science* (2019)
6. Bezerra, F., Wainer, J.: Algorithms for anomaly detection of traces in logs of process aware information systems. *Inf. Syst.* **38**(1), 33–44 (2013)
7. Böhmer, K., Rinderle-Ma, S.: Multi-perspective anomaly detection in business process execution events. In: Debruyne, C., et al. (eds.) *OTM 2016*. LNCS, vol. 10033, pp. 80–98. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48472-3_5
8. Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001)
9. Burattin, A.: *Plg2: Multiperspective processes randomization and simulation for online and offline settings* (2015)
10. De Koninck, P., vanden Broucke, S., De Weerd, J.: *act2vec, trace2vec, log2vec, and model2vec: representation learning for business processes*. In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) *BPM 2018*. LNCS, vol. 11080, pp. 305–321. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98648-7_18
11. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: *Proceedings of the 31st International Conference on International Conference on Machine Learning*, vol. 32. p. II-1188-II-1196. ICML 2014, JMLR.org (2014)
12. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Scalable process discovery with guarantees. In: Gaaloul, K., Schmidt, R., Nurcan, S., Guerreiro, S., Ma, Q. (eds.) *Enterprise, Business-Process and Information Systems Modeling*, pp. 85–101. Springer, Cham (2015)
13. Mikolov, T., Chen, K., Corrado, G.S., Dean, J.: Efficient estimation of word representations in vector space. *CoRR abs/1301.3781* (2013)
14. Nolle, T., Luettgen, S., Seeliger, A., Mühlhäuser, M.: Analyzing business process anomalies using autoencoders. *Mach. Learn.* **107**(11), 1875–1893 (2018)
15. Nolle, T., Luettgen, S., Seeliger, A., Mühlhäuser, M.: Binet: multi-perspective business process anomaly classification. *Inf. Syst.* **1**, 101458 (2019)
16. Nolle, T., Seeliger, A., Mühlhäuser, M.: Unsupervised anomaly detection in noisy business process event logs using denoising autoencoders. In: Calders, T., Ceci, M., Malerba, D. (eds.) *DS 2016*. LNCS (LNAI), vol. 9956, pp. 442–456. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46307-0_28

17. Nolle, T., Seeliger, A., Mühlhäuser, M.: BINet: multivariate business process anomaly detection using deep learning. In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) BPM 2018. LNCS, vol. 11080, pp. 271–287. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98648-7_16
18. Pedregosa, F., et al.: Scikit-learn: machine learning in python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
19. Rozinat, A., van der Aalst, W.: Conformance checking of processes based on monitoring real behavior. *Inf. Syst.* **33**(1), 64–95 (2008)
20. Tavares, G.M., Ceravolo, P., Turrise Da Costa, V.G., Damiani, E., Barbon Junior, S.: Overlapping analytic stages in online process mining. In: 2019 IEEE International Conference on Services Computing (SCC), pp. 167–175, July 2019
21. Tavares, G.M., Turrise Da Costa, V.G., Martins, V., Ceravolo, P., Barbon Junior, S.: Leveraging anomaly detection in business process with data stream mining. *iSys - Revista Brasileira de Sistemas de Informação* **12**(1), 54–75 (2019)

**The 1st International Workshop
on Assessing Impact and Merit
in Science (AIMinScience 2020)**



Exploring Citation Networks with Hybrid Tree Pattern Queries

Xiaoying Wu^{1(✉)}, Dimitri Theodoratos², Dimitrios Skoutas³,
and Michael Lan²

¹ Computer School of Wuhan University, Wuhan, China
xiaoying.wu@whu.edu.cn

² New Jersey Institute of Technology, Newark, USA
{dth,m1122}@njit.edu

³ Athena R.C., Marousi, Greece
dskoutas@athenarc.gr

Abstract. Scientific impact of publications is often measured using citation networks. However, traditional measures typically rely on direct citations only. To fully leverage citation networks for assessing scientific impact, it is necessary to investigate also indirect scientific influence, which is captured by citation paths. Further, the analysis and exploration of citation networks requires the ability to efficiently evaluate expressive queries on them. In this paper, we propose to use hybrid query patterns to query citation networks. These allow for both edge-to-edge and edge-to-path mappings between the query pattern and the graph, thus being able to extract both direct and indirect relationships. To efficiently evaluate hybrid pattern queries on citation graphs, we employ a pattern matching algorithm which exploits graph simulation to prune nodes that do not appear in the final answer. Our experimental results on citation networks show that our method not only allows for more expressive queries but is also efficient and scalable.

1 Introduction

The constantly growing volume of scientific publications across all disciplines, increases the interest and need for methods and algorithms to measure, assess and analyze their impact. This applies not only to the scientific community itself but also to practitioners, policy makers, and other professionals. A common means to measure scientific impact of publications is through citation networks, i.e., graphs where nodes correspond to publications and edges correspond to citations. Algorithms such as PageRank can be used to rank publications in citation networks. However, it is often not sufficient to simply assign a ranking score to each publication. Users may need to explore the network in order to

X. Wu—The research was supported by the National Natural Science Foundation of China under Grant No. 61872276.

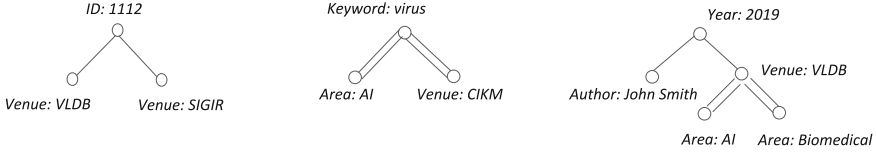


Fig. 1. Hybrid tree pattern queries on a citation network.

gain more insights about the links and relations between publications, e.g., to identify lines of research, discover which publications have been influenced by others, or reveal lines of work that have converged or diverged.

Traditional measures for the scientific impact of research papers mainly rely on direct citations only. A published paper creates some new knowledge based on absorbing knowledge produced by its references, and at the same time also passes its own knowledge to future papers through citations. Thus, paths of citations are formed. The information flow through these paths represents the knowledge flow in the underlying scientific domain [14]. Hence, to leverage citation networks for assessing scientific impact, it is necessary to investigate indirect scientific influence, which is captured by citation paths [8].

Indirect scientific influence has received increasing attention recently [5, 6, 8]. One such impact indicator is the single-publication h-index [10, 15], which relies on a paper’s second generation citations, i.e. citations to the direct citing papers. Second generation citations have also been used for identifying under-cited papers, i.e. papers receiving much fewer citations than the important highly cited follow-ups triggered by the paper [7]. Other efforts further exploit the citation network by incorporating multiple or all citation generations, i.e. the whole path of citations from the most recent publication to the seminal one. This includes PageRank-style ranking.

In previous works [17, 18], we have proposed efficient algorithms for evaluating *hybrid* tree pattern queries on large graphs. These focus on homomorphisms and, unlike most existing approaches [1–3, 11, 19–21], allow to seamlessly handle both edge-to-edge and edge-to-path mappings between the query pattern and the graph. Thus, hybrid patterns are able to extract both direct and indirect relationships in citation networks. Consequently, they can identify different citation paths in addition to different types of direct citations, effectively capturing the flow of knowledge in the network.

Moreover, the nodes in citation networks, which represent publications, usually include additional attributes, such as *authors*, *title*, *keywords*, *venue*, *publication date*, etc. The presence of these attributes characterizes these networks as attributed graphs. Hybrid pattern queries allow nodes annotated with different types of conditions on node attributes. Figure 1 shows some simple example patterns. Single line edges indicate child edges (direct citations), while double line edges indicate descendant edges (indirect or direct citations—citation paths). Conditions on the node attributes are specified by the nodes. Hybrid patterns annotated with conditions on node attributes can also be used to explore the impact of events beyond publications, e.g., an entire venue, the entire work of

an author, or the entire collection of publications over a period of time. Thus, by allowing hybrid patterns, our approach provides higher flexibility to users exploring a citation graph, since it is possible with a single query to obtain sets of results with higher coverage and diversity.

However, since citation networks can be very large, a challenge that arises is the efficiency and scalability of evaluating hybrid patterns. To overcome this, our method relies on several optimizations. In particular, conditions on node attributes can be efficiently evaluated using node inverted lists on atomic conditions. These can be implemented using bitmap indexes, which can also be used to flexibly evaluate more complex conditions by applying bitwise operations. Since hybrid patterns may comprise edges that map to paths in the citation network, their evaluation requires reachability indexes in addition to adjacency lists. To address this, we use a novel simulation-based technique for identifying and excluding nodes of the citation network which do not participate in the final answer. Our approach computes a binary match simulation relation to identify nodes occurring in the final query answer without generating any redundant nodes. This technique is employed by our holistic graph pattern matching algorithm, called GraphMatch-Sim, to match the query pattern against the input data as a whole. Unlike most of the existing graph matching algorithms that allow edge-to-path mapping [3, 11], GraphMatch-Sim is not tied to a specific reachability indexing scheme, thus being more generic and flexible. To evaluate the performance and scalability of our method, we have conducted experiments on large citation networks. We have also shown that it significantly outperforms TwigStackD [2, 20], a well-known previous graph matching algorithm.

The rest of the paper is structured as follows. Section 2 introduces preliminary concepts. Section 3 presents the graph simulation relation and the pattern matching algorithm that we use to match hybrid tree patterns to citation graphs. Section 4 presents our experimental evaluation. Section 5 concludes the paper.

2 Preliminaries

Given a list of attributes A_1, \dots, A_n , a *citation graph* $G = (V, E, f)$ comprises a set of nodes (publications) V , a set of edges (citations) E , and a function f that assigns to every $v \in V$ a tuple $A_1 = a_1, \dots, A_n = a_n$. A node u is said to *reach* node v , denoted $u \prec v$, if there exists an edge or a path from u to v . Abusing tree notation, we call v a *child* of u (and u a *parent* of v) if $(u, v) \in E$, while v a *descendant* of u (and u an *ancestor* of v) if $u \prec v$. We focus on *tree pattern queries*. Every node x in a pattern Q is associated with a Boolean expression on the attributes of the node. There can be two types of edges in Q . A *child* (resp. *descendant*) edge denotes a child (resp. descendant) relationship between the respective two nodes.

Given a tree pattern Q and a citation graph G , a *homomorphism* from Q to G is a function m mapping the nodes of Q to nodes of G , such that: (1) for any node $x \in Q$, the tuple associated with $m(x)$ satisfies the Boolean expression associated with x ; and (2) for any edge $(x, y) \in Q$, if (x, y) is a child edge, $(m(x), m(y))$ is an

edge of G , while if (x, y) is a descendant edge, $m(x) \prec m(y)$ in G . An *occurrence* of a tree pattern Q on a data graph G is a tuple indexed by the nodes of Q whose values are the images of the nodes in Q under a homomorphism from Q to G . The *answer* of Q on G is a relation whose schema is the set of nodes of Q , and whose instance is the set of occurrences of Q under all possible homomorphisms from Q to G . If x is a node in Q , the *occurrence list of x on G* is the list of nodes in the citation graph G that satisfy the expression associated with x . Our goal is to efficiently find the answer of Q on G . To simplify the presentation, below, we assume that citation graph nodes and pattern nodes are labeled by a single label instead of being annotated by attribute-value pairs and Boolean expressions.

To summarize the answer of a query, which can have many occurrences, we use the concept of *answer graph* [17, 18]. This compactly encodes all possible homomorphisms of a query in a graph. The answer graph G_A of a pattern query Q is a k -partite graph. It has an independent node set for every node $q \in V_Q$ which is equal to the occurrence list L_q of q . There is an edge (v_x, v_y) in G_A between a node $v_x \in L_x$ and a node $v_y \in L_y$ if and only if there is an edge $(x, y) \in E_Q$ and a homomorphism from Q to G which maps x to v_x and y to v_y .

The answer graph G_A losslessly summarizes all the occurrences of Q on G . It exploits computation sharing to reduce redundancy in the representation and computation of query results.

3 The Simulation-Based Graph Matching Algorithm

3.1 Graph Simulation

Before presenting our pattern matching approach, we introduce graph simulation. Simulations have been implemented in different graph database tasks [4, 9, 12, 13]. As opposed to a homomorphism, which is a function, a simulation is a binary relation on the node sets of two directed graphs. It provides one possible notion of structural equivalence between the nodes of two graphs. Since the structure of a node is determined by its incoming and outgoing paths, we define a type of simulation called *double simulation*, which handles both the incoming and the outgoing paths of the graph nodes. Double simulation is an extension of dual simulation [12] to allow edge-to-path mappings. In this paper, we consider the double simulation of hybrid tree pattern queries on citation graph data.

Given a query Q and a directed data graph G , let (q_i, q_j) be a query edge in Q , and v_i and v_j be two nodes in G , such that $label(q_i) = label(v_i)$ and $label(q_j) = label(v_j)$. The pair (v_i, v_j) is called an *occurrence* of the query edge (q_i, q_j) if: (a) (q_i, q_j) is a child edge in Q and (v_i, v_j) is an edge in G , or (b) (q, q_i) is a descendant edge in Q and $v_i \prec v_j$ in G .

Definition 1 (Double Simulation). *The double simulation of a query $Q = (V_Q, E_Q)$ by a directed data graph $G = (V_G, E_G)$ is the largest binary relation $S \subseteq V_Q \times V_G$ such that, whenever $(q, v) \in S$, the following conditions hold:*

1. $label(q) = label(v)$.
2. For each $(q, q') \in E_Q$, there exists $v' \in V_G$ such that $(q', v') \in S$ and (v, v') is an occurrence of the edge (q, q') .
3. If $q \neq root(Q)$, let q' be the parent of q in Q ; then there exists $v' \in V_G$ such that $(q', v') \in S$, and (v', v) is an occurrence of the edge (q', q) in E_Q .

The double simulation of Q by G is unique, since there is exactly one largest binary relation S satisfying the above three conditions. This can be proved by the fact that, whenever we have two binary relations S_1 and S_2 satisfying the three conditions between Q and G , their union $S_1 \cup S_2$ also satisfies those conditions.

We call the largest binary relation that satisfies conditions 1 and 2 *forward simulation* of Q by G , and we call the largest binary relation that satisfies conditions 1 and 3 *backward simulation*. We denote the forward, backward, and double simulation by \mathcal{F} , \mathcal{B} , and \mathcal{FB} , respectively. For $q \in V_Q$, $\mathcal{F}(q)$, $\mathcal{B}(q)$, and $\mathcal{FB}(q)$ denote the set of all nodes of V_G that forward, backward, and double simulate q , respectively. \mathcal{FB} preserves both incoming and outgoing edge types (child or descendant) between Q and G , whereas \mathcal{F} and \mathcal{B} preserve only outgoing and incoming edge types, respectively. Given a tree pattern query Q and a data graph G , there exists a homomorphism from Q to G that maps node $q \in V_Q$ to node $v \in V_G$ if and only if $v \in \mathcal{FB}(q)$. This result shows the significance of double simulation in graph pattern matching. All graph data nodes captured by the double simulation participate in the query's final answer.

Computing the double simulation is the algorithmic basis of our query evaluation approach. $\mathcal{FB}(q)$ cannot be computed by simply intersecting $\mathcal{F}(q)$ and $\mathcal{B}(q)$. Thus, we develop a 2-pass algorithm called *FBSim* to compute \mathcal{FB} for query Q and data graph G by traversing the nodes of Q twice. *FBSim* leverages the acyclic nature of the rooted tree pattern. It first computes the forward simulation \mathcal{F} , considering outgoing edges, and then refines \mathcal{F} by computing a subset of the backward simulation \mathcal{B} , considering the incoming edge.

3.2 The Simulation-Based Algorithm

We now present *GraphMatch-Sim*, our holistic pattern matching algorithm that builds the answer graph for the given pattern query in two phases. First, *GraphMatch-Sim* computes the double simulation relation to find all the answer graph nodes, filtering out those that do not participate in the final answer. Then, it links the answer graph nodes with edges to construct the final answer graph.

The outline of *GraphMatch-Sim* is presented in Algorithm 1. It takes as input a data graph G and a pattern query Q . It constructs the answer graph G_A of Q on G in two phases: (a) the *node selection* phase, and (b) the *node linking* phase. The first computes the double simulation relation of Q by G using algorithm *FB-Sim* (line 1). As shown in the previous section, after relation \mathcal{FB} is computed, the occurrence list L_q for each node q of Q is available (line 2). The node linking phase traverses Q in a top-down manner and links nodes in occurrence lists L_q with edges to produce the answer graph G_A (lines 3–5). This is implemented by procedure *expand*. Let q be the current query node under consideration. For each

node $v_q \in L_q$, it expands G_A by adding incident edges to v_q . More concretely, it iterates over every child q_i of q (line 1). For each node $v_{q_i} \in L_{q_i}$ of q_i , it determines whether (v_q, v_{q_i}) is an occurrence of the query edge (q, q_i) (line 3). If so, it adds the edge (v_q, v_{q_i}) to G_A (line 4). The top-down computation phase in *FBSim* and the node linking phase in *GraphMatch-Sim* can be combined into a single top-down process (details are omitted due to space limitations).

Algorithm 1. Algorithm *GraphMatch-Sim*.

Input: Data graph G , pattern query Q

Output: Answer graph G_A of Q on G

1. Use Algorithm *FBSim* to compute \mathcal{FB} of Q by G ;
2. Initialize G_A as a k -partite graph without edges having one data node which contains the occurrence list $L_q := \mathcal{FB}(q)$ for every node $q \in V_Q$;
3. **for** ($q \in V_Q$ in a top-down order) **do**
4. **for** ($v_q \in L_q$) **do**
5. expand(q, v_q);

Procedure expand(q, v_q)

1. **for** ($q_i \in \text{children}(q)$) **do**
 2. **for** ($v_{q_i} \in L_{q_i}$) **do**
 3. **if** ((v_q, v_{q_i}) is an occurrence of edge $(q, q_i) \in E_Q$) **then**
 4. Add the edge (v_q, v_{q_i}) to G_A ;
-

4 Experimental Evaluation

Next, we study the performance of our graph pattern matching approach in terms of execution time, memory usage and scalability, and we compare it with the well-known graph pattern matching algorithm *TwigStackD* [20].

Algorithms in Comparison. In our experimental evaluation, we used different variants of our simulation-based approach *GraphMatch-Sim* (*GM-sim*), and of the algorithm *TwigStackD* (*TS*) for comparison purposes. For *GM-sim*, we implemented two variants: (1) *GM-sim*, the version of *GraphMatch-Sim* described in Sect. 3.2, and (2) *GM-sim-flt*, the *GM-sim* combined with the node pre-filtering step of [20]. For *TS*, we implemented the following four variants: (1) *TS-post*, a version of *TS* that evaluates hybrid patterns in a postprocessing step. It first finds all the solutions for the input pattern regarded as a descendant-only pattern, and then filters out solutions violating the child edge constraints. *TS-post* returns query solution tuples (pattern occurrences) and does so using merge-join operations over query path solutions. (2) *TS-ag*, an extension of *TS* which evaluates hybrid patterns directly, and, unlike *TS-post*, produces answer graphs. (3) *TS-ag-flt*, the version of *TS-ag* with the node pre-filtering optimization of [20].

(4) *TS-sspi-ftt*, a version of *TS-ag-ftt* that uses the original *Surrogate Surplus Predecessor Index* (SSPI) index scheme [2].

Experimental Setting. To efficiently check the reachability relationship between two given nodes in a graph, every graph pattern matching algorithm uses some kind of reachability indexing scheme. Most schemes associate with every node a label which is an entry in the index for the data graph. With the exception of *TS-sspi-ftt*, all the other graph matching algorithms we implemented used a recent effective reachability scheme called *Bloom Filter Labeling* (BFL) [16].

Given two nodes u and v in a data graph, in order to efficiently check if u can be reached from v , every graph pattern matching algorithm uses some kind of reachability indexing scheme. Most graph reachability indexing schemes associate with every graph node a label which is an entry in the index for the data graph. Our approach uses a recent effective reachability scheme called *Bloom Filter Labeling* (BFL) [16]. The essence of the BFL approach lies in using the Bloom filter technique to summarize large data graphs into compact data structures while preserving important properties of the data. By converting node reachability checking into set containment testing implemented using bitwise operations, BFL greatly outperforms most of existing approaches in the index construction time, index size as well as nodes reachability checking time [16].

Our implementation was coded in Java. All the experiments reported here were performed on a workstation having an Intel Xeon CPU 1240V5@3.50 GHz processor with 32 GB memory.

Table 1. Dataset statistics. $|V|$, $|E|$ and $|L|$ are the number of nodes, edges and distinct labels, respectively. $maxout$ and $maxin$ are the maximum out-degree and in-degree of the graph. d_{avg} denotes the average degree of a graph.

Dataset	$ V $	$ E $	$ L $	$maxout$	$maxin$	d_{avg}
citation	1,397,240	3,016,539	16,442	717	4,090	4.32
cite-lb10000	6,540,401	15,011,260	8,343	181,247	203,695	4.59
cite-lb7000	6,540,401	15,011,260	5,662	181,247	203,695	4.59
cite-lb5000	6,540,401	15,011,260	3,970	181,247	203,695	4.59
cite-lb3000	6,540,401	15,011,260	2,434	181,247	203,695	4.59
cite-lb1000	6,540,401	15,011,260	885	181,247	203,695	4.59

Datasets. We ran experiments on two citation graphs with different structural properties. Their main characteristics are summarized in Table 1.

*citation*¹ models citations extracted from DBLP, ACM, MAG (Microsoft Academic Graph) publications, and consists of a directed graph with approximately 1.4M nodes (publications) and 3M edges (citations).

¹ www.aminer.cn/citation.

Table 2. Parameters for query generation.

Parameters	Range	Description
Q	300 to 3300	Number of queries
D	6 to 16	Maximum depth of queries
DS	0 to 1	Probability of setting an edge to be a descendant edge ('//')
NP	1 to 3	Number of branches per query node

*citeseerx*² represents a directed graph consisting of approximately 6.5M publications and 15M citations. Since the original graph does not contain labels, we synthetically added a number of distinct labels to graph nodes, following a Gaussian distribution. Using this process, we generated ten labeled citeseerx graphs whose number of labels ranges from 1K to 10K. Each graph is named as cite-lb x , where x is the number of labels in the graph. Table 1 shows statistics for five of these datasets.

Table 3. Query set statistics on *citeseerx*. ‘//’ denotes the descendant pattern edge. SIM% and FLT% are the percentage of the number of inverted list nodes retained by the simulation-based filtering and the traversal-based filtering (pre-filtering), respectively. #TUP denotes the query solution tuples.

Query set	#queries	V	Height	% of ‘//’	Maxout	SIM%	FLT%	FLT/SIM	Avg. #TUP
cite-lb10000.qry	10	3.5	1.4	65.71	1.9	1.08	3.97	3.67	219.9
cite-lb7000.qry	10	3.1	1.1	67.74	2	1.57	5.29	3.36	1,859.1
cite-lb5000.qry	10	3.2	1.1	65.63	2	1.50	5.87	3.91	582.9
cite-lb3000.qry	10	3.4	1.4	67.64	2	1.57	5.24	3.35	17,543.3
cite-lb1000.qry	10	3.3	1.2	63.64	2	1.46	6.57	4.50	40,064.1

Table 4. Query statistics over *citation*.

Query	V	Height	% of ‘//’	Maxout	SIM%	FLT%	FLT/SIM	#TUP
0	8	2	37.50	3	0.007	3.90	557.51	10
1	7	2	28.57	3	0.09	3.82	42.40	930
2	17	6	35.29	3	0.007	0.02	2.93	108
3	16	6	50.00	2	0.33	0.75	2.31	3.52548E+11
4	11	3	36.36	3	1.22	2.33	1.91	255,816
5	22	7	22.73	3	1.23	3.23	2.61	6.88239E+12
6	13	4	53.85	4	2.14	2.57	1.20	88,051
7	13	4	38.46	3	5.02	6.23	1.24	11,784,410,550
8	13	5	46.15	2	2.92	3.13	1.07	4.96216E+11
9	18	5	50.00	3	3.22	5.99	1.86	1.35026E+17

² citeseerx.ist.psu.edu.

Queries. We implemented a query generator that creates tree pattern queries based on the parameters listed in Table 2. For each data graph, we first generated a number of queries (in the range of 300 to 3300) using different value combinations of these parameters. Then, we formed a query set by randomly selecting 10 of them. Table 3 summarizes the statistics of the query sets on *citeseerx*. The detailed statistics of the ten queries used on *citation* are shown in Table 4.

Performance Comparison on *citation* Graph. We measured the performance of the different versions of *GM-sim*, *GM-bup* and *TS* for evaluating the ten queries (Table 4) over *citation*. In Table 5, we report the normalized query time of the algorithms (the timing of *GM-sim-opt* has been normalized to 1 for each query). The average query evaluation time and memory usage of the compared algorithms is shown in Table 6. Table 4 shows for each query the percentage of the number of inverted list nodes accessed by the algorithms using simulation-based technique (SIM%) and the node pre-filtering technique (FLT%) during the matching process. The three algorithms *GM-bup*, *TS-ag*, and *TS-post* access the entire inverted lists during the matching process since they do not have a filtering phase. We next discuss our findings.

Table 5. Normalized query evaluation time on *citation*.

Query	<i>GM-sim</i>	<i>GM-sim-flt</i>	<i>TS-ag</i>	<i>TS-ag-flt</i>	<i>TS-post</i>	<i>TS-sspi-flt</i>
0	1	8.17	5.96	7.74	2210.22	6.39
1	1	4.11	238.84	4.25	1260.34	120.20
2	1	1.45	822.73	1.34	983.15	1.09
3	1	0.90	862.73	7.69	na	161.52
4	1	0.76	17.20	0.85	56.80	34.54
5	1	0.74	289.94	24.33	na	55.80
6	1	0.87	81.18	1.50	80.03	45.53
7	1	0.91	26.56	0.99	na	0.44
8	1	0.68	34.28	0.98	na	22.73
9	1	0.88	7 5.10	27.09	na	290.76

Table 6. Average query evaluation time and memory usage on *citation*.

Average	<i>GM-sim</i>	<i>GM-sim-flt</i>	<i>TS-ag</i>	<i>TS-ag-flt</i>	<i>TS-post</i>	<i>TS-sspi-flt</i>
Time (sec.)	23.991	20.229	1962.846	352.864	na	3706.509
Memory (MB)	2586	3024	1832	2317	na	5834

Performance of Graph Matching. The time performance of *GM-sim* variants greatly outperforms the *TS* variants. This shows the effectiveness of the node pruning strategy with the graph simulation technique. In particular, *GM-sim*

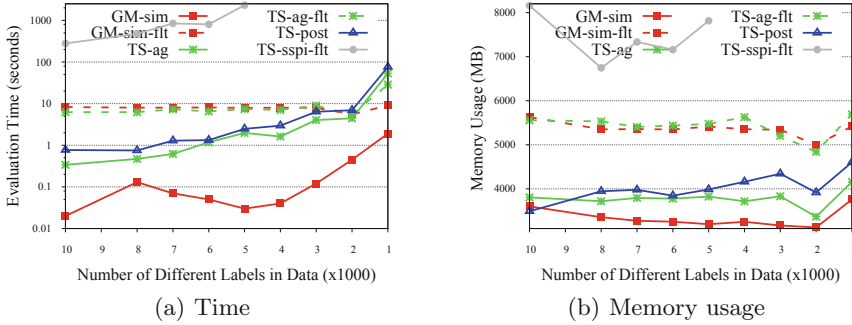


Fig. 2. Scalability comparison on the citeseerx graph.

has the best performance overall among all the algorithms in comparison. As shown in Table 5, the max speedup of *GM-sim* over both *TS-post* and *TS-ag* is around $2210\times$ and $862\times$, respectively.

TS-post has the worst performance among all the algorithms. This is due to the postprocessing strategy in *TS-post* for handling the child constraints in hybrid queries, which results in a large amount of intermediate results which do not participate in the query answer. Furthermore, *TS-post* is unable to evaluate queries having a large number of solution tuples (see column #TUP in Table 4): due to the large number of intermediate results produced by merge-join operations, *TS-post* was aborted with out-of-memory error, before returning solution tuples.

The pre-filtering technique can significantly improve the time performance of graph matching algorithms. For example, *TS-ag-flt* has a max and an average speedup of $612\times$ and $5\times$ over *TS-ag*, respectively (Table 6). This can be explained by the fact that *TS-ag-flt* accesses a small percentage of the inverted list nodes accessed by *TS-ag* (see Column FLT% in Table 4). Nevertheless, the improvement is only marginal for *GM-sim*. On Q_0 and Q_1 , *GM-sim-flt*, in fact, increases the evaluation time of *GM-sim* by over $7\times$ and $3\times$, respectively. This is due to the fact that the overhead cost associated with node pre-filtering offsets the reduction in the query execution cost.

From Table 4, we see that both the simulation-based pruning and the pre-filtering can drastically reduce the accessed inverted list nodes. We also observe that, from column FLT/SIM in Table 4, the former has a better pruning power than the latter, since the simulation-based pruning retains only nodes in the final query answer. The pre-filtering process increases the memory usage of the algorithms as well. The increase of *GM-sim-flt* over *GM-sim*, *TS-ag-flt* over *TS-ag*, is 60% and 26%, respectively.

Scalability Comparison on *citeseerx*. In this experiment, we examine the impact of the total number of distinct graph labels on the performance of the algorithms in comparison. We used the aforementioned ten labeled *citeseerx* graphs *cite-lbx* (Table 1), increasing the number of labels x from 1K to 10K. For

each *cite-lbx* graph, we generated a query set *cite-lbx.qry* with 10 distinct queries (Table 3).

Figure 2 reports the execution time and memory usage of the six algorithms. Again, *GM-sim* has the best overall performance. However, *TS-sspi-flt* exhibits the worst time and memory performance here. It is unable to finish after 10 h when the number of labels decreases below 5K. *TS-post* performs better than *TS-sspi-flt*, since the average number of result tuples per query is relatively small on *citeseerrx*. As shown in Fig. 2(a), the execution time of the algorithms tends to increase while decreasing the total number of labels. This is reasonable since the average cardinality ($|V|/|L|$) of the input inverted list per node label in a graph increases when the number of distinct labels in the graph decreases. This observation shows dependency of the execution time on the input size.

In contrast to the results on *citation*, in *citeseerrx* we observe a degenerate time performance of the algorithms when the pre-filtering technique is applied to graphs with a large (>1000) number of labels. The reason is that when the number of graph labels is large, the cardinality of the inverted lists is relatively small. In this case, the potential benefit of reducing intermediate results during graph matching can be offset by the overhead of node pre-filtering.

Finally, the memory performance of the algorithms on *citeseerrx* is consistent with the results on *citation* (Fig. 2(b)).

5 Conclusion

Citation graphs are typically employed to measure the scientific impact of publications. Current needs in assessing the impact of publications go beyond assigning a score and require an analysis and exploration of citation graphs. This involves extracting both direct citations (*edges*) and indirect scientific influence (*paths*). To this end, we proposed hybrid patterns to query citation networks, thus capturing both direct and indirect relationships between publications, and we addressed the problem of efficiently evaluating hybrid tree pattern queries over citation graphs. We employed a simulation-based holistic graph matching algorithm that proactively prunes all the nodes not appearing in the final answer. Our experimental evaluation showed that our algorithm outperforms a well-known pattern matching algorithm in terms of query execution time, scalability, and memory consumption, demonstrating the feasibility of our approach for exploring and analyzing large-scale citation graphs.

References

1. Aberger, C.R., Tu, S., Olukotun, K., Ré, C.: Emptyheaded: a relational engine for graph processing. In: SIGMOD, pp. 431–446 (2016)
2. Chen, L., Gupta, A., Kurul, M.E.: Stack-based algorithms for pattern matching on DAGs. In: VLDB, pp. 493–504 (2005)
3. Cheng, J., Yu, J.X., Yu, P.S.: Graph pattern matching: a join/semijoin approach. IEEE Trans. Knowl. Data Eng. **23**(7), 1006–1021 (2011)

4. Fan, W., Li, J., Ma, S., Tang, N., Wu, Y., Wu, Y.: Graph pattern matching: From intractable to polynomial time. *PVLDB* **3**(1), 264–275 (2010)
5. Fragkiadaki, E., Evangelidis, G.: Review of the indirect citations paradigm: theory and practice of the assessment of papers, authors and journals. *Scientometrics* **99**(2), 261–288 (2013). <https://doi.org/10.1007/s11192-013-1175-5>
6. Fragkiadaki, E., Evangelidis, G.: Three novel indirect indicators for the assessment of papers and authors based on generations of citations. *Scientometrics* **106**(2), 657–694 (2016)
7. Hu, X., Rousseau, R.: Scientific influence is not always visible: the phenomenon of under-cited influential publications. *J. Informetr.* **10**(4), 1079–1091 (2016)
8. Jiang, X., Zhuge, H.: Forward search path count as an alternative indirect citation impact indicator. *J. Informetr.* **13**(4), 100977 (2019)
9. Kaushik, R., Bohannon, P., Naughton, J.F., Korth, H.F.: Covering indexes for branching path queries. In: *SIGMOD*, pp. 133–144 (2002)
10. Kosmulski, M.: Hirsch-type approach to the 2nd generation citations. *J. Informetr.* **4**(3), 257–264 (2010)
11. Liang, R., Zhuge, H., Jiang, X., Zeng, Q., He, X.: Scaling hop-based reachability indexing for fast graph pattern query processing. *IEEE Trans. Knowl. Data Eng.* **26**(11), 2803–2817 (2014)
12. Ma, S., Cao, Y., Fan, W., Huai, J., Wo, T.: Strong simulation: capturing topology in graph pattern matching. *ACM Trans. Database Syst.* **39**(1), 4:1–4:46 (2014)
13. Mennicke, S., Kalo, J., Nagel, D., Kroll, H., Balke, W.: Fast dual simulation processing of graph database queries. In: *ICDE*, pp. 244–255 (2019)
14. Renoust, B., Claver, V., Baffier, J.: Multiplex flows in citation networks. *Appl. Netw. Sci.* **2**, 23 (2017)
15. Schubert, A.: Using the h-index for assessing single publications. *Scientometrics* **78**(3), 559–565 (2009)
16. Su, J., Zhu, Q., Wei, H., Yu, J.X.: Reachability querying: can it be even faster? *IEEE Trans. Knowl. Data Eng.* **29**(3), 683–697 (2017)
17. Wu, X., Theodoratos, D., Skoutas, D., Lan, M.: Efficiently computing homomorphic matches of hybrid pattern queries on large graphs. In: Ordonez, C., Song, I.-Y., Anderst-Kotsis, G., Tjoa, A.M., Khalil, I. (eds.) *DaWaK 2019*. LNCS, vol. 11708, pp. 279–295. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-27520-4_20
18. Wu, X., Theodoratos, D., Skoutas, D., Lan, M.: Evaluating mixed patterns on large data graphs using bitmap views. In: Li, G., Yang, J., Gama, J., Natwichai, J., Tong, Y. (eds.) *DASFAA 2019*. LNCS, vol. 11446, pp. 553–570. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-18576-3_33
19. Zeng, Q., Jiang, X., Zhuge, H.: Adding logical operators to tree pattern queries on graph-structured data. *PVLDB* **5**(8), 728–739 (2012)
20. Zeng, Q., Zhuge, H.: Comments on “stack-based algorithms for pattern matching on dags”. *PVLDB* **5**(7), 668–679 (2012)
21. Zervakis, L., Setty, V., Tryfonopoulos, C., Hose, K.: Efficient continuous multi-query processing over graph streams. In: *EDBT*, pp. 13–24 (2020)



ArtSim: Improved Estimation of Current Impact for Recent Articles

Serafeim Chatzopoulos^{1,2}(✉), Thanasis Vergoulis², Ilias Kanellos², Theodore Dalamagas², and Christos Tryfonopoulos¹

¹ Department of Informatics and Telecommunications,
University of the Peloponnese, 22100 Tripoli, Greece
{schatzop,trifon}@uop.gr

² IMSI, “Athena” Research Center, 15125 Athens, Greece
{vergoulis,ilias.kanellos,dalamag}@athenarc.gr

Abstract. As the number of published scientific papers continuously increases, the need to assess paper impact becomes more valuable than ever. In this work, we focus on citation-based measures that try to estimate the popularity (current impact) of an article. State-of-the-art methods in this category calculate estimates of popularity based on paper citation data. However, with respect to recent publications, only limited data of this type are available, rendering these measures prone to inaccuracies. In this work, we present **ArtSim**, an approach that exploits paper similarity, calculated using scholarly knowledge graphs, to better estimate paper popularity for recently published papers. Our approach is designed to be applied on top of existing popularity measures, to improve their accuracy. We apply **ArtSim** on top of four well-known popularity measures and demonstrate through experiments its potential in improving their popularity estimates.

Keywords: Scientific impact assessment · Scholarly knowledge graphs

1 Introduction

With the growth rate of scientific articles continuously increasing [8], the reliable assessment of their scientific impact is now more valuable than ever. As a result, a variety of *impact measures* have been proposed in the literature, aiming to quantify scientific impact at the article level. Such measures have various practical applications: for instance, they can be used to rank the results of keyword-based searches (e.g., [18]), facilitating literature exploration and reading prioritisation, or to compare and monitor the impact of different articles, research projects, institutions, or researchers.

Since scientific impact can be defined in many, distinct ways [3], the proposed measures vary in terms of the approach they follow (e.g., citation-based, altmetrics), as well as in the aspect of scientific impact they attempt to capture

(e.g., impact in academia, social media attention). In this work, we focus on citation-based measures, that attempt to estimate the current impact of each article, i.e., its current *popularity* or hype. Providing accurate estimations of article popularity is an open problem, as has been shown by a recent extensive experimental evaluation [6]. Furthermore, popularity distinctly differs from the overall (long-term) impact of an article that is usually captured by traditional citation-based measures (e.g., citation count).

One important issue in estimating article popularity is to provide accurate estimations for the most recently published articles. The estimations of most popularity measures rely on the existing citation history of each article. However, since very limited citation history data are available for recent articles, their impact estimation based on these data is prone to inaccuracies. Hence, these measures fail to provide correct estimations for recent articles. To alleviate this issue, in this work we introduce **ArtSim**, a new approach to assess article popularity. Our approach does not only rely on each article’s historical data, but also considers the history of older, similar papers, for which these data are more complete. The intuition behind our method is that similar papers are likely to follow a similar trajectory in terms of popularity. To quantify article similarity, we exploit the corresponding author lists and the involved topics¹. This information is available in *scholarly knowledge graphs*, a large variety of which has been made available in recent years (e.g., AMiner’s DBPL-based datasets [17], the Open Research Knowledge Graph [5], the OpenAIRE Research Graph [9,10].)

The real power of **ArtSim** comes from the fact that it can be applied on top of any existing popularity measure to improve its accuracy. To demonstrate this, we first apply **ArtSim** on top of the best performing popularity measures (according to [6]) to produce a set of improved measures. Then, we examine the achieved benefits (by replicating the experimental process in [6]). Our experiments indicate that **ArtSim** effectively enhances the performance of common measures in estimating article popularity.

2 Our Approach

2.1 Background

Our proposed method aims at transforming popularity scores based on any popularity measure, in order to increase the accuracy of its estimations. To achieve this, it attempts to improve the estimation for all recent articles by exploiting path-based article similarities in *scholarly knowledge graphs*.

Knowledge graphs, also known as *heterogeneous information networks* [15], are graphs that encode rich domain-specific information about various types of entities, represented as nodes, and the relationships between them, represented as edges. Figure 1 presents an example of such a knowledge graph, consisting of nodes representing papers, authors and topics (i.e., three different node types).

¹ Here we use similarity based on authors and topics as a proof of concept. However, our approach can be generalized using any other definition of article similarity.

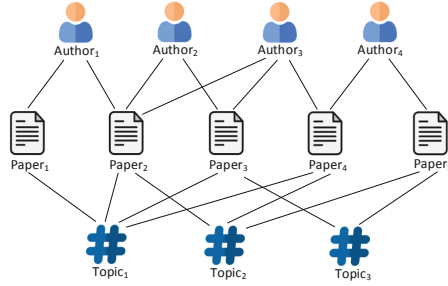


Fig. 1. A scholarly knowledge graph including papers, authors and topics.

Two types of (bidirectional) edges are present in this example network: edges between authors and papers, denoted as **Author - Paper** (AP or PA, for brevity), and edges between papers and topics, denoted as **Paper - Topic** (PT or TP). The former represent the authorship of papers, while the latter encode the information that a particular paper is written on a particular topic.

Various semantics are implicitly encoded in the paths of knowledge graphs. For example, in the graph of Fig. 1, a path from an author node to another one of the same type that involves an AP edge followed by a PT, a TP, and a PA edge relates two authors that have published works in the same topic (e.g., both *Author₁* and *Author₄* have papers about *Topic₁*). In fact, all paths that correspond to the same sequence of node and edge types encode latent, “multi-hop” relationships having the same interpretation. In the literature, such a sequence of node and edge types (e.g., the APTPA of the previous example) is known as a *metapath*. Metapaths are useful for many graph analysis and exploration applications. For example, in our approach, we use them to calculate *metapath-based similarities*: the similarity between two nodes of the same type, based on the semantics of a given metapath, can be captured by counting the number of instances of this metapath connecting these nodes (e.g., [16, 20]).

2.2 ArtSim

Our proposed method, called **ArtSim**, can be applied on top of any popularity measure to increase the accuracy of its estimations. As such, **ArtSim** takes the scores calculated by any popularity measure as input, applies transformations on them, and produces a new set of improved popularity scores. This process is presented in Fig. 2.

The transformations applied on popularity scores by **ArtSim** rely on the intuition that similar articles are expected to share similar popularity dynamics. To calculate the similarity between different papers, **ArtSim** relies on the Join-Sim [20] similarity measure calculated on PAP and PTP metapaths. Evidently, the similarity between papers is not uniquely defined, hence different metapaths encode different similarity semantics. For example, while PAP metapaths define paper paper similarity based on their common authors, PTP metapaths define paper

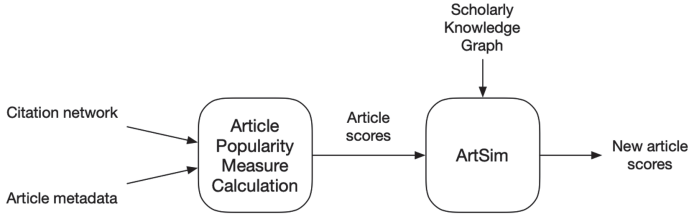


Fig. 2. Our proposed approach.

similarity based on their common topics. **ArtSim** uses the calculated similarity scores to provide improved popularity estimates (scores), focusing in particular on recent papers that have a limited citation history. The calculation of **ArtSim** scores is based on the following formula:

$$S(p) = \begin{cases} \alpha * S_{PAP}(p) + \beta * S_{PTP}(p) + \gamma * S_{initial}(p), & p.year \geq t_c - y \\ S_{initial}(p), & \text{otherwise} \end{cases}$$

where S_{PAP} and S_{PTP} are the average popularity scores of all the articles that are similar to p , based on metapaths PAP and PTP respectively. $S_{initial}$ is the popularity score of paper p , based on the original popularity measure and t_c denotes the current time. Finally, our method applies transformations on popularity scores for those papers published in years, which range in the time span $[t_c - y, t_c]$, where $y \geq 0$.

Note, that parameters $\alpha, \beta, \gamma \in [0, 1]$. Furthermore, we set α, β, γ so that $\alpha + \beta + \gamma = 1$. Varying these parameters in the range $[0 - 1]$ has the following effects: as α increases, article similarity is mostly calculated based on common authors. As β increases, article similarity depends mainly on common topics. Finally, as γ approaches 1 the popularity scores remain identical to those calculated by the initial popularity measure.

3 Evaluation

In this section, we discuss the experiments conducted to assess the effectiveness of our method. Section 3.1 discusses the experimental setup of our evaluation approach i.e., the datasets, methodology and popularity measures used, and Sect. 3.2 showcases the improvements that **ArtSim** brings to popularity measures.

3.1 Setup

Datasets. For our experiments, we used the following datasets:

- *DBLP Scholarly Knowledge Graph (DSKG) dataset.* It contains data for 3,079,008 papers, 1,766,548 authors, 5,079 venues and 3,294 topics from DBLP. It is based on AMiner’s citation network dataset [17] enriched with topics from the CSO Ontology [13] using the CSO Classifier [12] on paper abstracts.

- *DBLP Article Similarities (DBLP-ArtSim) dataset*. It contains similarities among articles in the previous network based on different metapaths. In particular, we calculated paper similarities based on their author list using meta-path **Paper** – **Author** – **Paper** and on common topics, captured by meta-path **Paper** – **Topic** – **Paper**. This dataset is openly available on Zenodo² under CC BY 4.0 license.

Methodology. To assess paper popularity we follow the experimental framework proposed in [6], which is based on the evaluation of total orderings (rankings) of papers based on their short-term future citations. As explained in the referenced work, the number of citations a paper receives in the near future, is a good a-posteriori indicator of its current popularity. Thus, the aforementioned rankings can be used as a ground truth for experiments to evaluate the effectiveness of measures in ordering papers based on their popularity. This approach is also suitable for our needs, since an overall ordering of papers can be used as a basis for the comparison of any pair of papers based on their relative impact.

Based on the above, we define a split time t_s that splits our dataset in half, into two equally sized sets. The first half, denoted as $S(t_s)$ contains papers published before t_s and is considered known to the examined popularity measures. We also consider a future state of the network in the time $t_s + \tau$ which we use to construct the ground truth. In our case, set $S(t_s + \tau)$ contains 30% more articles than $S(t_s)$. We finally rank each paper in the future state of the network based on the number of its citations (i.e., the citations it received in the time span from t_s to $t_s + \tau$). This ranked list acts as the ground truth and is used to evaluate the effectiveness of popularity measures.

We measure the effectiveness of any approach compared to the ground truth using the following two measures:

- *Kendall's τ* [7], is a non-parametric measure for the similarity in the ordering of two ranked lists, based on the number of concordantly ordered pairs of items between them. Its values range from -1 to 1 , with 1 denoting perfect agreement, -1 denoting perfect inversion, and 0 denoting no correlation.
- *Normalised Discounted Cumulative Gain at rank k (nDCG@ k)* is a measure of ranking quality using the graded relevance scale of documents in the ranking list. It is a normalized version of the *Discounted Cumulative Gain (DCG) at rank k* in the range $[0, 1]$. The value 1 corresponds to the ideal *DCG*, achieved when the ranking perfectly agrees with the ground truth.

We use Kendall's τ and nDCG@ k to capture the overall, and top- k similarity of the ranked lists to the ground truth, respectively.

Popularity Measures. To evaluate our method, we chose the four overall best performing popularity measures in terms of correlation in the DBLP dataset for

² <https://doi.org/10.5281/zenodo.3778916>.

Table 1. Parameter configuration for each popularity measure.

Method	Configuration
ECM	$\alpha = 0.2, \gamma = 0.4$
RAM	$\gamma = 0.4$
CR	$\alpha = 0.4, \tau_{dir} = 10$
FR	$\alpha = 0.5, \beta = 0.2, \gamma = 0.3, \rho = -0.42$

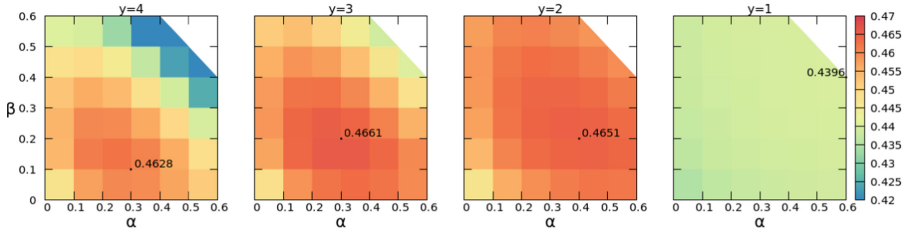
the scenario of popularity according to a recent experimental study [6]. The optimal parameter settings per measure were selected after running each one for various parameterisations and calculating the correlation of the ranked list produced by each one to the ground truth ranking. Table 1 presents the selected parameter setting per popularity measure. We briefly describe the intuition behind each measure below:

- *Retained Adjacency Matrix (RAM)* [4] estimates popularity using a time-aware adjacency matrix to capture the recency of cited papers. The parameter $\gamma \in (0, 1)$ is used as a basis of an exponential function to scale down the value of a citation link according to its age.
- *Effective Contagion Matrix (ECM)* [4] is an extension of RAM that also considers the temporal order of citation chains apart from direct links. It uses two parameters $\alpha, \gamma \in (0, 1)$ where α is used to adjust the weight of citation chains based on their length and γ is the same as in RAM.
- *CiteRank (CR)* [19] measures popularity by simulating the behaviour of researchers searching for new articles. It uses two parameters $\alpha \in (0, 1)$ and $\tau_{dir} \in (0, \infty)$ to model the traffic to a given paper. A paper is randomly selected with an exponentially discounted probability according to its age with τ_{dir} being the decay factor. Parameter α is the probability that a researcher stops its search, with $1 - \alpha$ being the probability that he continues with a reference of the paper he just read.
- *FutureRank (FR)* [14] scores are calculated combining PageRank scores with calculations on a bipartite graph with authors and papers, while also promoting recently published articles with time-based weights. It uses parameters $\alpha, \beta, \gamma \in (0, 1)$ and $\rho \in (-\infty, 0)$; α is the coefficient of the PageRank scores, β is the coefficient of the authorship scores and γ is the coefficient of time-based weights which exponentially decrease based on the exponent ρ .

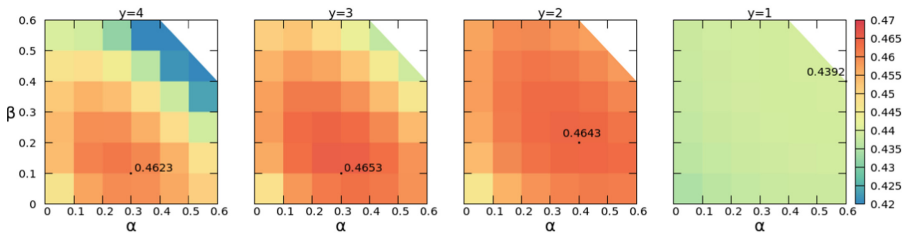
3.2 Effectiveness of Our Approach

Improvements in Correlation. In this experiment, we examine the gains of ArtSim in terms of Kendall’s τ correlation. For each examined popularity measure (ECM, RAM, CR and FR) we vary parameters α, β, γ of our method, as well as parameter y , which sets the number of past years for which we consider

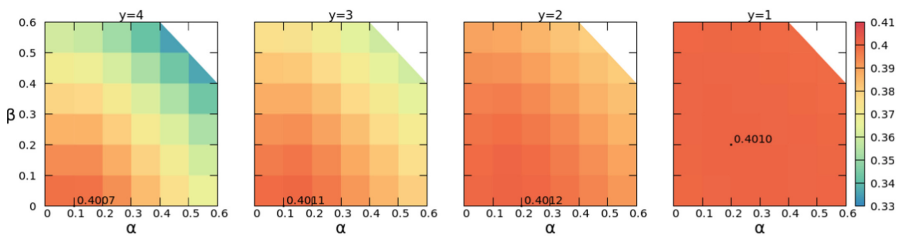
papers in the cold start phase. We visualise, in the form of heatmaps, the correlation achieved for each method for different configurations when $\alpha, \beta \in [0, 0.6]$ and $y \in [1, 4]$ (Fig. 3). Parameter γ is implied, since $\alpha + \beta + \gamma = 1$.



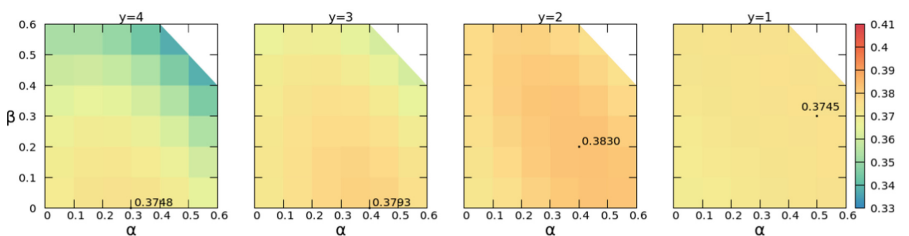
(a) ECM



(b) RAM



(c) CR



(d) FR

Fig. 3. Heatmaps depicting the effectiveness of our approach for different parameters in terms of Kendall's τ correlation for each popularity measure.

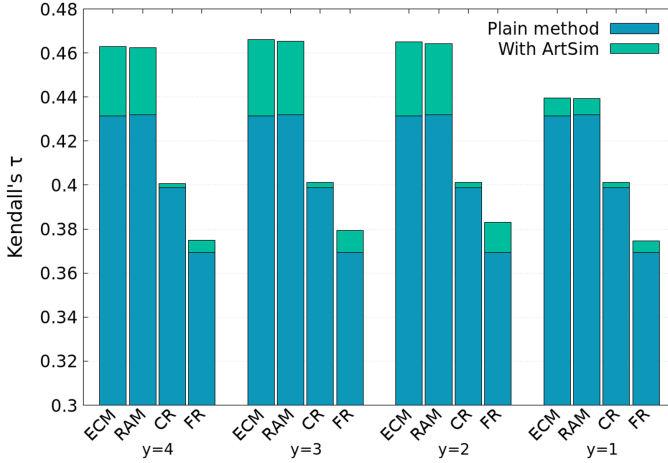


Fig. 4. Effectiveness of our approach in terms of Kendall's τ for the best parameterisation per year for each popularity measure.

In general, we observe that our approach achieves the maximum gains for $y \in \{2, 3\}$. For $y = 1$, we see that for all methods, the scores have a small deviation as expected, as our method adjusts the popularity scores of a small fraction of the overall articles, i.e., only those published in the last year. The heatmaps also validate that both scores based on similarity of authors and topics are important, since the correlation observed decreases when both parameters α and β approach zero.

Based on the experiments, **ArtSim** achieves the best correlation, $\tau = 0.4661$ using the ECM method when $\{\alpha = 0.3, \beta = 0.2, \gamma = 0.6, y = 3\}$. The best scores for **ArtSim** using the other methods are $\tau = 0.4653$ at $\{\alpha = 0.3, \beta = 0.1, \gamma = 0.7, y = 3\}$ for RAM, $\tau = 0.4012$ at $\{\alpha = 0.2, \beta = 0, \gamma = 0.8, y = 2\}$ for CR, and $\tau = 0.3830$ at $\{\alpha = 0.4, \beta = 0.2, \gamma = 0.4, y = 2\}$ for FR.

We further examined the gains of **ArtSim** in terms of Kendall's τ correlation compared to the plain popularity measures. The best parameter configuration for each method is selected for each year. The results are illustrated in Fig. 4. Overall, significant improvements in correlation are observed when **ArtSim** is applied on the ECM and RAM measures. In particular, ECM and RAM are improved by 8% for the best parameter configuration for $y \in [2, 4]$. As expected, smaller gains for all methods are achieved for $y = 1$. In that case, as previously mentioned, our approach affects the popularity score of the papers published only in the last year, affecting only a small fraction of the overall papers.

Improvements in nDCG. In this experiment, we examine the effectiveness of **ArtSim** in terms of $nDCG@k$ for all considered popularity measures compared to the ground truth. We performed two sets of experiments: (a) we measure the $nDCG@k$ achieved by **ArtSim**, varying k , and (b) we examine how **ArtSim** affects top- k results in two indicative keyword search scenarios.

Table 2. Effectiveness of our approach for $y = 3$ in terms of $nDCG@k$.

	Small values of k			Large values of k		
	5	50	500	400,000	500,000	600,000
ECM	0.8323	0.8634	0.8953	0.8780	0.8833	0.8884
ArtSim-ECM	0.8323	0.8634	0.8953	0.8837	0.8912	0.9003
RAM	0.8588	0.8521	0.8943	0.8774	0.8842	0.8881
ArtSim-RAM	0.8588	0.8521	0.8943	0.8836	0.8904	0.9008
CR	0.3530	0.5263	0.6060	0.7904	0.8149	0.8272
ArtSim-CR	0.3530	0.5263	0.6060	0.7983	0.8199	0.8307
FR	0.3403	0.5018	0.5526	0.7586	0.7934	0.8101
ArtSim-FR	0.3403	0.5018	0.5526	0.7731	0.7961	0.8152

Table 2 presents the $nDCG@k$ values, per popularity measure, both when plainly run, as well as when ArtSim is applied on them. In this experiment we select $y = 3$, which produces the best correlation according to the previously presented results. In particular, we separately examined ArtSim’s behaviour for small and for large values of parameter k . In particular we examine $nDCG@k$ for $k \in \{5, 50, 500\}$, as well as for $k \in \{400.000, 500.000, 600.000\}$. Interestingly, for small values of k , our approach performs equally to the initial popularity measures, at its best configuration. This behaviour indicates that existing state-of-the-art popularity measures accurately identify the top papers in terms of popularity. Another apparent explanation is that the set of most popular papers, at the global level, mainly includes those that already have a more extended citation history, i.e., they have become known by the scientific community and maintain their status. ArtSim’s performance gain becomes apparent for larger values of k . In relative terms, our method improves upon the $nDCG$ achieved by the popularity measures, starting at the top 7% of the most popular papers and beyond. In other words, our method does provide gains in terms of $nDCG$ for the large majority of papers, while maintaining the $nDCG$ values achieved for the overall most popular papers. Likely, these larger sets of top popular papers also include recently published ones for which the popularity estimations are improved by ArtSim. This is further supported by the observation that the $nDCG$ values achieved increase with k , i.e., the more recent papers are included, the more noticeable ArtSim’s effect.

In our second set of experiments we illustrate that the performance gains, which are observed at the global level only for large values of k , are not negligible in practical applications. For example, in a real scenario of literature exploration, academic search engine users usually refine their searches using multiple keywords and by applying filters (e.g., based on the venues of interest or the publication years). Their intention is to reduce the number of papers they have to examine, however even in this case usually at least hundreds of papers are contained in the results. Hence, effective ranking is crucial to facilitate the reading

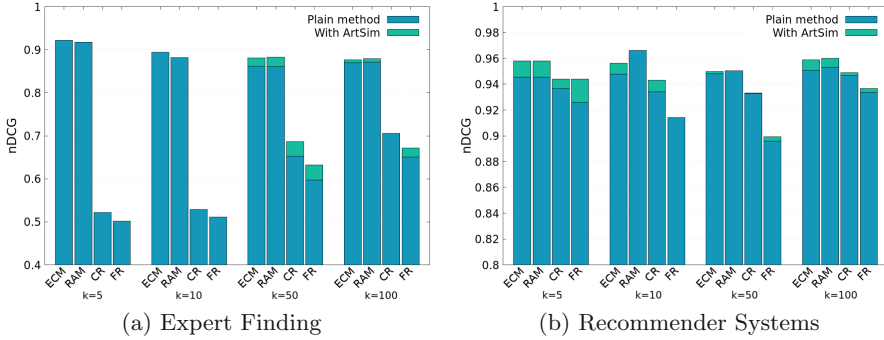


Fig. 5. Effectiveness of our approach in terms of nDCG for different keyword search scenarios with $y = 3$ and varying k .

prioritization. Furthermore, the resulting article lists usually contain only a small subset of all the articles in a dataset. We do not expect to only find the overall most popular papers in such subsets, owing to the different publication dynamics in each subdomain (i.e., different research communities have different sized and publish and/or cite at different speeds). Hence, in our second set of experiments we measure the $nDCG@k$ achieved by *ArtSim* for $k \in \{5, 10, 50, 100, 500\}$ on the results of two indicative keyword search scenarios.

In the first search scenario we used the query “expert finding”. This keyword search resulted in a set of 549 articles. Figure 5a presents the nDCG values for this search, per popularity measure, along with the gains of *ArtSim* for $y = 3$. We observe that *ArtSim* improves the nDCG values for $k = 50$ and $k = 100$. In our second scenario, we tried a conditioned query. Particularly, we used “recommender systems” as the search keywords keeping only papers published in well-known venues of data management and recommender systems, namely VLDB, SIGMOD, TKDE, ICDE, EDBT, RecSym and ICDM. The result set includes 525 articles. Figure 5b presents the nDCG results. We observe that *ArtSim* boosts nDCG scores for all measures, starting from the smallest value of $k = 5$. These results indicate that in addition to improving the overall correlation, our approach also offers improvements in the case of practical, keyword-search based queries with regards to the top returned results.

4 Related Work

There is a lot of work in the areas of bibliometrics and scientometrics to quantify the impact of scientific articles. In particular, much focus has been put on quantifying current or recent impact of scientific publications [4, 14, 19], in contrast to the overall impact traditionally estimated by bibliometric measures, such as the citation counts. In depth examinations of various impact measures that have been proposed in the literature can be found in [1, 6]. In contrast to the above, our own approach does not aim to introduce a new popularity measure,

but rather aims at improving the accuracy of existing ones. To the best of our knowledge, this is the first approach of this type to be introduced.

Our approach is built upon recent work on entity similarity in the area of heterogeneous information networks. Some of the first entity similarity approaches for such networks (e.g., PopRank [11] and ObjectRank [2]) are based on random walks. Later works, like PathSim [16], focus on providing more meaningful results by calculating node similarity measures based on user-defined semantics. Our own work is based on JoinSim [20], which is more efficient compared to PathSim, making it more suitable for analyses on large scale networks.

5 Conclusions

We presented **ArtSim**, an approach that can be applied on top of existing popularity measures to increase the accuracy of their results. The main idea of our approach is that the popularity of papers in their cold start period can be better estimated based on the characteristics of other, similar papers. We calculate the similarity of papers using metapath analyses on the underlying scholarly knowledge graphs. Our experimental evaluation showcases the effectiveness of **ArtSim**, yielding noteworthy improvements in terms of Kendall's *tau* correlation and nDCG when applied on four state-of-the-art popularity measures.

Acknowledgments. We acknowledge support of this work by the project “Moving from Big Data Management to Data Science” (MIS 5002437/3) which is implemented under the Action “Re-inforcement of the Research and Innovation Infrastructure”, funded by the Operational Programme “Competitiveness, Entrepreneurship and Innovation” (NSRF 2014–2020) and co-financed by Greece and the European Union (European Regional Development Fund). Icons in Fig. 1 were collected from www.flaticon.com and were made by [Freepik](#), [Good Ware](#) and [Pixel perfect](#).

References

1. Bai, X., et al.: An overview on evaluating and predicting scholarly article impact. *Information* **8**(3), 73 (2017)
2. Balmin, A., Hristidis, V., Papakonstantinou, Y.: Objectrank: authority-based keyword search in databases. In: *VLDB* (2004)
3. Bollen, J., Van de Sompel, H., Hagberg, A., Chute, R.: A principal component analysis of 39 scientific impact measures. *PloS One* **4**(6), e6022 (2009)
4. Ghosh, R., Kuo, T., Hsu, C., Lin, S., Lerman, K.: Time-aware ranking in dynamic citation networks. In: *International Conference on Data Mining Workshops*, pp. 373–380 (2011)
5. Jaradeh, M.Y., et al.: Open research knowledge graph: next generation infrastructure for semantic scholarly knowledge. In: *International Conference on Knowledge Capture* (2019)
6. Kanellos, I., Vergoulis, T., Sacharidis, D., Dalamagas, T., Vassiliou, Y.: Impact-based ranking of scientific publications: a survey and experimental evaluation. *IEEE Trans. Knowl. Data Eng.* (2019). <https://ieeexplore.ieee.org/document/8836082>

7. Kendall, M.G.: Rank correlation methods (1948)
8. Larsen, P.O., von Ins, M.: The rate of growth in scientific publication and the decline in coverage provided by science citation index. *Scientometrics* **84**(3), 575–603 (2010)
9. Manghi, P., et al.: OpenAIRE research graph dump (2019). <https://doi.org/10.5281/zenodo.3516918>
10. Manghi, P., et al.: The OpenAIRE research graph data model (2019). <https://doi.org/10.5281/zenodo.2643199>
11. Nie, Z., Zhang, Y., Wen, J.R., Ma, W.Y.: Object-level ranking: bringing order to web objects. In: *WWW* (2005)
12. Salatino, Angelo A., Osborne, Francesco., Thanapalasingam, Thiviyan, Motta, Enrico: The CSO classifier: ontology-driven detection of research topics in scholarly articles. In: Doucet, Antoine, Isaac, Antoine, Golub, Korajka, Aalberg, Trond, Jatowt, Adam (eds.) *TPDL 2019*. LNCS, vol. 11799, pp. 296–311. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30760-8_26
13. Salatino, Angelo A., Thanapalasingam, Thiviyan., Mannocci, Andrea., Osborne, Francesco, Motta, Enrico: The computer science ontology: a large-scale taxonomy of research areas. In: Vrandečić, D., et al. (eds.) *ISWC 2018*. LNCS, vol. 11137, pp. 187–205. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00668-6_12
14. Sayyadi, H., Getoor, L.: FutureRank: ranking scientific articles by predicting their future PageRank. In: *Proceedings of the 2009 SIAM International Conference on Data Mining*. SIAM (2009)
15. Shi, C., Li, Y., Zhang, J., Sun, Y., Yu, P.S.: A survey of heterogeneous information network analysis. *IEEE Trans. Knowl. Data Eng.* (2017). <https://doi.org/10.1109/TKDE.2016.2598561>
16. Sun, Y., Han, J., Yan, X., Yu, P.S., Wu, T.: PathSim: meta path-based top-k similarity search in heterogeneous information networks. In: *VLDB* (2011)
17. Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z.: ArnetMiner: extraction and mining of academic social networks. In: *ACM SIGKDD*. ACM (2008)
18. Vergoulis, T., Chatzopoulos, S., Kanellos, I., Deligiannis, P., Tryfonopoulos, C., Dalamagas, T.: Bip! finder: facilitating scientific literature search by exploiting impact-based ranking. In: *CIKM* (2019)
19. Walker, D., Xie, H., Yan, K., Maslov, S.: Ranking scientific publications using a model of network traffic. *JSTAT* **2007**(06), P06010 (2007)
20. Xiong, Y., Zhu, Y., Yu, P.S.: Top-k similarity join in heterogeneous information networks. *IEEE Trans. Knowl. Data Eng.* **27**, 1710–1723 (2015)



Link Prediction in Bibliographic Networks

Pantelis Chronis^{1,2}(✉), Dimitrios Skoutas², Spiros Athanasiou²,
and Spiros Skiadopoulos¹

¹ University of the Peloponnese, Tripoli, Greece
{chronis, spiros}@uop.gr

² Athena Research Center, Athens, Greece
{dskoutas, spathan}@athenarc.gr

Abstract. Analysing bibliographic networks is important for understanding the process of scientific publications. A bibliographic network can be studied using the framework of Heterogeneous Information Networks (HINs). In this paper, we comparatively evaluate two different algorithms for link prediction in HINs on an instance of a bibliographic network. These two algorithms represent two distinct categories: algorithms that use path-related features of the graph and algorithms that use node embeddings. The results suggest that the path-based algorithms achieve significantly better performance on bibliographic networks.

1 Introduction

Analysing information about scientific papers, authors and venues provides useful insight into the scientific process. This information is often represented as a network, where authors are connected with their papers, and papers with the venues where they were published. This kind of network is referred to as a *bibliographic network*. Link prediction is the task of modelling the formation of edges in a network. In a bibliographic network, link prediction can be useful for various reasons, including recommendation of papers or venues, discovery of information that may be missing from the original dataset, and for gaining insights into the connectivity patterns of the networks, and consequently into the publication process.

From a data management perspective, a bibliographic network falls into the wider category of Heterogeneous Information Networks (HINs). A HIN is a graph, i.e., a set of nodes and edges, with the additional property that there are multiple *types* of nodes and edges. The existence of multiple types is important for modelling the graph. In particular, a sequence of node and edge types of a HIN is called a *metapath*. A metapath represents a composite relation between two nodes. For example, in a bibliographic network, the relation *Author* $\xrightarrow{\text{writes}}$ *Paper* $\xrightarrow{\text{isPublishedAt}}$ *Venue* is a metapath that connects an author to a venue in which she has published a paper. The concept of metapath is important for link prediction algorithms in HINs. These algorithms can be categorised in two main categories: (a) those that are based on graph topology, i.e.,

that focus directly on the metapaths that exist between any two nodes [1–4], and (b) those that use embeddings, which use low dimensional vector representations of the nodes of the graph [5–7].

In this paper, we compare the performance of two algorithms, representing the two mentioned categories, on the task of predicting links on a bibliographic network. The evaluation is performed on the DBLP dataset, which has two types of edges, using multiple metrics. The results suggest that, for this type of data, the graph topology based method achieves better results.

2 Related Work

Existing algorithms for link prediction in HINs can be distinguished in two main categories, depending on whether they are based on counting the metapaths between any two given nodes or on embeddings. The idea of counting the instances of a metapath to determine the relationship between two nodes first appears in [1], where a similarity measure between nodes based on the number of instances of a given metapath between them is proposed. Based on this, the counts of all metapaths are used in [2, 3] as features for a logistic regression model, trained for link prediction in an instance of the DBLP bibliographic network. The same metapath-count based model has also been used in [4] to predict the time that a link will occur, which is useful in datasets where time information is available.

Most of the recent work on link prediction in HINs belongs to the category of embedding based methods. In these, each node is represented by a low-dimensional vector of real numbers. Edge types may also have a vector or matrix representation, depending on the specific method. Link prediction is then performed through calculations of these representations. Various ways of obtaining and using embeddings have been proposed. Notably, in [5], node embeddings are obtained via metapath-constrained random walks. For each node, and for each given metapath, a large number of random walks are executed. The resulting node sequences are given as input to the skip-gram algorithm [5], which makes the inner product between the vectors of two nodes smaller, if these nodes are encountered closely in the random walks. In [6], embeddings are also obtained through random walks. However, in this case, a different embedding is first obtained for each metapath, and then all embeddings are combined in order to predict the existing links. In [7], each node is represented with a vector and each edge type is represented with a matrix. In this case, the node vectors are multiplied by the relation matrix and the euclidean distance is calculated from the resulting vectors, to predict the probability that an edge exists.

3 Description of Methods

3.1 Bibliographic Network

For our evaluation, we use a bibliographic network produced by the DBLP dataset. This dataset contains information on major computer science

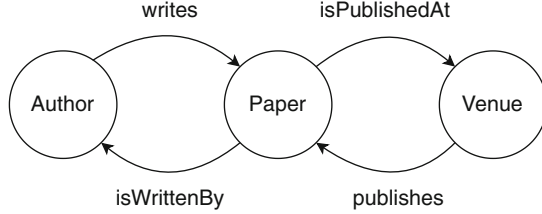


Fig. 1. The schema of the DBLP bibliographic network.

publications of the last five decades. Specifically it contains authors, papers, venues, and their relations. Next, we formulate the DBLP dataset as a HIN.

Formally, a HIN H is defined as $H = (V, E, L_V, L_E)$, where V is the set of nodes, E is the set of edges, L_V is the set of node types, and L_E is the set of edge types. For the DBLP network the node types are $L_v = \{Author, Paper, Venue\}$, while the edge types are $L_E = \{writes, isWrittenBy, publishes, isPublishedAt\}$. The way that the node types are connected, via the edge types, is represented by the network schema of H , which is shown in Fig. 1. For example, nodes of type *Author* are connected with edges of type *writes* to nodes of type *Papers*. A metapath is a path on the network schema of H . For example, *Author* \xrightarrow{writes} *Paper* $\xrightarrow{isPublishedAt}$ *Venue* is a metapath that connects an author to a venue he has published in. Since the edge types can be inferred from the node types, we refer to the metapaths using only their node types, i.e., *Author* \rightarrow *Paper* \rightarrow *Venue*, or $A - P - V$ for short.

3.2 Metapath-Counts

Metapath-Counts algorithm [2] models the probability of a link existing between two nodes as a function of the number of metapaths that connect them. The model is developed for each edge type separately. For edge type $l_e \in L_E$, there exists a set $R_m(l_e)$, with size $|R_m(l_e)| = d(l_e)$, comprising all metapaths up to length m which connect the node types corresponding to l_e . For convenience, we enumerate the metapaths of $R_m(l_e)$ as $r_{i,l_e}, 1 \leq i \leq d(l_e)$.

For a pair of nodes u, v and a metapath r , we define function $c(r, u, v)$ which returns the number of instances of metapath r that exists between u and v (e.g., the number of publications of an author to a venue). The vector of metapath counts between nodes u, v is defined as $\mathbf{x}_{l_e} : x_{i,l_e} = c(r_{i,l_e}, u, v)$. Then the probability that an edge of type l_e exists between u and v is modeled as:

$$p_{l_e}(u, v) = \sigma(\mathbf{x}_{l_e} \cdot \mathbf{w} + b) \tag{1}$$

where σ is the logistic sigmoid function, $\sigma(x) = \frac{1}{1+e^{-x}}$, and \cdot is the inner product operator. Vector \mathbf{w} contains the weights given by the model to the counts of the different metapaths and b is a constant bias on the probability.

To train the model for edge type l_e , we select all pairs u, v of H that are connected with an edge of type l_e in set $P(l_e)$. We also define a set of node

pairs z, y , of the node types defined by l_e , that do not have an edge between them, which we denote as $N(l_e)$. Since bibliographic networks tend to be sparse, which means that the size of $P(l_e)$ is $O(n)$, while $N(l_e)$ is $O(n^2)$, we do not include all pairs without an edge in $N(l_e)$. Instead we select a random sample. The technique of selecting a sample of the “negative” (non existing) edges is called negative sampling and is widely used in link prediction [7]. We select the negative sample in two steps. First, for each pair $u, v \in P(l_e)$, we randomly select α nodes y , of the same node type as v , from the m step neighborhood of u , asserting that edge does not exist ($y \notin P(l_e)$). Then, we randomly select α nodes y , of the same node type as v , from all nodes $V \setminus P(l_e)$, again asserting that the edge does not exist. Finally, the parameters are obtained by maximizing the log-likelihood on sets $P(l_e)$, $N(l_e)$:

$$L_{l_e}(\mathbf{w}, b) = \sum_{(u,v) \in P(l_e)} \log(p_{l_e}(u, v)) + \sum_{(u,v) \in N(l_e)} \log(1 - p_{l_e}(u, v)) \quad (2)$$

This function asserts that the probability that the model assigns to existing edges is as close to 1 as possible, while the probability that it assigns to non connected pairs is as close to 0 as possible. We note that in [2] a few other variations of the metapath-count features are also described, however we focus on the ones described in this section for conceptual and computational simplicity.

3.3 Metapath2Vec

Metapath2Vec algorithm [5] assigns a vector representation for each node of the HIN, based on its proximity to other nodes, and models the probability of a link as a function of these representations. The proximity is calculated via metapath constrained random walks on the HIN. Formally, each node v_i is associated with d dimensional representation \mathbf{x}_i . The probability that a link exists between v_i and v_j is modeled as:

$$p(v_i, v_j) = \frac{e^{\mathbf{x}_i \cdot \mathbf{x}_j}}{\sum_{v_k \in V} e^{\mathbf{x}_i \cdot \mathbf{x}_k}} \quad (3)$$

The algorithm receives a set of metapaths R as input. For each metapath $r \in R$ it performs I random walks of length k on the graph. At each step of the random walk, it chooses uniformly at random one of the available neighbors, having the type that is defined by metapath r . Subsequently, for each node v , the algorithm creates a multiset $C(v)$ containing all nodes that were encountered in a distance of w or less steps from v , for every random walk. Vectors \mathbf{x}_i are randomly initialized and optimized via gradient descent so that they maximize the following log likelihood:

$$L = \sum_{v_i \in V} \sum_{v_j \in C(v)} \log(p(v_i, v_j)) \quad (4)$$

Equation 4 require $O(n)$ evaluations of Eq. 3 which itself has $O(n)$ complexity, for calculating the denominator. To avoid this complexity, negative sampling is

used. Specifically α nodes v_k (Eq. 3) are sampled uniformly at random from all nodes in V with the same type as v_i . We note that Metapath2Vec models all edge types simultaneously.

4 Evaluation

We evaluate the two algorithms on a large sample of the DBLP dataset. We select a sample, instead of using the entire DBLP dataset, to limit the computational requirements of the experiments. This sample consists of authors, papers and venues from the academic field of data management. We obtain it by finding all venues that contain the word “data” in their title, and select the venues, the papers published in these venues and their authors. In total, the sample consists of 507 venues, 374,034 authors, and 357,091 papers, that span the time period from 1969 to 2017, with the majority occurring in the last 10 years of the dataset. The evaluation is performed separately for each of the two distinct edge types: “Writes” and “Publishes”. In each case we use 70% of the edges for training, 10% for validation/tuning and 20% for testing.

For each algorithm, we evaluate the performance in link prediction using three metrics: MeanRank, HITS@10 and MeanReciprocalRank(MRR). These measures are calculated by comparing the scores of positive and negative edges (i.e., edges that exist in the graph and edges that do not exist in the graph). We select the negative samples using the technique described in Sect. 3.2, with $\alpha = 50$. MeanRank corresponds to the mean rank of the score of the positive edges, if the algorithm is applied to all edges and the results are sorted in descending order. For MeanRank lower values are better and the optimal value is 1. To calculate HITS@10 and MRR, we partition all edges (u, v) (positive and negative) according to their starting node u . For each partition, we apply the algorithms and sort the scores in descending order. Hits@10 is the number of positive edges in the 10 higher scores, for each partition. MRR is the inverse of the rank of the first positive edge, for each partition. For HITS@10, MRR higher score are better, and the optimal values are 10,1 respectively.

The results are presented in Table 1. We see that MetapathCounts achieves a better score for both edge types and according to all metrics. The difference is larger for MeanRank metric, while it is not so large for HITS@10 and MRR metrics. This means that, while Metapath2Vec correctly identifies links that have high probability to exist, it does not manage to correctly differentiate between links that have medium and small probability to exist. On the other hand MetapathCounts achieves better performance for all edges, both those that are very likely to exist and those that exist with smaller probability than others. A possible explanation for the difference in performance of these two algorithms is that bibliographic networks, such as the DBLP network, tend to be too sparse for the embedding algorithm to be trained adequately.

Another interesting aspect of MetapathCounts algorithm is that it provides a weight for the importance of each metapath, optimized for the task of link prediction. The weights for our experimental setting are presented in Table 2.

Table 1. Link prediction scores

Edge type	Writes			Publishes		
	MeanRank	HITS@10	MRR	MeanRank	HITS@10	MRR
MetapathCounts	31815	3.88	0.95	30417	5.40	0.72
Metapath2Vec	53450	3.85	0.92	41570	4.70	0.71

Table 2. Metapath coefficients

Metapath	A-P-A-P	A-P-V-P	P-A-P-V
Coefficient	2.019	0.067	0.300

Metapath A-P-A-P, exists between author A_1 and paper P_1 , when P_1 has another author A_2 that has been a coauthor with A_1 on another paper P_2 . On the other side, metapath P-A-P-V exists when A_1 and A_2 have published to a common venue. The coefficients of Table 2 suggest that the relationship between coauthors is more important than the relationship between authors that have published to the same venue, and they also offer a quantification of this difference. This can be useful for various analytic tasks, such as estimating the importance of nodes. Also MetapathCounts is a simpler model, with significantly less parameters, than Metapath2Vec or other embedding models.

These results suggest that MetapathCounts is more effective than Metapath2Vec for modelling a bibliographic network. This is an interesting result and motivates the further improvement of the framework defined by MetapathCounts algorithm. A possible direction of improvement would be to include additional features, such as year of publication, into the formulation of the model.







Acknowledgments. This work was partially funded by the EU H2020 project Smart-DataLake (825041).

References

1. Sun, Y., Han, J., Yan, X., Yu, P.S., Wu, T.: PathSim: meta path-based top-k similarity search in heterogeneous information networks. VLDB 4, 992–1003 (2011)
2. Sun, Y., Barber, R., Gupta, M., Aggarwal, C.C., Han, J.: Co-author relationship prediction in heterogeneous bibliographic networks. In: ASONAM (2011)
3. Yu, X., Gu, Q., Zhou, M., Han, J.: Citation prediction in heterogeneous bibliographic networks. In: SDM (2012)
4. Sun, Y., Han, J., Aggarwal, C.C., Chawla, N. V. When will it happen?: Relationship prediction in heterogeneous information networks. In: WSDM (2012)
5. Dong, Y., Chawla, N., Swami, A.: metapath2vec: scalable representation learning for heterogeneous networks. In: KDD (2017)
6. Shi, C., Hu, B., Zhao, W.X., Yu, P.S.: Heterogeneous information network embedding for recommendation. In: TKDE (2019)
7. Chen, H., Yin, H., Wang, W., Wang, H., Nguyen, Q., Li, X.: PME: projected metric embedding on heterogeneous networks for link prediction. In: KDD (2018)



Open Science Observatory: Monitoring Open Science in Europe

George Papastefanatos¹(✉) , Elli Papadopoulou¹ , Marios Meimaris¹ ,
Antonis Lempesis¹ , Stefania Martziou¹, Paolo Manghi² ,
and Natalia Manola¹ 

¹ ATHENA Research Center, Marousi, Greece
gpapas@athenarc.gr

² Consiglio Nazionale delle Ricerche (CNR), Pisa, Italy

Abstract. Monitoring and evaluating Open Science (OS) practices and research output in a principled and continuous way is recognised as one of the necessary steps towards its wider adoption. This paper presents the Open Science Observatory, a prototype online platform which combines data gathered from OpenAIRE e-Infrastructure and other public data sources and informs users via rich visualizations on different OS indicators in Europe.

Keywords: Open science monitoring · Scholarly communication metrics · Data visualization

1 Introduction

Open Science (OS) refers to new paradigms and ongoing changes in the way research is conducted, based on open digital access to research artefacts and data-driven science and affecting all phases of the research lifecycle, from the conceptualization of an idea, the collaboration of researchers, the production of scientific results, the evaluation and validation of the research output. OS is recognized as one of the key drivers for promoting wider accessibility, transparency, reproducibility, trusted collaboration for research excellence and citizen participation in the scientific process.

The rapid emergence of OS initiatives and practices has highlighted the need for monitoring and assessing its impact in a principled way; a multitude of efforts exist capturing different quality aspects of the Open science movement, such as the degree of Openness in journal publications. The realization of such an environment is a continuous process, whose basic requirements include a principled approach for monitoring and measuring the uptake and impact of Open Science trends and practices, via a clear set of high level monitoring indicators, such as the openness, findability and accessibility to open science elements and a well-defined set of metrics that can quantify the above indicators.

This paper presents the Open Science Observatory (OSO¹), a prototype online platform which combines data gathered from OpenAIRE e-Infrastructure² and other public data sources and informs users via rich visualizations on different Open Science aspects in Europe for Horizon 2020 and other funding sources. In contrast with past studies, it offers an interactive and dynamic environment which leverages open data to provide up-to-date figures for indicators related to OS. It follows a top-down methodology for deriving indicators based on high level monitoring targets and employs metrics which can measure the openness of research output (publications, data, software or other research products) on various aspects (e.g., gold/green/fair) and the regional or thematic distributions (at EU, Country and Repository-level). It aims to provide services to funding agencies, policy makers, research organizations and researchers and help them assess different dimensions of OS research.

In Sect. 2, we provide RW, Sect. 3 provides the methodology, Sect. 4 presents the basic elements (metrics reported and functionality delivered) of the observatory, whereas Sect. 5 concludes and provides insights and next steps in its implementation.

2 Related Work

The first initiatives for OS monitoring are the Open Digital Science approach [4] and the EU-funded Open science monitor study (initially published in 2017 as a pilot and re-launched by the European Commission in 2018) [2]. Open Digital Science has introduced indicators considering the research lifecycle steps of “Conceptualisation & data gathering/creation”, “Analysis”, “Diffusion of results”, “Review and evaluation”, as well as measurements of other Open Science resources, including drivers and constraints, namely/those being “Reputation system, recognition of contributions, trust”, “OS skills & awareness”, “Science with society”. The Open Science Monitor study of 2017 put together different aspects of OS characteristics that involve research artefacts (mainly open access publications and research data), Scholarly communication activities (altmetrics, peer reviews etc), citizen science and public engagement. The follow up project focused on the provisioning of metrics assessing drivers and barriers to open science adoption; identifying the impacts (both positive and negative) of open science and finally supporting evidence-based policy actions. These studies capture and assess various OS characteristics in a specific point in time, without providing more concepts and solutions on how these characteristics can be continuously monitored or how results are updated to provide the overall OS community with up-to-date feedback.

One of the first efforts to measure OA in journals is the “HowOpenIsIt?” guide [1] which provided a standardised way to open principles and introduced levels of openness in journals, ranking them from closed access to more “open” approaches. Other attempts, such as Nichols & Twidale [6], have examined

¹ <http://oso.madgik.di.uoa.gr>.

² www.openaire.eu.

the possibility of developing an h-index contributing to the wider picture of researchers' OA practice to provide metrics in the form of an individual's index that informs about access, re-use and preservation of research outputs. Finally, the TOP Factor shows compliance of publishers with TOP Guidelines [7] for data sharing policies.

Several approaches have been proposed by the communities/groups to define framework within which the FAIRness [9] of published data can be assessed. RDA have published a maturity model [8] for assessing and evaluating each one of the F, A, I, R principles. FAIR Metrics gives a "core set of semi quantitative metrics having universal applicability for the evaluation of FAIRness, and a framework within which additional metrics can be generated by the community".

3 Methodology

The implementation of the Open Science Observatory follows the guidelines and methodological approach which was introduced in the context of the EOSCPilot project³. The different steps are described below.

Step 1. Identification of the Open Science Activities: It identifies the parts of the OS lifecycle which are of interest in the monitoring process, such as the conceptualization of a research task, the data and literature collection, the analysis and development of the research output, the publication, the review and evaluation of the research result as well as the reuse and reproducibility of results by the scientific community.

Step 2. Derivation of monitoring targets: The second step derives high level objectives, i.e., target dimensions to be measured in the monitoring process. Policies on Open Access at international, national and regional levels, as well as micro policies are considered as primary sources for deriving more concrete measurable targets (e.g., Openness, FAIRness, etc.) that should be monitored in the framework.

Step 3. Identification of the OS Resources and Indicators: In the next step, the monitoring targets are being mapped to OS resources they apply to, as well as to indicators that quantify these targets. OS resources are well-defined outputs of OS practices, such as a publication in open access journals, research data made available in open access repositories, open source software, etc.

Step 4. Design of monitoring processes and workflows: Each indicator must be associated with a set of processes, for the collection of data, the validation and scoring of metrics (e.g., weighting of metrics for deriving an accumulated score for a target dimension), the visualization of the results, and so on. These processes must be well documented in the form of workflows and tasks, to be performed at a regular basis for the collection and quantification of the indicators.

Step 5. Modelling and implementation of the framework: The next step involves the detailed design, implementation, and customization of the framework.

³ <https://eoscpilot.eu/themes/wp3-policy/eosc-open-science-monitor-specifications>.

Step 6. Continuous validation of the monitoring targets: The last step follows the operation of the OS monitoring framework and the continuous validation and refinement of the monitoring methodology (i.e., targets and indicators), such it can be effective and follow new OS practices and policies.

4 Open Science Observatory

The Open Science Observatory follows the aforementioned methodology offering a first implementation of these specifications.

Data Sources. The primary source of information is OpenAIRE's research graph [5]. OpenAIRE Research Graph primarily contains metadata records about i) research output such as research literature, data, software, other research products, and associates them with ii) organizations involved in the research life-cycle, such as universities, research organizations, funders, iii) data-sources from which content is harvested, including institutional and thematic repositories, journals, aggregators, funders' databases and finally iv) projects and funding agencies having funded research output. The graph is regularly updated via content harvesting of the datasources followed by a data deduplication and enrichment phase in order to sync and reflect changes from the sources to the an integrated Information Space. Other sources include: the Directory of Open Access Journals (DOAJ⁴) which brings information about OA journals, the Registry of Open Access Repository Mandates and Policies (ROARMAP⁵) which provides information about OA policies per country and Eurostat's open data for R&D expenditure per country⁶.

Metrics. In its current form, OSO primarily includes indicators related to the *Openness* of research outputs, namely OA publications, datasets and software, reporting on the performance of projects and organizations as regards the production of OS resources as well the performance of funders in funding OS practices. In a next phase, it will include high level targets and indicators measuring the *FAIRness* of research output, the *Collaboration* of research communities around OS practices, and the *Impact* of OS in research excellence and *Innovation*. Metrics are aggregated at the European level as well as reported at the country level. In each country, the user can also browse metrics referring to a specific organization, repository or project which are associated with research outputs from this country. In Table 1 the list of metrics visualized in the observatory are presented.

Demonstrated Functionality. The observatory aims at offering interactive visualizations for users to navigate and browse the reported metrics. The landing page provides an overview of the metrics for Europe; i.e., a map, in which each

⁴ <https://doaj.org/>.

⁵ <http://roarmap.eprints.org/>.

⁶ https://ec.europa.eu/eurostat/statistics-explained/index.php/R_%26_D_expenditure.

Table 1. Observatory metrics

Metric	Description	Scope
<i>General</i>		
OA publications	The total number of OA publications; their percentage to the total number of available publications	Europe, Country
OA datasets	The total number of OA datasets; their percentage to the total number of available publications	
OA repositories	The total number of OA repositories	
OA journals	The total number of OA journals	
OA policies	The total number of organizations that have at least one OA policy associated with them	
<i>Country InfoBox</i>		
RnD Expenditure	The total RnD expenditure for a country	Country
Funding sources	The number of funding sources in a country	
Organizations funded	The number of organizations that have been funded for research	
Organizations funded by EU	The number of organizations that have participated in a project funded by the EU	
<i>Green vs Gold Publications</i>		
Green vs Gold	The total number of publications that have been published through a green open access route, vs the total number of publications that have been published through a gold open access route	Country
Gold Open Access	Organizations in a country in descending order of their total count of gold open access publications	
Green Open Access	Organizations in a country in descending order of their total count of green open access publications	
<i>Funding Open Science</i>		
OA Research Output	Per-year count and comparison of OA publications, datasets and software resulted from a project funded by EU	Country
Organizations by Research Output	Organizations, along with the total number of OA publications, datasets and software that have authors that are affiliated with each organization	
Repositories by Research Output	Repositories, along with the total number of publications, datasets and software that reside in each repository	
Projects by Research Output	Projects, along with the total number of publications, datasets and software, funded by a project	

country is annotated with indicators. Next, the user can navigate to a country's page, where the metrics referring to this country are presented. Metrics are visually presented via interactive charts (bars, pies, lines) and tables and they are organized in sections as shown in Table 1; e.g., the country's infobox offers a brief glance at numbers related to the funding of OS in this country. In each visualized section, the user can download the data behind the chart, embed the chart in another web page or download a pdf version. The observatory is built on open-source technologies. Following best practices in the visualization of big data [3], it uses Apache Impala to query the big volume of the OpenAIRE research graph and precompute metrics, such that users can have better response times. The front end is built on AngularJS and HighChartJS libraries.

5 Conclusions

This paper presented the Open Science Observatory, a prototype online tool which offers a variety of visualizations and reports on metrics for open science. It aspires to offer benefits both to the research organisations using it to measure the levels of OS implementation and impact to their community as well as to funding agencies and policy makers for informed decision making. Next steps include the incorporation of new indicators, most notably, FAIR data metrics, that will capture broader aspects of the OS evolution and practices.

Acknowledgments. This work has been funded by H2020 OpenAIRE Advance project (Nr.777541) and VisualFacts project (#1614 - 1st Call of the Hellenic Foundation for Research and Innovation Research Projects for the support of post-doctoral researchers).

References

1. HowOpenIsIt? a guide for evaluating the openness of journals. SPARC and PLOS (2014), <https://sparcopen.org/our-work/howopenisit/>. Accessed Apr 2020
2. Open science monitor. EU. https://ec.europa.eu/info/research-and-innovation/strategy/goals-research-and-innovation-policy/open-science/open-science-monitor_en. Accessed Apr 2020
3. Andrienko, G.L., et al.: Big data visualization and analytics: future research challenges and emerging applications. In: BigVis2020 Workshop of EDBT/ICDT (2020). <http://ceur-ws.org/Vol-2578/BigVis1.pdf>
4. Lampert, D., Lindofer, M.: Initial set of open digital science indicators (2016). <https://doi.org/10.5281/zenodo.48991>
5. Manghi, P., et al.: Openaire research graph dump, December 2019. <https://doi.org/10.5281/zenodo.3516918>
6. Nichols, D.M., Twidale, M.B.: Metrics for openness. *J. Assoc. Inf. Sci. Technol.* **68**(4), 1048–1060 (2017). <https://doi.org/10.1002/asi.23741>
7. Nosek, B.A., et al.: Promoting an open research culture. *Science* (2015). <https://doi.org/10.1126/science.aab2374>
8. RDA FAIR Data Maturity Model Working Group: Fair data maturity model: specification and guidelines. <https://doi.org/10.15497/RDA00045>
9. Wilkinson, M., et al.: A design framework and exemplar metrics for fairness. *Sci. Data* **5**, 180118 (2018). <https://doi.org/10.1038/sdata.2018.118>



Skyline-Based University Rankings

Georgios Stoupas¹, Antonis Sidiropoulos², Dimitrios Katsaros³,
and Yannis Manolopoulos⁴

¹ Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece
`grgstoupas@csd.auth.gr`

² International Hellenic University, 57400 Thessaloniki, Greece
`asidirop@gmail.com`

³ University of Thessaly, 38221 Volos, Greece
`dkatsar@e-ce.uth.gr`

⁴ Open University of Cyprus, 2220 Nicosia, Cyprus
`yannis.manolopoulos@ouc.ac.cy`

Abstract. University rankings comprise a significant tool in making decisions in our modern educational process. In this paper, we propose a novel university ranking method based on the *skyline* operator, which is used on multi-dimensional objects to extract the non-dominated (i.e. “prevailing”) ones. Our method is characterized by several advantages, such as: it is transparent, reproducible, without any arbitrarily selected parameters, based on the research output of universities only and not on publicly not traceable or random questionnaires. Our method does not provide meaningless absolute rankings but rather it ranks universities categorized in equivalence classes. We evaluate our method experimentally with data extracted from Microsoft Academic.

Keywords: University rankings · Skyline · Rainbow ranking index

1 Introduction

University rankings are of major importance for decision making by prospective students, by academic staff, and by funding agencies. The placement of universities in these lists is a crucial factor and academic institutions adapt their strategy according to the particular criteria of each evaluation system. The reason for this inclination can be understood by considering the “Thomas theorem” from sociology, which states “if men define situations as real, they are real in their consequences” [8]. As mentioned in [2]: “if rank positions between two universities define performance differences as real, they are real in their consequences (although the university ranking shows only slight differences between the universities’ scores)”.

Probably, the 3 most popular global rankings are: ARWU, QS and THE. Another quite well-known ranking list is Webometrics, whereas there also exist a few ranking lists developed by universities, e.g., CWTS ranking of Leiden

University, École Nationale Supérieure des Mines de Paris, Middle East Technical University, Wuhan University, and Shanghai Jiao Tong University, which founded the ARWU organization. All these lists base their respective ranking on some set of indicators, which differ from one organization to the other. The reader can retrieve these indicators from the respective sites^{1, 2, 3, 4, 5}. Despite their popularity, these rankings are heavily criticized for their reproducibility, statistical soundness, etc. [1, 4, 5, 9].

Here, we focus on an academic-based (teaching and research performance) ranking, i.e., in a similar approach to CWTS ranking of the Leiden University, and we propose an orthogonal method to rank academic institutions based on the *skyline* operator, which is applied on multi-dimensional objects to extract the non-dominated (i.e. “prevailing”) ones. The contribution of our method and its advantages over the popular university rankings are the following:

- it focuses on the research output of the universities, and does not rely on questionnaires,
- it uses a set of indicators well-known to the whole academic community from Google Scholar metrics,
- it does not use arbitrary weights for each indicator, but treats all indicators equally in a symmetric manner,
- it avoids the nonsense absolute rankings, where there is no serious meaning in claiming that the i -th university is better the $(i+1)$ -th one.
- it provides a list with a single structure, contrary to the popular rankings, where paradoxically the first few hundreds of universities are ranked in absolute order, whereas the rest follow in groups.
- it is not prone to inconsistent fluctuations from year to year,
- it is fully customizable in the sense that it can use any set of research key-performance indicators.

The structure of the remaining part of this paper is as follows. Section 2 explains the Skyline operator and its derivative, namely Rainbow Ranking. Section 3 gives the results of the application of the Rainbow Ranking to university ranking. Finally, Sect. 4 concludes the article.

2 Skyline and Rainbow Ranking

The Skyline operator is used as a database query to filter only those ‘objects’ that are not worse than any other (they are not dominated) [3]. A useful application of Skylines in scientometrics is reported in [6] where 3-d Skyline sets of ‘dominating’ researchers for each year of the period 1992–2013 were produced. An extension

¹ https://en.wikipedia.org/wiki/Academic_Ranking_of_World_Universities.

² https://en.wikipedia.org/wiki/QS_World_University_Rankings.

³ https://en.wikipedia.org/wiki/Times_Higher_Education_World_University_Rankings.

⁴ https://en.wikipedia.org/wiki/Webometrics_Ranking_of_World_Universities.

⁵ https://en.wikipedia.org/wiki/CWTS_Leiden_Ranking.

of the Skyline operator, namely the Rainbow Ranking [7], applies iteratively the Skyline operator until all entities (i.e., scientists) of a dataset have been classified into a Skyline level. More specifically, given a set of scientists X_1 , the first call of the Skyline operator produces the first Skyline level, which is denoted as S_1 . Next, the Skyline operator is applied on the dataset $X_1 - S_1$, to derive the second Skyline layer, denoted as S_2 . This process continues until all the scientists of the dataset have been assigned to a particular Skyline level S_i . To give more semantics to the method, a particular value should characterize the Skyline levels. Should this value be the iteration number, then this would convey limited interpretability since the relativeness would be lost. It is crucial to designate the position of scientist among their peers. Therefore, a normalization of this value is necessary. Thus, the RR -index of a researcher a is defined as:

$$RR(a) = 100 - 100 \times \frac{|A_{above}(a)| + |A_{tie}(a)|/2}{|A|}$$

where A is the set of scientists, $A_{above}(a)$ is the number of scientists at higher Skyline levels than scientist a , and $A_{tie}(a)$ is the number of scientists at the same Skyline level with scientist a , excluding scientist a . Apparently, it holds that: $0 < RR(a) \leq 100$. A key component for the RR -index concept is the number of the Skyline dimensions. By selecting different bibliometric indices as Skyline dimensions, RR -index can be fully customizable.

3 Ranking Universities with the RR -index

Here, the RR -index is generalized to higher conceptual levels. We present the dataset used and the Skyline dimensions. Then, we present the experimental results at three levels: at author, faculty and institutional level.

Dataset. For our experiments we have used the Microsoft Academic Search (MAS⁶) database. We have downloaded the Microsoft Academic Graph from the Open Academic Graph work-group (AMiner⁷). The initial dataset consisted of 253,144,301 authors with 208,915,369 publications. Out of this initial dataset we kept only the publications having a Document Object Identifier (DOI⁸) as well as the publication year. This cleaning led to selecting 77,080,039 publications authored by 84,818,728 distinct researchers. For our experiments, the authors of the Greek Universities were identified and two data sets were created:

1. the first dataset consists of the academic staff of 19 CS faculties of 17 major Greek universities, i.e., 539 persons.
2. the second dataset consists of all authors with affiliation in the aforementioned 17 Greek universities. This dataset consists of the academic staff of the universities plus every researcher affiliated to any of these universities.

⁶ <https://academic.microsoft.com>.

⁷ <https://www.aminer.cn/oag2019>.

⁸ <https://doi.org>.

Skyline Dimensions. The dimensions of RR -index are the indicators used by Google Scholar: (a) Cit : number of citations to all publications, (b) $Cit-5$: number of citations during the last 5 years to all publications, (c) h -index, (d) h -index-5: largest number h such that h publications have at least h new citations during the last 5 years, (e) $i10$: number of publications with at least 10 citations, (f) $i10-5$: number of publications that have received at least 10 new citations during the last 5 years.

3.1 RR -index for Faculty Members

Initially the RR -index was calculated at the level of individuals for the 539 members of the Greek CS faculties. The RR -index clusters the individuals into 47 groups. Table 1 presents the top-3 ranking levels as derived by the RR -index.

Table 1. Rainbow Ranking for authors, the 3 top RR -levels.

Author	RR -level	RR -index	Cit	h	$i10$	$Cit-5$	$h-5$	$i10-5$
Nikos Hatzirygiou	1	99.81	11009	42	115	5501	26	73
George Karagiannidis	1	99.81	8079	49	155	3856	34	98
Ioannis Pitas	1	99.81	12568	55	226	2774	25	77
K.A. Antonopoulos	2	98.77	1744	23	38	948	20	26
Minos Garofalakis	2	98.77	4824	40	80	871	18	29
Yannis Manolopoulos	2	98.77	5651	33	104	1569	17	40
Petros Maragos	2	98.77	7499	42	122	1500	17	47
Konstantina Nikita	2	98.77	3405	29	98	1246	18	34
John Psarras	2	98.77	3212	30	89	1342	19	39
Grigorios Tsoumakas	2	98.77	3383	24	38	1785	18	28
Ioannis Vlahavas	2	98.77	3468	28	64	1560	18	32
Aggelos Bletsas	3	96.98	4285	19	34	1205	13	21
Pavlos Georgilakis	3	96.98	2367	25	57	1439	14	29
Aggelos Kiayias	3	96.98	5862	25	33	554	14	16
Stefanos Kollias	3	96.98	4783	31	95	1007	14	19
Aristidis Likas	3	96.98	4068	34	64	1350	17	32
Sotiris Nikolettas	3	96.98	2331	26	75	679	13	21
Stavros Papanthanas	3	96.98	2620	25	41	1286	19	27
Ioannis Pratikakis	3	96.98	2729	28	55	1153	19	38
Anastasios Tefas	3	96.98	2675	26	64	1178	17	37
Sergios Theodoridis	3	96.98	3535	27	72	1128	16	30
Yannis Theodoridis	3	96.98	3920	31	65	1095	17	32

3.2 RR -index for CS Faculties

Stepping now to a higher conceptual level and generalizing the previous approach, we compute the RR -index of the 19 largest CS faculties, where the previous 539 individuals belong. This generalization is achieved by accumulating

all the values of the adopted 6 features of all the faculty members belonging to each faculty. For example, the *Cit* value expresses the total number of citations received by all faculty members of each department. Table 2 shows the top-3 *RR*-index of these 19 CS faculties, which are grouped into 9 Skyline levels.

Table 2. Rainbow Ranking for CS faculties

Fac-Univ	#Staff	<i>RR</i> -level	<i>RR</i> -index	<i>Cit</i>	<i>h</i>	<i>i</i> 10	<i>Cit</i> -5	<i>h</i> -5	<i>i</i> 10-5
ece-ntua	71	1	100	91077	105	1822	30979	60	421
di-uoa	39	2	92.11	46897	88	899	12066	40	184
inf-auth	29	2	92.11	53740	98	877	17588	50	197
csd-uoc	24	3	78.95	29925	77	552	8188	36	120
ece-tuc	24	3	78.95	28858	75	432	8902	38	115
ee-auth	28	3	78.95	28842	72	612	10542	44	191

The first ranking level consists of 1 faculty only: the School of Electrical and Computer Engineering of the National Technical University of Athens. On the other hand, we notice that the second level consists of 2 CS faculties, whereas the third level consists of 3 CS faculties. This fact is a proof of concept, i.e. these faculties have the same *RR*-index and belong to the same equivalence class, without any of them dominating the others.

Table 3. Rainbow Ranking for Greek Universities

University	<i>RR</i> -level	<i>RR</i> -index	<i>Cit</i>	<i>h</i>	<i>i</i> 10	<i>Cit</i> -5	<i>h</i> -5	<i>i</i> 10-5
uoa	1	100	7078897	841	61172	3016544	552	24711
auth	2	91.18	3356467	548	32761	1578225	368	11249
ntua	2	91.18	2663388	498	19926	1416386	378	6886
uoi	3	79.41	2156665	513	20445	952141	344	8387
uoc	3	79.41	2125246	466	24320	805642	264	8769

3.3 *RR*-index for 17 Greek Universities

Finally, the *RR*-index values for the above 17 Greek universities were calculated using the second dataset. Again, note that each feature value was accumulated over the total number of the academic staff in each university. Notably, these 17 universities are grouped in 12 ranking levels. Table 3 shows the top 3 *RR*-index of these accumulated results for the 6 Skyline features. Table 4 shows the full names of the universities. The grouping created by applying our Rainbow Ranking method was relatively limited. This is due to the fact that the number of universities is small and the feature values vary widely. In turn, the latter fact is due to the different sizes of the universities both in terms of the number of faculties as well as the number of academic staff.

Table 4. Greek Universities acronyms and full names

Acronym	University name	Acronym	University name
auth	Aristotle University of Thessaloniki	tuc	Technical University of Crete
ntua	National Technical University of Athens	uoc	University of Crete
uoa	National & Kapodistrian University of Athens	uoi	University of Ioannina

4 Conclusions

This article proposes an alternative approach to rank universities by elaborating on the multidimensional Skyline operation, and the Rainbow Ranking methodology. In particular, our method provides ranked sets, in terms of equivalence classes, instead of ranked lists as provided by the traditional university rankings. The method alleviates many of the shortcomings of previous university rankings methods. The obtained results prove the validity of our approach. The proposed methodology can be further elaborated and tested towards richer multidimensional data representing other sets of key-performance indicators, such as more academic or non-academic ones. It can also be expanded across universities around the world and compared to existing rankings.

References

1. Angelis, L., Bassiliades, N., Manolopoulos, Y.: On the necessity of multiple university rankings. *COLLNET J. Scientometrics Inf. Manage.* **13**(1), 11–36 (2019). <https://doi.org/10.1080/09737766.2018.1550043>
2. Bornmann, L., Marx, W.: Thomas theorem in research evaluation. *Scientometrics* **123**(1), 553–555 (2020). <https://doi.org/10.1007/s11192-020-03389-6>
3. Börzsönyi, S., Kossmann, D., Stocker, K.: The skyline operator. In: *Proceedings 17th IEEE International Conference on Data Engineering (ICDE)*, pp. 1–20 (2001). <https://doi.org/10.1109/ICDE.2001.914855>
4. Johnes, J.: University rankings: what do they really show? *Scientometrics* **115**(1), 585–606 (2018). <https://doi.org/10.1007/s11192-018-2666-1>
5. Manolopoulos, Y., Katsaros, D.: Metrics and rankings: Myths and fallacies. In: *Revised Selected Papers, 18th International Conference on Data Analytics & Management in Data Intensive Domains (DAMDID/RCDL)*, pp. 265–280. Moscow, Russia (2017). https://doi.org/10.1007/978-3-319-57135-5_19
6. Sidiropoulos, A., Gogoglou, A., Katsaros, D., Manolopoulos, Y.: Gazing at the skyline for star scientists. *J. Inform.* **10**(3), 789–813 (2016). <https://doi.org/10.1016/j.joi.2016.04.009>
7. Stoupas, G., Sidiropoulos, A., Gogoglou, A., Katsaros, D., Manolopoulos, Y.: Rainbow ranking: an adaptable, multidimensional ranking method for publication sets. *Scientometrics* **116**(1), 147–160 (2018). <https://doi.org/10.1007/s11192-018-2731-9>
8. Thomas, W., Thomas, D.: *The Child in America*. Knopf, Oxford (1928)
9. Van Raan, A.F.: Fatal attraction: conceptual and methodological problems in the ranking of universities by bibliometric methods. *Scientometrics* **62**(1), 133–143 (2005). <https://doi.org/10.1007/s11192-005-0008-6>

Doctoral Consortium



Supervised Machine Learning Model to Help Controllers Solving Aircraft Conflicts

Md Siddiqur Rahman^(✉)

ENAC, IRIT UMR5505 CNRS, Univ. Toulouse Capitole, Univ. de Toulouse,
Toulouse, France
ronicse59@gmail.com

Abstract. When two or more airplanes find themselves less than a minimum distance apart on their trajectory, it is called a conflict situation. To solve a conflict, air traffic controllers use various types of information and decide on actions pilots have to apply on the fly. With the increase of the air traffic, the controllers' workload increases; making quick and accurate decisions is more and more complex for humans. Our research work aims at reducing the controllers' workload and help them in making the most appropriate decisions. More specifically, our PhD goal is to develop a model that learns the best possible action(s) to solve aircraft conflicts based on past decisions or examples. As the first steps in this work, we present a Conflict Resolution Deep Neural Network (CR-DNN) model as well as the evaluation framework we will follow to evaluate our model and a data set we developed for evaluation.

Keywords: Air traffic control · Aircraft conflict resolution · Machine learning · Deep Neural Network

1 Introduction

In the domain of air traffic, two or more planes are considered as in a conflict situation when their trajectories cross each other in certain circumstances of distance at the same time [13]. Air Traffic Management (ATM) has adopted some rules to avoid such conflicts [14] but the increasing density of aircraft flights makes conflict situations more and more difficult to anticipate and solve in an optimal way.

Decisions to solve conflicts are made manually in real-time and consist of changing aircraft trajectories to maintain a safe distance between planes. When a conflict is identified, the Air Traffic Controller (ATCO) has to make a quick decision about the best possible solution using his/her knowledge and experience. ATCOs have to take into account all the aircraft flight parameters such as its speed, positioning coordinate, destination, flight plan, its environment, weather, wind direction, military zone, etc. and the other flights. The air traffic growth

is so that the ATCOs will not be able to solve optimally conflicts in the future if they are not assisted effectively.

Some methods have been developed to solve the problems of aircraft conflict detection and resolution. James *et al.* provide a complete -although little aged now- overview of these approaches [13]. The first methods were mathematically grounded [18]; more recently, studies focus more on machine learning (ML) models [21], including deep learning based methods [6, 15].

Solving conflicts is a difficult problem because of the many types of information that have to be considered in real time. Many organizations keep data that could serve solving these challenges. The historical data basically includes aircraft information, initial flight plan, real trajectory information during the entire flight, immediate voice order from ATCOs to the pilot, and weather data. However there is no publicly available dataset that can be used by researchers.

In this work, we consider mid-range conflicts - that will occur around ten minutes after detection. In operational systems, controllers are automatically alerted in such cases; then s/he takes the decision on flight change and gives the action order to the pilot. Our PhD work aims at defining a model that learns the best possible action(s) to solve aircraft conflicts based on past decisions or examples and flight plans which is acknowledged to be the most important information in this context [10, 18]. In this paper, Sect. 2 is the related work, Sect. 3 introduces the model we will develop and the evaluation framework, and presents the dataset we developed, and Sect. 4 concludes this paper.

2 Related Work

The main solution for conflict resolution is to maintain a minimum distance between aircraft. In their earlier work, Zeghal and Karim [24] reviewed methods to solve conflicts that are based on a variation of the force fields approach. Warren applied performance analysis to compare fixed threshold conflict detection, covariance method conflict detection, and conformance bound conflict detection in three different situations for free routing conditions [23].

Until recently, mathematical modeling was the most commonly used method. Prandini *et al.* proposed a model for mid-range conflict detection [18]. They used different optimization techniques to minimize different resolution cost functions chosen for the random combination of flight plans in different scenarios with uncertainty [16]. Later, the same authors proposed two different models for both short-range and mid-range conflicts [18]. They compared their algorithm with the popular Center-TRACON Automation System algorithm from Erzberger *et al.* [8]. Eby and Kelly applied a distributed algorithm [7]. They consider flights where each aircraft could modify its flight plan; in reality, this is not allowed. Pham *et al.* [17] showed there are some limitations to use mathematical models. For instance, all the noise-free information related to the conflict scenario is required; otherwise the model is poor because of the uncertainty. The authors applied a rule-based reinforcement learning approach which will be too complex and probably impossible to write when considering all the parameters.

Classification methods are also applied to solve conflict situations such as mathematical models [3,18], Lagrangian models [5], Eulerian models [1]. More recently new ML methods have been applied for aircraft conflict resolution [2,6,11,17]. Jiang *et al.* use Support Vector Machine (SVM) for a binary classification of multi-aircraft conflicts in Free Flight [11]. They mainly consider the current position, velocity, and predicted look-ahead time as the main parameters. Kim *et al.* [12] proposed two separated models to solve a conflict between two airplanes: a neural network based and a SVM-based. The SVM model combines 9 SVM, one per class label. Similarly, the neural network model is composed of 9 nodes in the output layers. This model gives an output vector of 9 class labels that are all zero except the most probable one. Brittain and Wei in [6] applied an agent-based hierarchical deep reinforcement learning algorithm. Pham *et al.* applied a reinforcement learning for conflict resolution between two airplanes that are at the same height level [17]. Srinivasamurthy *et al.* used a semi-supervised model for the first time to predict controller immediate orders in [21]. As we mentioned earlier in this section, recent research work relies on agent-based reinforcement learning [6,17]. In that case, a reward function is used to search for solutions without using a pair of labeled input-output. In this work, we mainly focused on a complete supervised machine learning model. In order to meet our goal, we generated conflict scenarios and the resolution actions in heading change. Then, our supervised model can fit itself from this generated dataset. Since we know that no model has a 100% guarantee for resolving a conflict, our model not only will predict the best suggested resolution action but also provide multiple alternate resolution actions to the human agent.

3 Contributions in the PhD

A New Conflict Resolution Model: the CR-DNN Model

The main aim of our PhD work is to design and develop a ML model that will be able to make decisions in new conflict situations. Our model will predict the actions for any new scenario using different kinds of information that an ATCO takes into account although we will first focus on trajectory data.

The conflict resolution problem can be considered in several ways. It can be cast into a *ranking problem* where conflict resolution actions can be ranked; the top one being the most appropriate one. The advantage of this solution is that constraints can be easily added considering variables such as delay, proximity to destination, and flight time as used by Archibald *et al.* [4]. The problem can also be cast into a multi-class classification one, where each class would correspond to a heading change. We decided to opt for this second. The model will take a pair of aircraft trajectories causing a conflict situation as an input; the decisions and their binary (or graded) status will be the output. In this PhD work, we propose a deep neural network based model. Our study extends [12]' model to solve its limitation. Indeed, Kim's model has some limitations: (i) the conflict resolution is limited, (ii) input feature contains a single positioning coordinate of each aircraft, and (iii) output is only the best one.

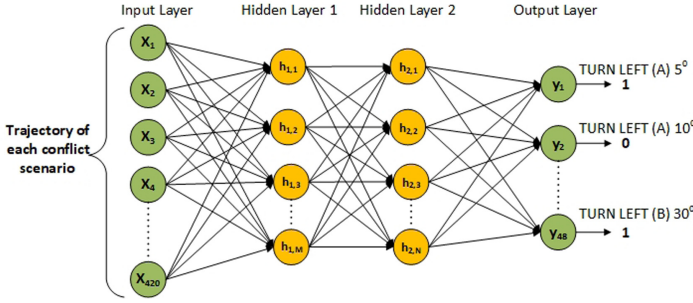


Fig. 1. Conflict Resolution Deep Neural Network (CR-DNN): The model predicts conflict resolution actions with binary decision from the 2 airplanes trajectories.

Figure 1 illustrates the model we propose named CR-DNN for Conflict Resolution Deep Neural Network. Our work extends [12] related work (a) to provide more specific heading change(s) (b) to rank possible solutions (c) to use 5 minute positioning coordinates for each aircraft. There are 7 features in an input trajectory, 3 for the two trajectories plus time stamp. Since we store the two airplanes location every 5 s, for 5 min we have 60 values; that makes 420 (7×60) input features. Therefore, the input layer of our model contains 420 nodes. The number of hidden layers and their number of nodes will be chosen using different parameter search algorithms. The output layer contains 48 nodes as the number of possible actions. For each action (node) the network will provide its binary decision (the training examples come with binary annotations) but later decisions will be graded considering thresholds using softmax function -we will use graded annotations for the examples. Thus, one of the strong point of the CR-DNN model is it will suggest optional multiple actions and their grade. The model aims at helping a human agent by providing ordered suggestions; the controller remains the one who will give her/his order to the pilot. In future architectures, we would also like to use an ensemble method that will integrate the above presented architecture and a model based on multiple SVMs.

Our model will be tested on the dataset of simulated data (see Sect. 3) since there is no publicly available data set. We will use the usual metrics from classification for performance analysis [20]: *Accuracy* (the number of correctly classified samples divided by the total number of samples), *Sensitivity* (the number of correctly classified positive examples over the total positive ones), *Specificity* (the number of correctly classified negative examples over the total number of negative ones), and *Mathew’s Correlation Coefficient* (the correlation coefficient between observation and prediction). During training, we will use the trajectory data associated with the two airplanes and the class label for all the possible actions. Once trained the model will be tested with pairs of unseen airplane trajectories; it will predict the labels of the different actions. We will use cross-fold validation method. All performance measures will be calculated for each fold and averaged. As a baseline, we will use Kim et al.’s model [12].

A New Trajectory Dataset From Simulator Each aircraft in the airspace is identified by the three basic coordinates latitude, longitude, and altitude; the fourth dimension being the time [22]. Eurocontrol, OpenSky Network, and Flightradar24 organizations give access to real trajectory data sets. OpenSky Network is one of the largest open-source aviation data source [19]. The collection includes a table with trajectories only. The problem to use such a data set is to synchronize the heading change with the conflicts. Another issue is that there is no information on the heading change if it is a conflict with the aircraft or not. Trajectory data is very sensible and for that reason, real data is generally kept as confidential. There is no simulated data publicly available either. Another contribution of this PhD thesis is thus to generate a data collection consisting of simulated trajectories. The advantages of simulated data are (a) we potentially can generate a lot of data (b) we can include a lot of variation in the data to study some specific cases, which is difficult to get when considering real data. To generate simulated data, we used the open-source python simulator named BlueSky and developed at TU Delft by Hoekstra and Ellerbroek [9]. Using this simulator, we made examples of conflict scenarios with ATCOs orders. An example corresponds to a conflict situation. It consists in (i) time in seconds: we simulated trajectory data that updates every 5 s (ii) aircraft A coordinates (latitude, longitude, altitude) (iii) aircraft B coordinates (iv) the controller action (v) a binary annotation (whether it can solve the conflict). For (iv) since there are many ways to solve a conflict, we consider the ones where the angle varies from 5° to 30° with a change of 5° . We created 122 trajectory samples with associated actions. The domain expert manually verified 41 scenarios (trajectory and associated action). The remaining 81 will be played in the simulator to generate trajectory as augmented data from those verified ones. The 122 samples will also be augmented considering small variations of the angle between two airplanes.

4 Conclusion

The purpose of our research is to create a novel conflict resolution model that will suggest ATCOs different heading modification decisions with priority to resolve the conflict between two airplanes. This paper presents the preliminary study regarding this objective that we will develop further during our PhD. We propose the CR-DNN model. It takes a pair of aircraft's trajectory as an input and predicts the resolution actions. While our first model will provide binary decisions, the second step will be to provide decision with grade to better help the ATCOs. We also plan to develop an ensemble model that will combine SVM and NN. This paper also describe the new data set we built with simulated trajectory data, conflicts and possible actions. This dataset contains a binary ground truth that will be enhanced into a graded one in a close future. We plan to evaluate the CR-DNN and compare it to related work [12]. Later we will compare the results with our ensemble model. We will expand the data collection and will make it available to the research community. In a more long term perspective, we will study the case where more than two airplanes are involved in a conflict. Finally, our research work will tackle the problem of ethics and diversion use.

Acknowledgments. This research work is conducted under the supervision of Josiane Mothe (IRIT, UMR5505 CNRS & INSPE, UT2J, Univ. de Toulouse) and Laurent Lapasset (DEVI, ENAC).

References

1. Agogino, A.K., Tumer, K.: A multiagent approach to managing air traffic flow. *Autonom. Agents Multi-Agent Sys.* **24**(1), 1–25 (2012)
2. Alam, S., Shafi, K., Abbass, H.A., Barlow, M.: An ensemble approach for conflict detection in free flight by data mining. *Transp. Res.* **17**(3), 298–317 (2009)
3. Alonso-Ayuso, A., Escudero, L.F., Olaso, P., Pizarro, C.: Conflict avoidance: 0–1 linear models for conflict detection & resolution. *Top* **21**(3), 485–504 (2013)
4. Archibald, J.K., et al.: A satisficing approach to aircraft conflict resolution. *IEEE Trans. Syst. Man Cybern.* **38**(4), 510–521 (2008)
5. Bayen, A., et al.: Lagrangian delay predictive model for sector-based air traffic flow. *J. Guidance Control Dyn.* **28**(5), 1015–1026 (2005)
6. Brittain, M., Wei, P.: Autonomous aircraft sequencing and separation with hierarchical deep reinforcement learning. In: *Conference for Research in Air Transport* (2018)
7. Eby, M.S., Kelly, W.E.: Free flight separation assurance using distributed algorithms. *IEEE Aerospace Conf.* **2**, 429–441 (1999)
8. Erzberger, H., Davis, T.J., Green, S.: Design of center-tracon automation system (1993)
9. Hoekstra, J.M., Ellerbroek, J.: Bluesky ATC simulator project: an open data and open source approach. In: *Conference on Research in Air Transport*, pp. 1–8 (2016)
10. Hu, J., et al.: Aircraft conflict prediction and resolution using brownian motion. *Conf. Decis Control.* **3**, 2438–2443 (1999)
11. Jiang, X.R., Wen, X.X., Wu, M.G., Wang, Z.K., Qiu, X.: A SVM approach of aircraft conflict detection in free flight. *J. Adv. Transport.* (2018)
12. Kim, K., Hwang, I., Yang, B.J.: Classification of conflict resolution methods using data-mining techniques. In: *AIAA*, p. 4075 (2016)
13. Kuchar, J.K., Yang, L.C.: A review of conflict detection and resolution modeling methods. *IEEE Trans. Intell. Transpor. Syst.* **1**(4), 179–189 (2000)
14. Mao, Z.H., Feron, E., Bilimoria, K.: Stability and performance of intersecting aircraft flows under decentralized conflict avoidance rules. *IEEE Trans. Intell. Transport. Syst.* **2**(2), 101–109 (2001)
15. Nanduri, A., Sherry, L.: Anomaly detection in aircraft data using RNN. In: *Integrated Communications Navigation and Surveillance*. pp. 5C2-1 (2016)
16. Paielli, R.A., Erzberger, H.: Conflict probability estimation for free flight. *J. Guidance Control Dyn.* **20**(3), 588–596 (1997)
17. Pham, D.T., Tran, N.P., Alam, S., Duong, V., Delahaye, D.: A machine learning approach for conflict resolution in dense traffic scenarios with uncertainties (2019)
18. Prandini, M., Hu, J., Lygeros, J., Sastry, S.: A probabilistic approach to aircraft conflict detection. *IEEE Trans. Intell. Transp. Syst.* **1**(4), 199–220 (2000)
19. Schäfer, M., et al.: Bringing up OpenSky: a large-scale ADS-B sensor network for research. In: *Symposium on Information Processing in Sensor Networks*, pp. 83–94 (2014)
20. Sokolova, M., Lapalme, G.: A systematic analysis of performance measures for classification tasks. *Inf. Process. Manage.* **45**(4), 427–437 (2009)

21. Srinivasamurthy, A., et al.: Iterative learning of speech recognition models for air traffic control. In: Interspeech, pp. 3519–3523 (2018)
22. Wandelt, S., Sun, X.: Efficient compression of 4D-trajectory data in air traffic management. *IEEE Trans. Intell. Transport. Syst.* **16**(2), 844–853 (2014)
23. Warren, A.: Medium term conflict detection for free routing: Operational concepts and requirements analysis. *Dig. Avionics Syst. Conf.* **2**, 3–9 (1997)
24. Zeghal, K.: A review of different approaches based on force fields for airborne conflict resolution. In: *Guidance, Navigation, and Control Conference*, p. 4240 (1998)



Handling Context in Data Quality Management

Flavia Serra^{1,2} 

¹ Universidad de la República, Montevideo, Uruguay
fserra@fing.edu.uy

² Université de Tours, Blois, France

Abstract. Data Quality Management (DQM) concerns a wide range of tasks and techniques, largely used by companies and organizations for assessing and improving the quality of their data. Data Quality (DQ) is defined as fitness for use and naturally depends on application context and usage needs. Moreover, context is embedded in DQM tasks, for example, in the definition of DQ metrics, in the discovery of DQ rules or in the elicitation of DQ requirements. However, despite its recognized importance for DQM, the literature only manages obvious contextual aspects of data, and lacks of proposals for context definition, specification and usage within major DQM tasks. This PhD thesis is at the junction of these two main topics: Data Quality and Context. Our objective is to model context for DQM, exploiting the contextual nature of data, at each phase of the DQM process. We aim to provide a general model of context for DQM, an approach for using the model within a DQM project, and a proof of concept in the domain of Digital Government.

Keywords: Data Quality · Context · Data Quality Management

1 Introduction

Data Quality (DQ) is a very wide research area that involves many different aspects, problems and challenges. The growing need to discover and integrate reliable information from heterogeneous data sources, distributed in the Web, Social Networks, Cloud platforms or Data Lakes, makes DQ an unavoidable topic, particularly hot in recent years (see e.g., [3, 9, 16, 17]). In addition, DQ has enormous relevance for the industry, due to its great impact on information systems in all application domains.

Furthermore, there are numerous public initiatives for ensuring DQ in Digital Government. Several countries (e.g. England [20], Estonia [22]) publish their data, to ensure the transparency of their public services. Open DQ is a particular challenge, because data from public services have special characteristics (e.g. mostly statistical, confidential, highly geo-referenced, often duplicated and contradictory). In addition, the users of these systems are very varied, from civil

servants (with different levels of training and responsibility) to citizens (users of applications). This PhD is related to a research project for the e-Government Agency and Information and Knowledge Society (AGESIC¹) in Uruguay.

DQ is defined as *fitness for use* and is widely recognized to be multidimensional [23]. Quality dimensions express the characteristics that data must have, such as their accuracy, completeness and consistency. In the literature, there is no agreement on the set of dimensions characterizing DQ nor on their meanings [1], however, most works agree that the choice of DQ dimensions depends on the type of application, on the task at hand, and on users' requirements. These aspects can be seen as the *context* in which DQ dimensions are defined and used in the different Data Quality Management (DQM) tasks. Early works classify DQ dimensions according to their dependency on context or not [21]. Recent works extend such dependency to current architectures, specially big data projects, evidencing the need for explicit context consideration [7] and showing that contextual DQ increases the retrieval of valuable information from data [15]. In addition, DQ problems are typically separated into context-dependent (e.g. violation of domain or business rules) and context-independent (e.g. incorrect values, spelling errors) [11]. Data are always *put in context* [4]. In their way from sources to users' applications, data are not used for a single purpose nor in a single analytical domain. Thus, data can be used in different contexts throughout each phase of the DQM process.

Usually, context is embedded in DQM tasks, for example when it is considered in the definition of DQ metrics [1, 6, 12], in the discovery of DQ rules or in the elicitation of DQ requirements [14]. In particular, a DQ model cannot provide fixed, all purpose DQ metrics, because they depend on specific data and their requirements [1]. Furthermore, DQ requirements depend on the specific context of use [8, 12, 14]. In turn, the role of DQ methodologies is to guide in the complex decisions to be made, and at the same time, methodologies have to be adapted to the application domain [1]. Indeed, specific methodologies for evaluating DQ have been applied in different domains. For example, for Digital Government, the work in [22] includes a DQ model and a DQM process for providing digital services to citizens and for the functioning of critical data infrastructure.

Despite its recognized importance for DQM, context is typically taken as an abstract concept, which is not clearly defined nor formalized. The literature lacks of a concise and globally accepted definition for context in DQ [18]. Indeed, some works claim that literature manages obvious contextual aspects of data, like geographic and temporal dimensions, and highlight that a general notion of context, its formalization and use in DQM have been missing so far [4]. A systematic literature review (presented in Sect. 2) evidenced this lack. Nevertheless, context definition and formalization are key tasks, essential for carrying out DQ projects, and being an independent asset that impacts many DQM tasks. For example, in DQ metrics definition, context management would provide with: (i) flexibility, since they could adapt to context variations, (ii) generality, since

¹ <https://www.gub.uy/agencia-gobierno-electronico-sociedad-informacion-conocimiento/>.

they could include many particular context-dependent cases, and (iii) richness, leading to include more aspects to the metric.

The goal of this PhD project is to model context for DQM, considering context at each phase of the DQM process. We aim to provide a general model of context for DQ, an approach for using the model within a DQM project, and a proof of concept within a real case study in the domain of Digital Government.

The rest of the document is organized as follows: Sect. 2 summarizes the results of a Systematic Literature Review on the topic, and Sect. 3 describes the PhD thesis proposal and highlights future incomes.

2 Systematic Literature Review

The first stage of the PhD project was to carry out a Systematic Literature Review (SLR). This kind of methodology allows us to identify, evaluate and interpret all available research relevant to a particular research question or topic area. In particular, we apply a SLR to know current evidence in the relationship between *Context* and *Data Quality* areas. In short, we wanted to identify any gap at the intersection of these areas in order to detect new research lines. The research questions asked are the following: *How is context used in data quality models? How is context used within quality metrics for the main data quality dimensions? How is context used within data quality concepts?*

Based on our research questions, we defined several search strings that were run in the search engines of 5 digital libraries: ACM digital library, IEEE Xplore, Science Direct, Springer LNCS and Google Scholar, the latter restricted to 3 open source venues. We asked for articles and book chapters, in the period 2010–2019, written in English and published in PDF format. We obtained a total of 2132 scientific works, called primary studies (PS for short).

The SLR methodology allowed for the selection of 43 PS based on their relevance, that were analyzed with a particular attention on the definition of context in DQM. A first observation concerns the lack of works formalizing context: only 5 works propose formal definitions of context. Concerning types of works, most PS, and in particular the ones formalizing context, propose models, frameworks and methodologies for DQM. Some research domains appear in many PS, in particular Big Data and Business Intelligence, but most PS deal with DQ in a general way. Noticeably, most proposals are specific to structured data. Finally, we remark that the number and quality² of PS increased from 2016. Indeed, PS published in or after 2016 represent 75% of the selected PS and 78% of the PS published in venues ranked A* or A.

The main findings of the review are the following. Context is included in DQ models by two main ways: distinguishing contextual DQ dimensions from non-contextual ones, or considering specific context through DQ requirements. Regarding quality metrics, context is included mainly through DQ requirements,

² According to Scopus (<https://www.scopus.com/sources>) and Core (<http://portal.core.edu.au/conf-ranks>) rankings (accessed March 2020).

user preferences and domain-specific rules. Context is used mainly for measurement tasks, into DQ methodologies, through DQ requirements and related to DQ problems. Interestingly, most works consider context within DQ requirements in several phases of DQM, exploiting the subjective nature of DQ [15].

On the other hand, we note that while the importance of context in DQM is acknowledged in all the studied works, only few of them present a formal definition of context. The formalisms are varied (relational schemas, ontologies, SKOS concepts, predicates, entities) and mostly specific to structured data, showing that a consensual definition is yet to come.

The results of the SLR show that research work is still needed to have a clear and consensual understanding of context in DQM. Major open research questions include: (Q_1) how to define context, accounting for variety in data format and DQ related tasks, in particular quality assessment, (Q_2) how context should be included in DQ methodological aspects, and (Q_3) how context should be formalized.

3 The PhD Project

Objective: Our objective is to model context for DQM, by exploiting the contextual nature of data at each phase of the DQM process. Hence, based on the state of the art and the open research questions raised by the SLR, we draw the following research problems: (P1) which components should be included in the definition of context for DQM and, (P2) how context should be included in each phase of a DQM process.

Context Components (P1): The state of the art revealed that although there are general operational definitions of context and context-aware computing [10], context representation is neglected in DQM. In particular, we highlight conclusions of Bertossi et al. [4] who report that the literature only deals with obvious contextual aspects of data, like geographic and temporal dimensions. As suggested by Bolchini et al. [5], other contextual aspects should be specified, e.g. users (person, application, device, etc.), presentations (system capabilities to adapt content presentation to different channels/devices), communities (set of relevant variables shared by a group of peers), and data tailoring (only selecting relevant data, functionalities and services). Preferences, documents content, DQ requirements and domain rules also emerge as important components (a preliminary review can be found in [19]).

We consider that context not only fits a single perspective, but could be defined by elements taken from different perspectives (user, presentation, data tailoring, etc.). Therefore, and supported by open research question Q_1 raised by the SLR, we set out our first problem: **to review which components should be included in the definition of context for DQM.** For tackling this problem, we will review the most important context definitions in Pervasive Computing research area, where the context has an important role [13]. In this stage of the PhD thesis, we do not focus on what we want to contextualize

(i.e. the DQM processes). Indeed, we start by eliciting the components (e.g. user task, domain rules, DQ requirements, etc.) that should be included in the context definition. For example, in [2], authors argue that context could be analyzed through six essential components: constraint, influence, behavior, nature, structure and system. That is, the context acts like a set of constraints that influence the behavior of a system (a user or a computer) embedded in a given task.

Context in DQM Process (P2): Batini and Scannapieco [1] present and discuss a diversity of DQ methodologies proposed in the literature for DQM. While these methodologies agree in the classical phases of DQM processes, such as elicitation of DQ requirements, DQ assessment, data analysis and DQ improvement, methodologies could be adapted to the specific application domain [1]. In this PhD thesis we will use (and possibly adapt) a DQM process developed for AGESIC. It has seven phases involving scenario characterization, choosing target data, and definition of a DQM strategy, with support for typical Digital Government scenarios.

In addition, we believe that context changes at each phase of the DQM process, because each phase is influenced by different context components. For example, domain rules capture context during requirement elicitation, while context is embedded in subjective metrics during DQ assessment. The SLR showed that few works use context when performing DQM tasks (data profiling, data cleaning, data evaluation, etc.), DQ measurement tasks being the ones that most use context.

Therefore, and supported by open research question Q_2 raised by the SLR, we set out our second problem: **how context should be included in each phases of a DQM process**. By solving this problem, we want to solve which are the context components that are involved in each phase of the DQM process.

Once all the important components of the context have been identified for each phase of the DQM process, context can be formalized (supported by open research question Q_3). Actually, context formalization starts within P1, i.e. specifying context components, and continues while solving P2, by also formalizing their instantiation to the phases of the DQM process. We expect to obtain a general framework, relating all context-related elements, and helping users in the application of DQ methodologies enriched by the context.

Evaluation: We intend to test our proposal within the DQM process defined for Digital Government, instantiating our context definition in each phase of this process. Furthermore, a complete case study will be developed within AGESIC environment (data, involved users, questionnaires, domain rules, DQ requirements, processes, etc.). The evaluation protocol will be defined in accordance with AGESIC representatives.

Organization: This thesis started in September 2019, as a Uruguayan-French co-supervised project. The first step, the SLR, was conducted during the first semester. The remaining of the first year will be devoted to address problem P1. Problem P2 will be studied along the second year, the first quarter overlapping

components specification and DQ methodologies. The evaluation protocol will be set early during second year, but evaluation will be developed during third year. We expect to start the writing of the manuscript by January 2022.

References

1. Batini, C., Scannapieco, M.: *Data and Information Quality - Dimensions*. Springer, Principles and Techniques (2016). <https://doi.org/10.1007/978-3-319-24106-7>
2. Bazire, M., Brézillon, P.: Understanding context before using it. In: *CONTEXT* (2005)
3. Bertossi, L.: Database repairs and consistent query answering. In: *PODS* (2019)
4. Bertossi, L., Rizzolo, F., Jiang, L.: Data Quality is context dependent. In: *BIRTE* (2011)
5. Bolchini, C., Curino, C., Orsi, G., Quintarelli, E., Rossato, R., Schreiber, F.A., Tanca, L.: And what can context do for data? *CACM* **52**(11), 136–140 (2009)
6. Bors, C., Gschwandtner, T., Kriglstein, S., Miksch, S., Pohl, M.: Visualinteractive creation customization and analysis of data quality metrics. *JDIQ* **10**(1), 1 (2018)
7. Caballero, I., Serrano, M., Piattini, M.: A data quality in use model for big data. In: *MoBiD (ER workshops)* (2014)
8. Cappiello, C., Samá, W., Vitali, M.: Quality awareness for a successful big data exploitation. In: *IDEAS* (2018)
9. Chu, X., Ilyas, I., Krishnan, S., Wang, J.: Data cleaning: overview and emerging challenges. In: *SIGMOD* (2016)
10. Dey, A.K.: Understanding and using context. *PUC* **5**(1), 5 (2001)
11. Foidl, H., Felderer, M.: Risk-based data validation in machine learning-based software systems. In: *MaLTesQuE (ACM SIGSOFT Workshops)* (2019)
12. Heinrich, B., Hristova, D., Klier, M., Schiller, A., Szubartowicz, M.: Requirements for data quality metrics. *JDIQ* **9**(2), 12 (2018)
13. Henriksen, K., Indulska, J., Rakotonirainy, A.: Modeling context information in pervasive computing systems. In: *Pervasive* (2002)
14. Marotta, A., Vaisman, A.: Rule-based multidimensional data quality assessment using contexts. In: *DaWaK* (2016)
15. McNab, A.L., Ladd, D.A.: Information quality: The importance of context and trade-offs. In: *HICSS* (2014)
16. Pena, E., Almeida, E.C.d., Naumann, F.: Discovery of approximate (and exact) denial constraints. *PVLDB* **13**(3), 92–108 (2019)
17. Sadiq, S., et al.: Data quality: the role of empiricism. *ACM SIGMOD Record* **46**(4), 40 (2018)
18. Serra, F., Marotta, A.: Data warehouse quality assessment using contexts. In: *WISE* (2016)
19. Serra, F., Marotta, A.: Context-based data quality metrics in data warehouse systems. *CLEI Electron. J.* **20**(2), 2–38 (2017)
20. Statistics and Regulatory Data Division: *DQ Framework*. Bank of England. <https://www.bankofengland.co.uk/-/media/boe/files/statistics/data-quality-framework.pdf> (2014). Accessed Apr 2020
21. Strong, D.M., Lee, Y.W., Wang, R.Y.: Data quality in context. *CACM* **40**(5), 1 (1997)
22. Tepandi, J., et al.: The data quality framework for the estonian public sector and its evaluation. *TLDKS* **35**, 1–26 (2017)
23. Wang, R.Y., Strong, D.M.: Beyond accuracy: What data quality means to data consumers. *J. Manage. Inf. Syst.* **12**(4) (1996)

Author Index

- Alatrística-Salas, Hugo 49
Amavi, Joshua 36
Andres, Frederic 122
Angioni, Simone 219
Arani, Zachary 237
Aryani, Amir 195
Athanasίου, Spiros 335
- Barbon Jr., Sylvio 296
Basiuk, Taras 237
Bellatreche, Ladjel 3
Bentayeb, Fadila 3
Białas, Paweł 158
Bieliková, Mária 3
Boussaid, Omar 3
Brzeski, Adam 169
Brzeziński, Jakub 158
- Capelo, Piero L. 85
Catania, Barbara 3
Ceravolo, Paolo 3
Chapman, Drake 237
Chatzopoulos, Serafeim 323
Chialva, Diego 183
Chronis, Pantelis 335
Ciferri, Cristina D. A. 85
Costa, Umberto 72
Cychnerski, Jan 169
- d'Orazio, Laurent 122, 237
Dalamagas, Theodore 323
de Toledo, Damien Alvarez 122
Delibasic, Boris 131
Demidova, Elena 3
Diesner, Jana 207
Draszawka, Karol 169
Duarte, Mariana M. G. 249
Dziubich, Krystyna 169
Dziubich, Tomasz 158, 169
- Eidizadehakhcheloo, Sanaz 259
Espinoso-Oviedo, Javier A. 271
- Fenner, Martin 195
- Gagnon, Michel 23
Gómez-López, María Teresa 283
Grigorev, Semyon 72
Gruenwald, Le 237
- Hajdu, László 145
Halfeld Ferrari, Mirian 3, 36
Halman, Joanna 158
Han, Kanyao 207
Hara, Carmem S. 3, 249
Hiot, Nicolas 36
- Imine, Abdessamad 259
- Kanellos, Ilias 323
Katsaros, Dimitrios 347
Kordić, Slavica 3
Korčub, Waldemar 169
Kovacevic, Ana 131
Krész, Miklós 145
- Lan, Michael 311
Laurent, Anne 110
Leite, Maria C. A. 122
Lempeis, Antonis 341
Lin, Jian 98
Lopez, Maria Teresa Gomez 3
Luković, Ivan 3
- Magalhães, Dimmy 60
Manghi, Paolo 3, 195, 341
Mannocci, Andrea 3, 195
Manola, Natalia 341
Manolopoulos, Yannis 347
Marchand, Erwan 23
Martziou, Stefania 341
Medeiros, Ciro 72
Meimaris, Marios 341
Melgar, Andrés 49
Mishra, Shubhanshu 207

- Motta, Enrico 219
Mugabushaka, Alexis-Michel 183
Musicante, Martin A. 72
- Oncevay, Arturo 49
Orero, Joseph Onderi 110
Osborne, Francesco 3, 219
Owuor, Dickson Odhiambo 110
- Papadopoulou, Elli 341
Papastefanatos, George 341
Papatheodorou, Christos 3
Parody, Luisa 283
Pijani, Bizhan Alipour 259
Pozo, Aurora 60
- Radovanovic, Sandro 131
Rahman, Md Siddiqur 355
Ramos-Gutiérrez, Belén 283
Recupero, Diego Reforgiato 219
Ristić, Sonja 3
Rocha, Guilherme M. 85
Romero, Oscar 3
Rościszewski, Paweł 169
Rusinowitch, Michaël 259
- Sacharidis, Dimitris 3
Salatino, Angelo A. 3, 219
Schroeder, Rebeca 249
Serra, Flavia 362
Sidiropoulos, Antonis 347
- Skiadopoulos, Spiros 335
Skoutas, Dimitrios 311, 335
Sobrevilla, Marco 49
Stocker, Markus 195
Stoupas, Georgios 347
- Talens, Guilaine 3
Tavares, Gabriel Marques 296
Theodoratos, Dimitri 311
Tryfonopoulos, Christos 323
- van Keulen, Maurice 3
Vargas-Solar, Genoveva 271
Vergoulis, Thanasis 3, 323
Vukicevic, Milan 131
- Wang, Chenxiao 237
Weber, Tobias 226
Wu, Xiaoying 311
- Xie, Dongming 98
- Yang, Pingjing 207
- Zechinelli-Martini, José-Luis 271
Znanięcki, Łukasz 158
Zouaq, Amal 23
Zumer, Maja 3