Joint Proceedings of

PROFILES 2019

The 6th International Workshop on Dataset Profiling and Search

Elena Demidova, Stefan Dietze, John Breslin and Simon Gottschalk

&

SEMEX 2019

The 1st Workshop on Semantic Explainability

Philipp Cimiano, Basil Ell, Agnieszka Lawrynowicz,
Laura Moss and Axel-Cyrille Ngonga Ngomo

co-located with
The 18th International Semantic Web Conference
(ISWC 2019)

Volume Editors

PROFILES 2019

Elena Demidova
L3S Research Center, Germany
demidova@L3S.de

Stefan Dietze
GESIS Leibniz Institute for the Social Sciences, Germany
stefan.dietze@gesis.org

John Breslin
National University of Ireland Galway, Ireland
john.breslin@nuigalway.ie

Simon Gottschalk
L3S Research Center, Germany
gottschalk@L3S.de

SEMEX 2019

Philipp Cimiano
Bielefeld University, Germany
cimiano@cit-ec.uni-bielefeld.de

Basil Ell
Bielefeld University, Germany and Oslo University, Norway
bell@techfak.uni-bielefeld.de

Agnieszka Lawrynowicz
Poznan University of Technology, Poland
agnieszka.lawrynowicz@put.poznan.pl

Laura Moss
University of Glasgow, UK
Laura.Moss@glasgow.ac.uk

Axel-Cyrille Ngonga Ngomo
Paderborn University, Germany
axel.ngonga@upb.de

Preface

This joint volume of proceedings gathers papers from the 6th International Workshop on Dataset Profiling and Search (PROFILES 2019) and the 1st Workshop on Semantic Explainability (SEMEX 2019), both held on October 27, 2019 during the 18th International Semantic Web Conference (ISWC 2019) in Auckland, New Zealand. While the PROFILES 2019 workshop focused on dataset profiling and search, the SEMEX 2019 workshop targeted semantic explainability.

PROFILES 2019. The Web of Data has seen tremendous growth recently. New forms of structured data have emerged in the form of knowledge graphs, Web markup, such as schema.org, as well as entity-centric data in Web tables. Considering these rich, heterogeneous and evolving data sources which cover a wide variety of domains, exploitation of Web Data becomes increasingly important in the context of various applications, including dataset search, question answering and fact verification. These applications require reliable information on dataset characteristics, including general metadata, quality features, statistical information, dynamics, licensing, and provenance. Lack of a thorough understanding of the nature, scope and characteristics of data from particular sources limits their take-up and reuse, such that applications are often limited and focused on well-known reference datasets. The PROFILES workshop series started in 2014 and has since then offered a highly interactive forum for researchers and practitioners, bringing together experts in the fields of the Web, Semantic Web, Web Data, Semantic Search, Databases, NLP, IR, and application domains, to discuss such challenges and identify synergies for joint initiatives.

The contributions of the papers accepted at PROFILES 2019 include new technologies for dataset profiling, specifically for the generation of descriptive datasets snippets, the provision of data with license annotations, and the automatic classification of Linked Open Data vocabularies. Such dataset profiles do not only enable fine-grained dataset search, but are also valuable resources for the configuration of data analytics workflows and knowledge mining, illustrated by the two invited talks.

SEMEX 2019. In recent years, the explainability of complex systems such as decision support systems, automatic decision systems, machine learning-based/trained systems, and artificial intelligence in general has been expressed not only as a desired property, but also as a property that is required by law. For example, the General Data Protection Regulation's (GDPR) „right to explanation" demands that the results of ML/AI-based decisions are explained. The explainability of complex systems, especially of ML-based and AI-based systems, becomes increasingly relevant as more and more aspects of our lives are influenced by these systems' actions and decisions.

Several workshops address the problem of explainable AI. However, none of these workshops has a focus on semantic technologies such as ontologies and reasoning. We believe that semantic technologies and explainability coalesce in two ways. First, systems that are based on semantic technologies must be explainable like all other AI systems. In addition, semantic technology seems predestined to support in rendering explainable those systems that are not themselves based on semantic technologies.

This workshop aims to bring together international experts interested in the application of semantic technologies for explainability of artificial intelligence/machine learning to stimulate research, engineering and evaluation – towards making machine decisions transparent, re-traceable, comprehensible, interpretable, explainable, and reproducible. Semantic technologies have the potential to play an important role in the field of explainability since they lend themselves very well to the task, as they enable to model users' conceptualizations of the problem domain. However, this field has so far only been only rarely explored.

The papers accepted to SEMEX 2019 include a systematic literature review that presents current approaches of combining Machine Learning with Semantic Web Technologies in the context of model explainability; an approach that makes the structure of a natural language argument and the background knowledge the argument is built on explicit; an interactive method to build a probabilistic relational model from any given domain represented by a knowledge graph; and an approach that verbalizes the inconsistencies identified by a reasoner so that users can be persuaded to change unhealthy behaviour if they do not follow dietary rules to manage their diseases. Furthermore, Freddy Lecue will give an invited talk about the role of knowledge graphs in explainable AI;

We would like to take this opportunity to sincerely thank the authors for their invaluable and inspiring contributions to the workshops. Our sincere thanks are given to the program committee members for reviewing the submissions and thereby assuring the high quality of the workshop program. We are also very grateful to the organisers of the ISWC 2019 conference and in particular to the Workshops & Tutorials Chairs Sofia Pinto and Hideaki Takeda for their support in the workshop organisation.

October 2019

Elena Demidova
Stefan Dietze
John Breslin
Simon Gottschalk
Philipp Cimiano
Basil Ell
Agnieszka Lawrynowicz
Laura Moss
Axel-Cyrille Ngonga Ngomo

The organisation of the PROFILES 2019 workshop was partially funded by the Federal Ministry of Education and Research (BMBF), Germany under Data4UrbanMobility (02K15A040) and Simple-ML (01IS18054), and by Science Foundation Ireland (SFI) under Grant Numbers SFI/12/RC/2289_P2 and SFI/16/RC/3918, co-funded by the European Regional Development Fund.

The SEMEX workshop has been funded by the Deutsche Forschungsgemeinschaft (DFG) within the Priority Program "Robust Argumentation Machines (RATIO)" (SPP-1999).

Organization (PROFILES 2019)

Organizing Committee

Elena Demidova – L3S Research Center, Germany
Stefan Dietze – GESIS Leibniz Institute for the Social Sciences, Germany
John Breslin – National University of Ireland Galway, Ireland
Simon Gottschalk – L3S Research Center, Germany

Program Committee

Charlie Abela – University of Malta, Malta
Enrico Daga – The Open University, UK
Liubov Kovriguina – NRU ITMO, Russia
Joanna Lytra – University of Bonn, Germany
Dmitry Mouromtsev – NRU ITMO, Russia
Bernardo Pereira Nunes – PUC-Rio, Brazil
Anisa Rula – University of Milano–Bicocca, Italy
Nicolas Tempelmeier – L3S Research Center, Germany
Konstantin Todorov – University of Montpellier, France
Raquel Trillo-Lado – Universidad de Zaragoza, Spain
Maria Esther Vidal – Leibniz Information Centre For Science and Technology, Germany
Ran Yu – GESIS Leibniz Institute for the Social Sciences, Germany
Amrapali Zaveri – Maastricht University, The Netherlands

Organization (SEMEX 2019)

Organizing Committee

Philipp Cimiano – Bielefeld University, Germany
Basil Ell – Bielefeld University, Oslo University, Norway
Agnieszka Lawrynowicz – Poznan University of Technology, Poland
Laura Moss – University of Glasgow, UK
Axel-Cyrille Ngonga Ngomo – Paderborn University, Germany

Program Committee

Ahmet Soylu – Norwegian University of Science and Technology / SINTEF Digital, Norway
Amrapali Zaveri – Maastricht University, Netherlands
Andreas Harth – Fraunhofer IIS, Germany
Anisa Rula – University of Milano – Bicocca, Italy
Axel-Cyrille Ngonga Ngomo – Paderborn University, Germany
Axel Polleres – Wirtschaftsuniversität Wien, Austria
Basil Ell – Bielefeld University, Germany and University of Oslo, Norway
Benno Stein – Bauhaus-Universität Weimar, Germany
Christos Dimitrakakis – Chalmers University of Technology, Sweden
Ernesto Jimenez-Ruiz – The Alan Turing Institute, UK
Evgenij Thorstensen – University of Oslo, Norway
Francesco Osborne – The Open University, UK
Gong Cheng – Nanjing University, China
Heiner Stuckenschmidt – University of Mannheim, Germany
Jürgen Ziegler – University of Duisburg-Essen, Germany
Mariano Rico – Universidad Politécnica de Madrid, Spain
Maribel Acosta – Karlsruhe Institute of Technology, Germany
Martin G. Skjæveland – University of Oslo, Norway
Mathieu d'Aquin – National University of Ireland Galway, Ireland
Menna El-Assady – University of Konstanz, Germany
Michael Kohlhase – Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
Pascal Hitzler – Wright State University, USA
Philipp Cimiano – Bielefeld University, Germany
Ralf Schenkel – Trier University, Germany
Serena Villata – Université Côte d'Azur, CNRS, Inria, I3S, France
Stefan Schlobach – Vrije Universiteit Amsterdam, The Netherlands
Steffen Staab – University of Koblenz-Landau, Germany

Contents

Towards Multi-Facet Snippets for Dataset Search

Xiaxia Wang[1], Gong Cheng[1], and Evgeny Kharlamov[2,3]

[1] National Key Laboratory for Novel Software Technology, Nanjing University, China
xxwang@smail.nju.edu.cn, gcheng@nju.edu.cn
[2] Department of Informatics, University of Oslo, Norway
evgeny.kharlamov@ifi.uio.no
[3] Bosch Center for Artificial Intelligence, Renningen, Germany
evgeny.kharlamov@de.bosch.com

Abstract. Due to a recent significant increase in the number of RDF datasets available on the Web, there is a pressing need in effective search techniques for finding the right data on demand. A promising approach is to present retrieved datasets as snippets that aim at concisely explaining to the user why this dataset fulfils their demand. Snippets in particular can illustrate the main content of the dataset and explain its relevance to the user's query. Computing optimal snippets is a non-trivial task and a number of approaches have emerged to address this problem. In this short paper, we report our ongoing work on snippets that address multiple facets of optimality. Based on our recently proposed evaluation metrics for dataset snippets, we formulate a weighted maximum coverage problem which directly optimizes three evaluation metrics. We solve the problem with a greedy algorithm, and our current implementation has outperformed four baseline methods.

1 Introduction

The open data movement brings increasingly many datasets to the Web, many of which are in the RDF format. Reusing these datasets is of great importance to researchers and developers. In order to enable the reuse there is a pressing need in effective search techniques for finding the *right* data on demand. A promising approach is to query for datasets with keywords as in Google Dataset Search [1] and to present each retrieved RDF dataset as a *snippet*, its small representative subset [2]. Dataset snippets aim at concisely explaining to the user *why* this dataset fulfils their demand and in particular can illustrate the main content of the dataset and explain its relevance to the user's query.

Computing optimal snippets is a non-trivial task and a number of approaches have emerged to address this or related problems [2–6]. In [7], we presented four metrics for evaluating the quality of a dataset snippet. In this short paper, we report our ongoing work on snippets that address multiple facets of optimality. In particular, in order to improve the quality of a snippet for dataset search, we

formulate the selection of RDF triples as a combinatorial optimization problem that directly optimizes three evaluation metrics proposed in [7]. A dataset snippet generated by our approach, which we refer to as **KSD**, is expected to have a good coverage of the query **K**eywords and the content of the dataset at both the **S**chema and the **D**ata level. We solve the problem with a greedy algorithm, and our evaluation demonstrates that KSD outperforms the baselines reported in [7] and that there is still a considerable room for quality improvement.

The remainder of this paper is structured as follows. Section 2 defines the problem and reviews the evaluation metrics proposed in [7]. Section 3 describes the implementation of KSD. Section 4 presents evaluation results. Section 5 concludes the paper with future work.

2 Preliminaries

2.1 Problem Statement

An RDF dataset is a set of RDF triples denoted by $T = \{t_1, t_2, \cdots, t_n\}$, where each $t_i = \langle t_i^s, t_i^p, t_i^o \rangle$ is a subject-predicate-object triple of RDF resources. The subject t_i^s of a triple t_i is an entity (i.e., a non-literal resource at the instance level) that appears in T. The predicate t_i^p represents a property. The object t_i^o is a value of t_i^p, which can be a class, a literal, or another entity in T.

A keyword query is a set of keywords denoted by $Q = \{q_1, q_2, \cdots, q_m\}$. Given a dataset T, a keyword query Q, and a positive integer k as the size bound, a *dataset snippet* is an optimum subset of triples selected from T, denoted by $S \subseteq T$, satisfying $|S| \leq k$. We will give our definition of optimality in Section 3.

2.2 Evaluation Metrics

We briefly review the four metrics proposed in [7] for evaluating the quality of a dataset snippet S: `coKyw`, `coCnx`, `coSkm`, and `coDat`, all in the range of $[0, 1]$.

Coverage of Query Keywords (`coKyw`). A resource r covers a keyword q if r's textual form (e.g., `rdfs:label` of an IRI or blank node, lexical form of a literal) contains a keyword match for q. A triple t covers a keyword q, denoted by $t \prec q$, if r covers q for any $r \in \{t^{\mathrm{s}}, t^{\mathrm{p}}, t^{\mathrm{o}}\}$. For a snippet S, the `coKyw` metric evaluates its coverage of query keywords:

$$\texttt{coKyw}(S) = \frac{1}{|Q|} \cdot |\{q \in Q : \exists t \in S,\ t \prec q\}| \,. \tag{1}$$

Coverage of Connections between Query Keywords (`coCnx`). A snippet S covers the connection between two keywords $q_i, q_j \in Q$, denoted by $S \prec (q_i, q_j)$, if there is a path in the RDF graph representation of S that connects two resources: one covering q_i and the other covering q_j. For S, the `coCnx` metric evaluates its coverage of connections between query keywords:

$$\texttt{coCnx}(S) = \begin{cases} \frac{1}{\binom{|Q|}{2}} \cdot |\{\{q_i, q_j\} \subseteq Q : q_i \neq q_j \text{ and } S \prec (q_i, q_j)\}| & \text{if } |Q| > 1\,, \\ \texttt{coKyw}(S) & \text{if } |Q| = 1\,. \end{cases} \tag{2}$$

When there is only one keyword, coCnx is meaningless and we set it to coKyw.

Coverage of Data Schema (coSkm). Consider the RDF schema of a dataset. The relative frequency of a class c observed in a dataset T is

$$\texttt{frqCls}(c) = \frac{|\{t \in T : \ t^{\text{p}} = \texttt{rdf:type} \text{ and } t^{\text{o}} = c\}|}{|\{t \in T : \ t^{\text{p}} = \texttt{rdf:type}\}|}\,. \tag{3}$$

Analogously, the relative frequency of a property p observed in T is

$$\texttt{frqPrp}(p) = \frac{|\{t \in T : \ t^{\text{p}} = p\}|}{|T|}\,. \tag{4}$$

For a snippet S, its coverage of the schema of T is the harmonic mean (hm) of the total relative frequency of the classes and properties it contains:

$$\texttt{coSkm}(S) = \texttt{hm}(\sum_{c \in \texttt{Cls}(S)} \texttt{frqCls}(c), \sum_{p \in \texttt{Prp}(S)} \texttt{frqPrp}(p))\,, \tag{5}$$

where $\texttt{Cls}(S)$ is the set of classes instantiated in S and $\texttt{Prp}(S)$ is the set of properties instantiated in S.

Coverage of Data (coDat). Central entities represent the key content of a dataset. Let $\texttt{d}^+(e)$ and $\texttt{d}^-(e)$ be the out-degree and in-degree of an entity e in the RDF graph representation of a dataset T, respectively. For a snippet S, its coverage of the entities in T is the harmonic mean (hm) of the mean normalized out-degree and in-degree of the entities it contains:

$$\begin{aligned}\texttt{coDat}(S) = \texttt{hm}(&\frac{1}{|\texttt{Ent}(S)|} \cdot \sum_{e \in \texttt{Ent}(S)} \frac{\log(\texttt{d}^+(e)+1)}{\max_{e' \in \texttt{Ent}(T)} \log(\texttt{d}^+(e')+1)}, \\ &\frac{1}{|\texttt{Ent}(S)|} \cdot \sum_{e \in \texttt{Ent}(S)} \frac{\log(\texttt{d}^-(e)+1)}{\max_{e' \in \texttt{Ent}(T)} \log(\texttt{d}^-(e')+1)})\,,\end{aligned} \tag{6}$$

where $\texttt{Ent}(X)$ is the set of entities that appear in a set of triples X.

3 Approach

Given the evaluation metrics presented in Section 2.2, a straightforward idea is to formulate the selection of RDF triples as a combinatorial optimization problem, and directly optimize these evaluation metrics. Our current work considers three metrics: coKyw, coSkm, and coDat, leaving coCnx as future work. The three selected metrics all require a snippet to cover some elements: query keywords in coKyw, classes and properties in coSkm, and entities in coDat. Furthermore, the classes, properties, and entities to cover are with different weights. It inspires us to formulate an instance of the weighted maximum coverage problem. We formalize this idea in Section 3.1, and present a solution in Section 3.2.

Algorithm 1 Greedy Algorithm

Input: A dataset T, a keyword query Q, and a size bound k
Output: An optimum dataset snippet $S \subseteq T$

$S \leftarrow \emptyset$;
while $|S| < k$ **do**
  $t^* \leftarrow \texttt{argmax}_{t \in (T \setminus S)}(\mathtt{q}(S \cup \{t\}) - \mathtt{q}(S))$;
  $S \leftarrow S \cup \{t^*\}$;
end while
return S;

3.1 Snippet Generation as Weighted Maximum Coverage

Weighted Maximum Coverage. Given a collection of sets, a weighted maximum coverage (WMC) problem is to select a limited number of sets from the collection such that the total weight of the covered elements is maximized.

Snippet Generation as WMC. We formulate the generation of an optimum dataset snippet as an instance of the WMC problem. Each RDF triple $t_i \in T$ corresponds to a set denoted by $\mathtt{cov}(t_i)$ which consists of: the query keywords covered by t_i, the class instantiated in t_i, the property instantiated in t_i, and the entities that appear in t_i. The universe of elements is denoted by

$$\Omega = Q \cup \mathtt{Cls}(T) \cup \mathtt{Prp}(T) \cup \mathtt{Ent}(T)\,. \tag{7}$$

Each element $x \in \Omega$ has a non-negative weight:

$$\mathtt{w}(x) = \begin{cases} \alpha \cdot \frac{1}{|Q|} & x \in Q\,, \\ \beta \cdot \mathtt{frqCls}(x) & x \in \mathtt{Cls}(T)\,, \\ \beta \cdot \mathtt{frqPrp}(x) & x \in \mathtt{Prp}(T)\,, \\ \gamma \cdot \left(\frac{\log(\mathtt{d}^+(x)+1)}{\sum_{e \in \mathtt{Ent}(T)} \log(\mathtt{d}^+(e)+1)} + \frac{\log(\mathtt{d}^-(x)+1)}{\sum_{e \in \mathtt{Ent}(T)} \log(\mathtt{d}^-(e)+1)}\right) & x \in \mathtt{Ent}(T)\,, \end{cases} \tag{8}$$

In our experiments, we set $\alpha = 2, \beta = 1, \gamma = 1$, to balance between the coverage of query keywords in $\mathtt{coKyw}$ (α), the coverage of classes and properties in $\mathtt{coSkm}$ (β), and the coverage of entities in $\mathtt{coDat}$ (γ) in our objective function.

An optimum dataset snippet $S \subseteq T$ is one that

$$\text{maximizes } \mathtt{q}(S) = \sum_{x \in \bigcup_{t_i \in S} \mathtt{cov}(t_i)} \mathtt{w}(x)\,, \quad \text{subject to } |S| \leq k\,, \tag{9}$$

where k is a predefined size bound, and $\mathtt{q}(\cdot)$ is the objective function.

3.2 Solution

Algorithm 1 presents the greedy algorithm for the WMC problem which at each stage chooses a set that contains the maximum weight of uncovered elements. It achieves an approximation ratio of $1 - \frac{1}{e}$.

	coKyw	coCnx	coSkm	coDat	Average
IlluSnip	0.1000	0.0540	0.6820	0.3850	0.3053
TA+C	0.9590	0.4703	0.0425	0.0915	0.3908
PrunedDP++	1	1	0.0898	0.2133	0.5758
CES	0.9006	0.3926	0.3668	0.2684	0.4821
KSD	0.8352	0.3595	0.8651	0.4247	0.6211

Table 1: Average scores of different methods over all the query-dataset pairs.

	coKyw	coCnx	coSkm	coDat	Average
data.gov.uk	0.7643	0.2882	0.8249	0.3870	0.5661
DMOZ-1	0.8977	0.7955	0.8873	0.4726	0.7633
DMOZ-2	0.8433	0.2444	0.8710	0.4569	0.6039
DMOZ-3	0.8395	0.2337	0.8693	0.4145	0.5893
DMOZ-4	0.7936	0.1877	0.8521	0.3731	0.5516

Table 2: Average scores of KSD over each group of query-dataset pairs.

Assuming $\mathtt{q}(S \cup \{t\}) - \mathtt{q}(S)$ is computed in $O(1)$, the overall running time of a naive implementation of the algorithm is $O(k \cdot n)$, where n is the number of RDF triples in T. A more efficient implementation may use a priority queue to hold candidate triples, which is left as our future work.

4 Evaluation

Our evaluation reused the 387 query-dataset pairs in [7] where datasets were collected from DataHub and queries included 42 real queries submitted to data.gov.uk and 345 artificial queries comprising i category names in DMOZ referred to as DMOZ-i for $i = 1, 2, 3, 4$. We compared our proposed KSD with four baseline methods evaluated in [7], namely IlluSnip [2], TA+C [5], PrunedDP++ [6], and CES [4]. Following [7], we set $k = 20$, i.e., a snippet contained at most 20 triples.

4.1 Quality of Snippets

Table 1 presents the average scores of the four evaluation metrics over all the query-dataset pairs. Compared with the baselines, KSD achieved the highest overall score of 0.6211. In particular, its coverage of schema ($\mathtt{coSkm} = 0.8651$) and data ($\mathtt{coDat} = 0.4247$) were at the top. Its coverage of query keywords ($\mathtt{coKyw} = 0.8352$) was close to TA+C, PrunedDP++, and CES which are query-focused methods. Therefore, KSD achieved a satisfying trade-off between these evaluation metrics. On the other hand, its coCnx score was not high because coCnx was not explicitly considered in our approach.

Table 2 breaks down the scores of KSD into groups of query-dataset pairs. The scores on different groups were generally consistent with each other, demonstrating the robustness of our approach. One exception was the very high coCnx score on DMOZ-1, due to Eq. (2) where $\mathtt{coCnx} = \mathtt{coKyw}$ when $|Q| = 1$.

4.2 Running Time

We tested the running time of our approach on an Intel Core i7-8700K (3.70GHz) with 10GB memory for the JVM.

Among all the 387 query-dataset pairs, for 234 (60.47%) a dataset snippet was generated within 1 second, and for 341 (88.11%) one was generated within 10 seconds. The median time was 0.51 second, showing promising performance

for practical use. In the worst case, it took 150 seconds to process a large dataset containing more than 2 million RDF triples. Future work would be needed to improve the performance of our implementation to handle large datasets.

5 Conclusion and Future Work

In this ongoing work, we proposed KSD, a new approach to generating snippets for dataset search. By directly optimizing three evaluation metrics, KSD outperformed four baselines. It has established new state-of-the-art results for future work. We are working towards a full version of KSD which will also optimize `coCnx`. We will implement our approach in a prototype of a new dataset search engine, to help users conveniently judge the relevance of a retrieved dataset.

There are limitations in our work. First, the current version of KSD has considered three metrics but we exclude `coCnx`. The other three metrics are all about covering some elements with selected RDF triples, whereas `coCnx` is related to graph connectivity. The weighted maximum coverage problem seems not expressive enough to model `coCnx`. We will explore other possibilities. Second, although the running time of our current implementation is acceptable in most cases, its performance is not satisfying on large datasets. We will consider using priority queue and appropriate indexes to make the generation process faster.

Acknowledgements

This work was supported by the NSFC under Grant 61572247. Cheng was funded by the Six Talent Peaks Program of Jiangsu Province under Grant RJFW-011.

References

1. Brickley, D., Burgess, M., Noy, N.F.: Google dataset search: Building a search engine for datasets in an open web ecosystem. In: WWW 2019. pp. 1365–1375 (2019)
2. Cheng, G., Jin, C., Ding, W., Xu, D., Qu, Y.: Generating illustrative snippets for open data on the web. In: WSDM 2017. pp. 151–159 (2017)
3. Ellefi, M.B., Bellahsene, Z., Breslin, J.G., Demidova, E., Dietze, S., Szymanski, J., Todorov, K.: RDF dataset profiling - a survey of features, methods, vocabularies and applications. Semantic Web **9**(5), 677–705 (2018)
4. Feigenblat, G., Roitman, H., Boni, O., Konopnicki, D.: Unsupervised query-focused multi-document summarization using the cross entropy method. In: SIGIR 2017. pp. 961–964 (2017)
5. Ge, W., Cheng, G., Li, H., Qu, Y.: Incorporating compactness to generate term-association view snippets for ontology search. Inf. Process. Manage. **49**(2), 513–528 (2013)
6. Li, R., Qin, L., Yu, J.X., Mao, R.: Efficient and progressive group steiner tree search. In: SIGMOD 2016. pp. 91–106 (2016)
7. Wang, X., Chen, J., Li, S., Cheng, G., Pan, J., Kharlamov, E., Qu, Y.: A framework for evaluating snippet generation for dataset search. In: ISWC 2019 (2019), https://arxiv.org/abs/1907.01183

Towards Employing Semantic License Annotations for Sensor Data Profiling

Anna Fensel[1], Tassilo Pellegrini[2], Oleksandra Panasiuk[1]

[1] University of Innsbruck, Department of Computer Science,
Semantic Technology Institute (STI) Innsbruck
Technikerstr. 21a, A-6020 Innsbruck, Austria
{anna.fensel, oleksandra.panasiuk}@sti2.at

[2] Department of Media Economics,
University of Applied Sciences St. Pölten
Matthias Corvinus Strasse 15, A-3100 St. Pölten, Austria
tassilo.pellegrini@fhstp.ac.at

Abstract. The paper outlines the most up to date version of the semantic licenses library of Data Licenses Clearance Center (DALICC), and discusses the possibilities of employing it for data profiling. In particular, we outline possible real-life use case directions from the domain of the vehicle sensor data profiling, an approach for the evaluation of the DALICC system in use, as well as possible further directions for the settings requiring cooperation with the data owners, such as at digital workplaces.

Keywords: Data licensing, knowledge graph, semantic technology, sensor data, use case, evaluation.

1. Introduction

With large amounts of data being available, the profiling of data gets very important and has become an active research and development area [4]. The methods suggested up to now have focused mainly on the annotation of the contents of data, for example on datasets recommendation and linking [1], or vocabulary and vocabulary terms recommendations [11]. Licensing of data has been recognized as an important part of the data profiling [2]. Further, explicit license information in the data profiling will facilitate the implementation of laws such as General Data Protection Regulation (GDPR). To make the reuse of the data and content more efficient, such profiling should include semantic representations of the deontic conditions (permissions, prohibitions and duties) specified within licenses and provenance information about the associated data broadly in practice. The approaches and tools for such developments are still actively evolving.

The creation of derivative data works, i.e. for purposes like content creation, service delivery or process automation, is often accompanied by legal uncertainty about usage rights

and high costs in the clearance of licensing issues. The DALICC project [8] has developed a software framework[1] that supports the automated clearance of rights issues in the creation of derivative data and software works [7]. In essence, DALICC helps to determine which information can be shared with whom to what extent under which conditions, thus lowering the costs of rights clearance and stimulating the data economy.

Specifically, we present the current, most up-to-date version of the DALICC License Repository [10] containing the basic international and a large variety of the national licenses, with the semantic license models as well as the corresponding documentation. The repository's last extensions have significantly increased the number of licenses present in the platform, as well as substantially improved the documentation.

The linked data empowered repository for storing the structured semantic license data for specific license data is set up, and is currently the most or one of the most complete repositories of this kind. The data access is provided via appropriate interfaces using in particular REST/Web Service access, SPARQL endpoint (for semantic data). The repository is serving anybody who wants to address checking of the licenses' specificities and their compatibility, using reasoning engines or license design tools. The tools are suitable for license engineering in various scenarios, making them a very good foundation for use cases from different sectors.

One of the new scenarios related to the collection and processing of the sensor data includes enabling the data owners to give consent on how their data is used. This implies making the data sharing and usage policies explainable to the data owners, as well as giving the ability to the data owners to license their data to the service provider. In this paper, we elaborate how a semantic data licensing solution, such as DALICC, can be used for such sensor data sharing scenarios, particularly, for the vehicle sensor data. The data collected by various sensors in a modern vehicle are large in quantity and variety, record in detail various performance and usage aspects of the vehicle, and are broadly used in scenarios such as quality assurance and predictive maintenance of the vehicles, as well as increasingly in other scenarios, such as traffic flow optimisation or insurance policies. Such data has a sensitive character, e.g. it may characterize the driving style of the vehicle owner. As the vehicle user (as the data producer) is the owner of the generated sensor data, provisioning him/her a legally grounded data sharing or contracting solution is essential.

The paper is structured as follows. Section 2 introduces the semantic license library of DALICC. Section 3 describes examples of technical settings where the DALICC solution can be employed, particularly, in a scenario involving vehicle sensor data. Section 4 describes an evaluation approach for the DALICC system, and Section 5 concludes the paper and provides an outlook for future work.

2. Semantic License Library

During the DALICC project runtime, we have performed an in-depth Rights Expression Languages (RELs) evaluation that laid the foundation for compiling the relevant set of machine-processable RELs and complementing vocabularies [9]. Based on the research of existing

[1] www.dalicc.net

RELs, we chose the Open Digital Rights Language (ODRL)[2] as it is particularly suitable for modeling licenses. Furthermore, the ODRL vocabulary includes terms that are deprecated or supplemented by terms from Creative Commons (CC) REL. However, we discovered that even a combination of various vocabularies is not sufficient to represent all of the necessary license concepts. To fill this gap, we constructed a DALICC vocabulary and introduced additional terms. Moreover, DALICC utilizes a dependency graph for representing the semantic relationships between defined concepts (see Figure 1). The interdependencies here are crucial for the license modelling, and particularly for interoperation and reasoning scenarios, where e.g. the data comes from various platforms and is being profiled using various license semantic annotations.

We elaborated a modelling workflow whose purpose is to govern a user through the process of a license composition. This stream of activity was accompanied by a steep learning curve, especially as we detected multiple complex interdependencies between the domain ontology, the rights expression languages and the dependency graph (see Figure 2) representing the logical relationships between domain concepts. In DALICC, the license engineering is performed by letting the user to answer basic questions on what he/she needs from the license, and the answers are utilized in the workflows for the license selection or creation. The resulting licensing outcomes are semantically interoperable with existing semantic license semantic models, in particular employing the developed dependency graph.

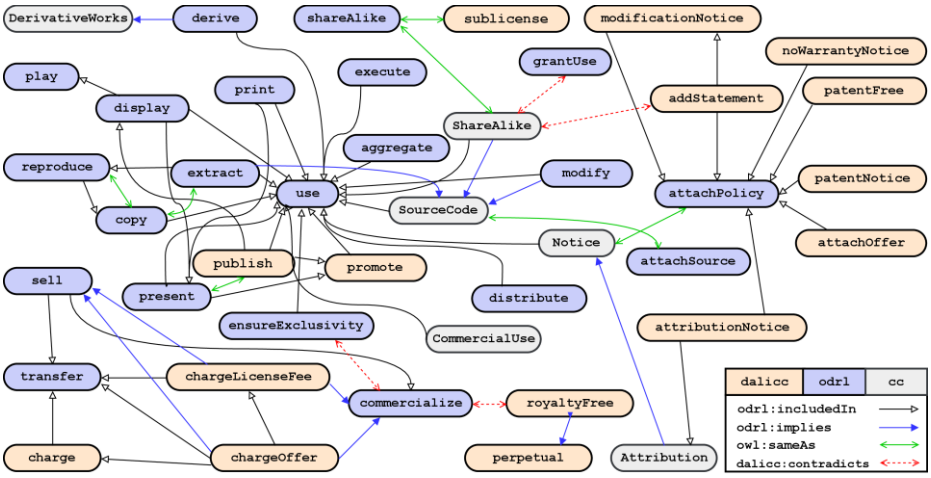


Fig. 1: Semantic Model for the Knowledge Graph of DALICC

[2] https://www.w3.org/TR/odrl-model

Finally, we set up the License Library [10], a repository containing currently 117 legally valid and approved licenses in human-readable and machine-readable form relevant for the licensing of digital assets. The data can be accessed via the publicly available demonstrator, and it can be retrieved via a REST/Web Service[3] and via the SPARQL endpoints: for the licenses[4] and for the licenses' metadata[5].

3. Employing the Semantic License Library for Data Profiling

The semantic licenses are intended to be used within the use cases of the CampaNeo project and are described below. In the CampaNeo project[6] (according to its proposal), "an open platform will be developed on which private and public institutions can create campaigns and collect and analyze vehicle data in real time. The goal is to set up a prototype platform for secure campaign-based data collection in Hannover, Wolfsburg and in cross-regional scenarios, as well as the implementation of the first smart use cases based on the campaign data. The focus is in particular on the data ownership of vehicle owners and the traceability of data processing". The campaigns will be run to collect and process the vehicle data to improve certain real-life situations e.g. in the city and regional traffic, insurance, etc., and ensure that the vehicle and the data owners are active first class participants of these campaigns.

When approaching the implementation of this goal, there are two challenges or use cases, where data licensing is of relevance. First, it should be communicated to the user which of his/her data may be used and in which manner, and an option to authorize the usage should be available; second, the usage contract has to be formed for the data, so that the data can be used at the platform.

3.1 Use Case 1: Transparent and explainable data sharing

A concept and a light-weight prototype for an approach for transparent and explainable sharing of the data will be designed. It will facilitate the understanding of the data sharing obligations and permissions, both for the data owners as well as for the data users. The actual data sharing workflows and the usages are also to be made traceable and displayable for the data owners, giving them a better understanding of the actual value of their data.

When the visualized data comes from a knowledge graph that has been built with machine learning (e.g. such as in Google's Knowledge Vault [5], or in the scenarios employing aggregated sensor data), the probabilities of the correctness of certain knowledge graph constructs will also be taken into account when visualizing the data. For example, the probability that the data will have an impact on one or another geographical region will be displayed. The latter can be implemented by analyzing the specifics of the structured and non-structured data of the geographical regions with similar characteristics, as well as taking of the known or expected trends into account. The approach and the solution will also contribute to the

[3] https://dalicc.net/license-library
[4] https://dalicc-virtuoso.poolparty.biz/sparql
[5] https://dalicc.net/license-library-meta
[6] https://www.sti-innsbruck.at/research/projects/platform-real-time-vehicle-data-campaigns

field of Explainable Artificial Intelligence. The works in the latter field up to now mainly focus on explaining the machine learning (in particular, deep learning) outcomes to the users, but little has been done so far in explaining the data sharing practices (especially the ones that are of larger scale and not easily comprehensible for the users) employing knowledge graphs, and especially in this project's domain.

The prototype (proof of concept) will be based on a web and mobile framework (such as Angular), which will enable it to be deployed in various settings and be independent from the specificities of proprietary app platforms.

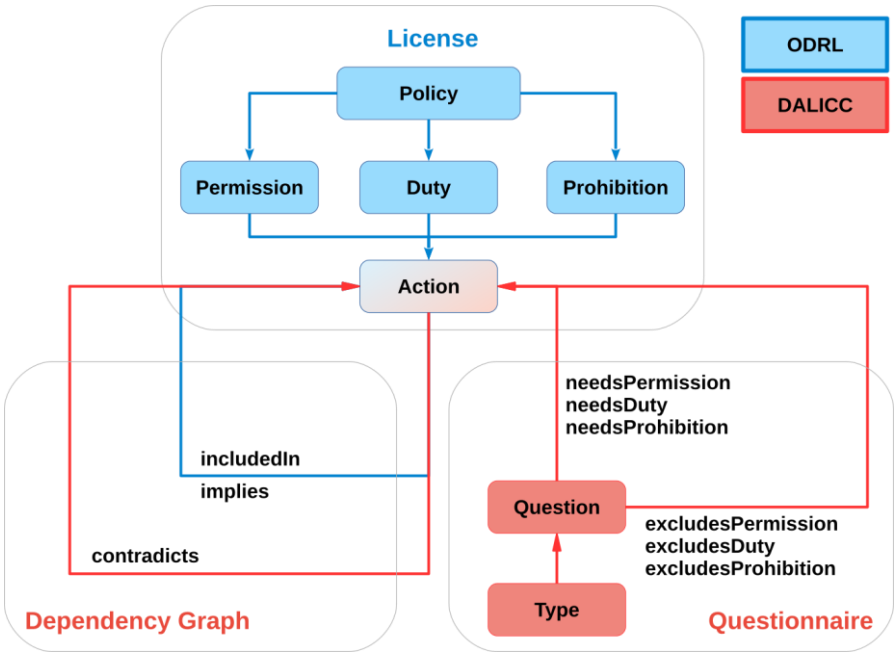


Fig. 2: Interplay of License Ontology, Dependency Graph and Modelling Workflow

3.2 Use Case 2: Knowledge graph based models for smart contracts

We will conceptualize the models needed for explaining the data sharing practices to the user, as well as for the formation of the smart contracts i.e. formalizations and protocols that are to be used for defining, controlling and executing the agreements comprising the data sharing rights and obligations. Technically, knowledge graphs, applying semantic modelling techniques, will be utilized. The modeled concepts will comprise the information needed for the representation of a smart contract, and will take into account the information about the relevant context (e.g. such information may comprise the records about the use of different parts

of the vehicle, geo data) and the users/user groups – typical data providers and consumers (e.g. drivers, manufacturer companies, public authorities). The knowledge graph based models will be used to comprehensively explicate the parts of the semantic models relevant for the transparency and the explanations. The CampaNeo data analytics module will be one of the sources of the raw data for the construction of the knowledge graph, and a part of the graph will be built employing machine learning techniques. The task's resulting models will be technically embedded in the project's smart contracts on a blockchain infrastructure.

Various techniques may be used to enable and facilitate the attachment of the semantic licenses to the data and content. The possibilities include such options as:

- employing meta tags: RDF file attached or link to the file. In particular, the Extensible Metadata Platform (XMP) ISO standard can be used to include links to the specific data licenses. XMP can be used in several file formats such as PDF, JPEG, JPEG 2000, JPEG XR, GIF, PNG, WebP, HTML, TIFF, Adobe Illustrator, PSD, MP3, MP4, Audio Video Interleave, WAV, RF64, Audio Interchange File Format, PostScript, Encapsulated PostScript, and proposed for DjVu,
- introducing hashtags determining the authors, timestamps and applied licenses: this technique would be useful for systems such as blockchains. It remains to be clarified whether the use cases will require storing of the whole license history and its evolution, in the way the blockchain systems typically enable it.

The corresponding tool support, such as a web application accessible with an API (that could work together with some platforms like GitHub, Facebook,...) may be realised. In particular, in Github or in Zip archives, the license can be inserted as the RDF/XML-file, instead of the usual text-file. A part of the solution may also include a Python library (e.g. employing an XMP Python toolkit) for connecting data and content files and licenses, or a stand-alone web service attaching the licenses to the data and content.

4. A Use Case-based Approach to DALICC Evaluation

While the main experimental contribution of DALICC is the design and implementation of a platform that facilitates correct usage of the semantically defined licenses, it is important to systematically approach the evaluation of the DALICC solution in the use cases.

In this way, we raise four hypotheses regarding the effectiveness of the proposed solution:

- H1: The use of the proposed platform facilitates correct selection and/or creation of the semantic licenses.
- H2: The solutions suggested by the proposed platform are clear and explainable to the users.
- H3: The use of the proposed system advances the content and data sharing economy.
- H4: The use of the system increases users' satisfaction / meets the users' goals.

For realizing the platform's components, we have been adopting parts of the well-known design science paradigm for information systems proposed by Hevner et al. [6]. All developed artefacts have been evaluated using representative sample scenarios (e.g. "create a new license", "assign a license to a dataset", etc.), and investigated with the help of well-defined case studies (up to now, the initial case studies from the DALICC project, and from now on

also the presented use cases of the CampaNeo project). These case studies allow us to draw conclusions about the general applicability of the developed artefacts and provide feedback for potential further refinement. Both development and evaluation of the platform's components have been carried out in close collaboration with legal experts from a law firm[7]. They ensured that the platform not only functions correctly technically, but also delivers correct results from the legal point of view. H1 and H2 have been already initially tested within the scope of the DALICC project development. The evaluations have been taking place involving the project experts and network (5 organisations from the project, and 5 organisations in addition, ca. 10 people in total) to exploit the platform with the aim to check if the basic scenarios are working correctly. For further use case driven evaluations, a similar approach is being followed.

In order to evaluate the proposed system with respect to all the hypotheses (and especially, H3 and H4), user studies are being performed. For this, a prototype of the proposed system has been deployed on the Web, and is available via the DALICC's website. We are facilitating further users, who have not been involved in the project, with targeted hands-on workshops, to get feedback on all the hypotheses. Further, we communicate our results to relevant bodies that can have a multiplier effect on the application of our solution, such as political bodies (e.g. the EC), recommendation/standardization bodies (e.g. W3C). The success of the work with them, in particular, impacts the outcomes for H3 and H4, as these depend on the level of priority set for such solutions by the regulators. Here, we are however optimistic, as the solutions for making the data usage practices more transparent and interoperable with the semantic technologies, as well as for making the data adding more value to the data owners are in the highest demand, as revealed in an EU Big Data research roadmap that takes into consideration the societal impact of the data [3].

5. Conclusion and Future Work

We envision approaches such as DALICC to change particularly the digital workplaces of the future, making the data profiling and sharing policies even more accessible. As defined by Gartner "The Digital Workplace enables new, more effective ways of working; raises employee engagement and agility; and exploits consumer-oriented styles and technologies."[8]. Ontological, or semantic, sharing of meaning is essential for the state of the art work scenarios, applicable to knowledge intensive labor, where also customers become collaborators. For example, the vehicle owners may choose to contribute their vehicle sensor data for one or another purpose (e.g. choosing to contribute the data to the city authorities, insurance companies, etc.). With development and application of new data and content licensing semantic techniques, we aim to bring the area of Digital Workplace to the new level, by assisting humans in highly intellectual tasks, that so far are barely being delegated to the machines:

[7] https://h-i-p.at/en

[8] https://www.gartner.com/it-glossary/digital-workplace/

namely, in decision making, content and data selection, creation and distribution, and management activity.

This will be achieved in the chosen application domains going beyond the current state of the art of ontology-based service interfacing, integration and participation involvement. The potential further work directions are as follows:

- Enabling easier license modeling of the data and content in both design time and the run time of the digital workplace scenario – and eventually the organisations creating their own applications and workflows based on these models,
- License-relying schemes rewarding and motivating data, content and service providers, that can be deployed in transparent infrastructures, such as blockchains; advancing the design and implementation of the data and content value chain and economy,
- Speeding up the velocity of the data and content flow in information systems (e.g. in scenarios connected with content generation, reporting),
- Making the decision processes transparent, traceable, and easier to optimize (e.g. it can be easily established which nodes are causing delays, inconsistencies), integrate new techniques facilitating easier data use in decision making, particularly, with the semantic information on how the data and content can be licensed,
- Visualisation of the data and workflows in a form that is actionable to humans in a digital workplace scenario, taking into account the license and provenance information.

Acknowledgements. The work is partly funded by the Austrian Federal Ministry of Transport, Innovation and Technology (BMVIT) - DALICC and CampaNeo projects.

References

1. M. Achichi, M. Ben Ellefi, Z. Bellahsene, and K. Todorov. Doing Web Data: from Dataset Recommendation to Data Linking. *NoSQL Data Models: Trends and Challenges*, *1*, 57-91, 2018.
2. M. Ben Ellefi, Z. Bellahsene, J. G. Breslin, E. Demidova, S. Dietze, J. Szymański, and K. Todorov. RDF dataset profiling–a survey of features, methods, vocabularies and applications. *Semantic Web*, (Preprint), 1-29, 2018.
3. M. Cuquet and A. Fensel. The societal impact of big data: A research roadmap for Europe, *Technology in Society*, Elsevier, 2018. DOI: https://doi.org/10.1016/j.techsoc.2018.03.005
4. S. Dietze, E. Demidova, and K. Todorov. RDF Dataset Profiling. *Encyclopedia of Big Data Technologies*, Springer International Publishing, pp.1378-1385, 2019.
5. X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 601-610, 2014.
6. A. Hevner, S. T. March, J. Park, and S. Ram. Design science in information systems research. *MIS Quarterly, 28(1)*, 75-105, 2004.
7. T. Pellegrini, G. Havur, S. Steyskal, O. Panasiuk, A. Fensel, V. Mireles-Chavez, T. Thurner, A. Polleres, S. Kirrane, and A. Schönhofer. DALICC: A License Management Framework for Digital Assets, in: *Data Protection / LegalTech - Proceedings of the 22nd International Legal Informatics Symposium IRIS 2019, Colloquium. Presented at the IRIS 2019 - 21st International Legal Informatics Symposium*, Salzburg, Austria, 2019.

8. T. Pellegrini, V. Mireles, S. Steyskal, O. Panasiuk, A. Fensel, and S. Kirrane. Automated Rights Clearance Using Semantic Web Technologies: The DALICC Framework. In *Semantic Applications*, pp. 203-218, 2018.
9. T. Pellegrini, A. Schönhofer, S. Kirrane, S. Steyskal, A. Fensel, O. Panasiuk, V. Mireles-Chavez, T. Thurner, M. Dörfler, and A. Polleres. A Genealogy and Classification of Rights Expression Languages – Preliminary Results, in: *Data Protection / LegalTech - Proceedings of the 21st International Legal Informatics Symposium IRIS 2018*, Salzburg, Austria, pp. 243–250, 2018.
10. O. Panasiuk, S. Steyskal, G. Havur, A. Fensel, and S. Kirrane. Modeling and Reasoning over Data Licenses. In: Gangemi A. et al. (eds) *The Semantic Web: ESWC 2018 Satellite Events*. Lecture Notes in Computer Science, vol 11155. Springer, pp.218-222, 2018.
11. I. Stavrakantonakis, A. Fensel, and D. Fensel. Linked Open Vocabulary ranking and terms discovery. In *Proceedings of the 12th International Conference on Semantic Systems*, pp. 1-8, 2016.

Mining Machine-Readable Knowledge from Structured Web Markup

Ran Yu

GESIS - Leibniz Institute for the Social Sciences
50676 Köln, Germany
ran.yu@gesis.org

Abstract. The World Wide Web constitutes the largest collection of knowledge and is accessed by billions of users in their daily lives through applications such as search engines and smart assistants. However, most of the knowledge available on the Web is unstructured and is difficult for machines to process which leads to the lowered performance of such smart applications. Hence improving the accessibility of knowledge on the Web for machines is a prerequisite for improving the performance of such applications.

Knowledge bases (KBs) here refers to RDF datasets contains machine-readable knowledge collections. While KBs capture large amounts of factual knowledge, their coverage and completeness vary heavily across different types of domains. In particular, there is a large percentage of less popular (long-tail) entities and properties that are under-represented.

Recent efforts in knowledge mining aim at exploiting data extracted from the Web to construct new KBs or to fill in missing statements of existing KBs. These approaches extract triples from Web documents, or exploit semi-structured data from Web tables. Although the extraction of structured data from Web documents is costly and error-prone, the recent emergence of structured Web markup has provided an unprecedented source of explicit entity-centric data, describing factual knowledge about entities contained in Web documents. Building on standards such as RDFa, Microdata and Microformats, and driven by initiatives such as *schema.org*, a joint effort led by Google, Yahoo!, Bing and Yandex, markup data has become prevalent on the Web. Through its wide availability, markup lends itself as a diverse source of input data for KBA. However, the specific characteristics of facts extracted from embedded markup pose particular challenges.

This work gives a brief overview of the existing works on mining machine-readable knowledge from both structured and unstructured data on the Web, and introduces the KnowMore approach for augmenting knowledge bases using structured Web markup data.

Using Semantic Domain-Specific Dataset Profiles for Data Analytics

Simon Gottschalk

L3S Research Center
Leibniz University Hannover, Germany
gottschalk@L3S.de

Abstract. The availability of a vast amount of heterogeneous datasets provides means to conduct data analytics in a wide range of applications. However, operations on these datasets demand not only data science expertise, but also knowledge about the structure and semantics behind the data. Semantic data profiles can enable non-expert users to interact with heterogeneous data sources without the need for such expertise.
To support efficient semantic data analytics, a domain-specific data catalog, that describes datasets utilizable in a given application domain, can be used [1]. Precisely, such a data catalog consists of dataset profiles, where each dataset profile semantically describes the characteristics of a dataset. Dataset profile features not only include a set of well-established features (e.g. statistical and provenance features), but also connections to a given semantic domain model. Such a domain model describes concepts and relations in a specific domain and thus helps to automate data processing in a semantic meaningful manner. An example is the mobility domain and the integration of different spatial representations.
Once created, a domain-specific data catalog can support a whole data analytics workflow. This includes, but is not limited to search through the use of semantic concepts (e.g. datasets about street segments), domain-specific feature extraction (e.g. geo-transformations), and machine learning with the help of concept-based type checking. These examples demonstrate that the provision of semantic domain-specific profiles is a valuable step towards data analytics when dealing with heterogenous datasets.

Acknowledgements

This work was partially funded by the Federal Ministry of Education and Research (BMBF), Germany under Simple-ML (01IS18054).

References

1. S. Gottschalk, et al. "Simple-ML: Towards a Framework for Semantic Data Analytics Workflows." SEMANTiCS (2019).

Towards Automatic Domain Classification of LOV Vocabularies

Alexis Pister, Ghislain Atemezing

MONDECA, 35 boulevard de Strasbourg 75010 Paris, France.
<firstname.lastname@mondeca.com>

Abstract. Assigning a topic or a domain to a vocabulary in a catalog is not always a trivial task. Fortunately, ontology experts can use their previous experience to easily achieve this task. In the case of Linked Open Vocabularies (LOV), a few number of curators (only 4 people) and the high number of submissions lead to find automatic solutions to suggest to curators a domain in which to attach a newly submitted vocabulary. This paper proposes a machine learning approach to automatically classify new submitted vocabularies into LOV using statistical models which take any texts description found in a vocabulary. The results show that the Support Vector Machine (SVM) model gives the best micro F1-score of 0.36. An evaluation with twelve vocabularies used for testing the classifier shades light for a possible integration of the results to assist curators in assigning domains to vocabularies in the future.

Keywords: Ontologies, Classification, Machine Learning, Linked Open Vocabularies

1 Introduction

Linked Open Data (LOD) refers to the ecosystem of all the open source structured data which follows the standard web technologies such as RDF, URIs and HTTP. As the number of available data grows with time, new datasets following these principles appear. Linked Open Vocabulary (LOV) [1] is an initiative which aims to reference all the available vocabularies published on the Web following best practices guided by the FAIR (Findable - Accessible - Interoperable - Reproducible) principles. Each vocabulary can be seen as a knowledge graph, describing the properties and the purpose of the vocabulary, and which can be connected to other vocabularies by different types of links. Therefore, LOV can be seen as a knowledge graph of interlinked vocabularies [16] accessible on the Web of data.

When a new ontology is submitted for integration into LOV, a curator needs to assign at least one tag representing a domain or a category to the vocabulary

[1] https://lov.linkeddata.es/dataset/lov/

among existing 43 categories, such as “Environment”, “Music” or “W3C REC”. A category aims at grouping ontologies according to a domain. For example, the tag “W3C REC” represents ontologies recommended by the W3 Consortium, such as `rdf` or `owl`. As the number of domains increases and some vocabularies[2] can be relatively small, the tagging process can be biased. Figure. 1 depicts the list of the tags available in LOV as the time of writing this paper, while Figure 2 depicts their distribution. One of the benefit of assigning a tag to a vocabulary is to index it according to a domain and make it easy to access from the interface. For example, to access to vocabularies in the IOT domain, the direct URL in LOV is `https://lov.linkeddata.es/dataset/lov/vocabs?tag=IoT`. Additionally, any newly added vocabulary should belong to at least one domain.

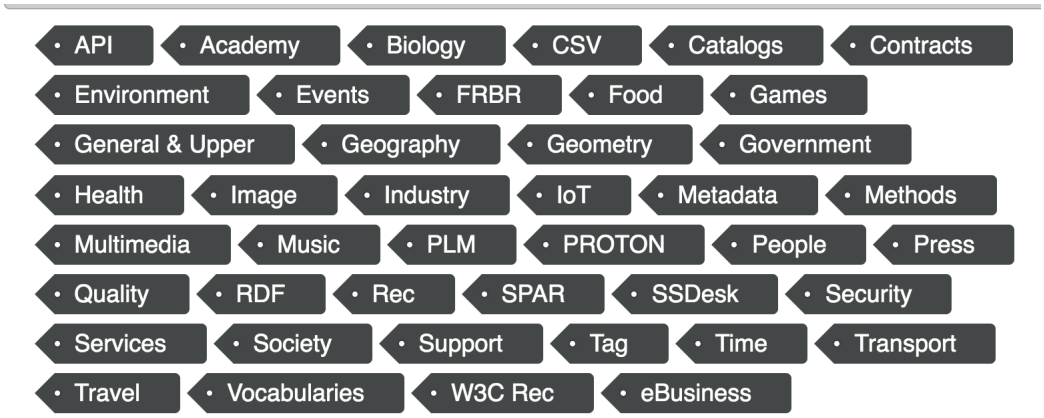


Fig. 1: A view of the list of the tags available in LOV backend used for classifying ontologies

We propose a machine learning approach to automatically classify newly submitted vocabularies with statistical models which take texts describing the subjects of the vocabularies as input. Indeed, the majority of the graphs contains a lot of text describing the subjects and the properties of the vocabularies, in the form of string literals. For example, the URI in a given ontology (Class or Property) is often described by the predicate `rdfs:comment` with a text mentioning the comment of a given resource. Other predicates are often linked to texts containing information, such as `rdfs:label` or `dct:description`. We used all this text information to train several machine learning models in the purpose of classifying the vocabularies into different categories. This paper is structured as follows: Section 2 describes related work in graph classification, followed by the

[2] In this paper, the terms ontology and vocabulary are interchangeable

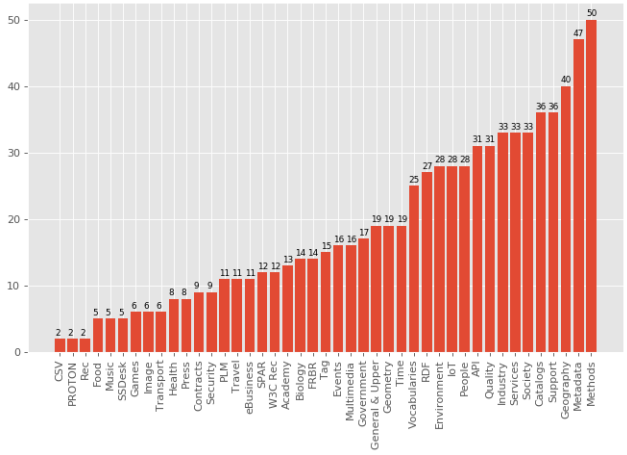


Fig. 2: Distribution of the tags among the vocabularies in LOV

machine learning approach to build the classifier in Section 3. Section 5 provides an evaluation of our approach and a brief conclusion in Section 6

2 Related Work

Graph classification is a problem well studied in the literature. Several strategies have been developed to tackle this problem such as kernel methods or more recently graph neural networks [13]. However, there is way less work made in knowledge graph classification. What comes closest are the entity or triples classification problems which consist in the categorization of a really small subset of a knowledge graph [17]. It is because these types of graphs are mainly described by their entities and relations, so it would be very difficult to find similarities or dissimilarities between knowledge graphs which share very little or not a common entity or relation, like it is often the case. This is why we used a different approach of traditional graph classification methods for our problem, using a text mining strategy. Indeed, a lot of work has been made in document classification [1]. Various processing methods have been elaborated such as Bag of Words or Latent Semantic Analysis (LSA), whose input can be easily exploited by machine learning algorithms.

Classifying datasets created using semantic technologies has been applied in the literature. The most closest work in the literature is described in [7] and [15]. Meusel et al. present a methodology to automatically classify LOD datasets based on the different categories presented in the LOD cloud diagram. The paper uses eight feature sets from the LOD datasets, among others are text from `rdfs:label`. One of the main conclusions of the paper is that vocabulary-level features are good indicator for the topical domain.

While the mentioned approach uses supervised learning, we apply two more steps in preparing the corpus for input of the classifier, using Bag-of-Word and a Truncated SVD transformation. Additionally, we have a very small amount of corpus inherent to the size of vocabularies compared to the entire LOD datasets, and a higher number of available tags (43 in LOV compared to 8 for the LOD cloud).

3 Data Preparation and Machine Learning Models

3.1 Data Preparation

Our approach has been to use the texts contained in the vocabularies to classify them into categories. Indeed, usually the subject of a RDF graph and the purpose of its entities are described in string literals following some specific predicates. We first extract this relevant textual information (string or literal) inside each graph (a dump representing the latest version of the vocabulary in N3), and concatenate it into one paragraph describing their subjects. To this end, we first download each recent version of the vocabulary from LOV SPARQL endpoint (taking the most recent version tracked by LOV) and import them into graph objects with RDFLib[3]. Listing 1.1 depicts the SPARQL query used to retrieve the latest version of each vocabulary, alongside with their domains and unique prefix.

```
SELECT DISTINCT  ?vocabPrefix ?domain ?versionUri  {
        GRAPH <https://lov.linkeddata.es/dataset/lov>{
        ?vocab a voaf:Vocabulary.
        ?vocab vann:preferredNamespacePrefix ?vocabPrefix.
        ?vocab dcterms:modified ?modified.
    ?vocab dcat:keyword ?domain.
    ?vocab dcat:distribution ?versionUri.
    BIND ( STRAFTER(STR(?versionUri), "/versions/") as ?v)
    BIND(STRBEFORE(STR(?v), ".") as ?v1)
    BIND (STR(?modified) as ?date )
    FILTER ( ?date = ?v1)
  }} GROUP BY ?vocabPrefix ?domain ?versionUri
ORDER BY ?vocabPrefix ?domain ?versionUri
```

Listing 1.1: SPARQL query to retrieve the latest versions of vocabularies stored in LOV

We then concatenate all the strings followed by the predicates having one of these suffixes : `comment`, `description`, `label` and `definition`. The predicate `rdfs:label` is often used to give a name of an URI in natural language, while the suffixes `comment`, `description` and `definition` are used to give insight on the meaning and purpose of a given ontology or entity. The result of this step has been the generation of a paragraph for each vocabulary. As the texts describe the

[3] `https://github.com/RDFLib/rdflib`

RDF properties of the graphs, they often contain the suffixes of these properties formed of several words not separated by spaces, in camel case format. For example, if an extracted text mentions the property "UnitPriceSpecification", this expression will remains as a single unit in the final text. However, it can imply a bias on the statistical model to be applied on this data. Consequently, we separate all these types of expression with spaces, when a uppercase occurs in the middle of a word. Therefore, by using this method, the expression "UnitPriceSpecification" will be transformed to "Unit Price Specification" in the final text. After this transformation, the whole corpus' vocabulary is formed of $21,435$ different words. The mean word count for the paragraphs is 1168.5, the maximum is 86208 and the minimum 0. Two paragraphs were empty and 25 of them have less than 20 words. The text describing the rooms vocabulary [4] obtained with the pre-processing step described in this section is presented in Listing 1.2. This ontology describes the rooms one can find in a building and has the following assigned tags in LOV: Geography and Environment.

Floor Section. Contains. Desk. Building. Floor. A space inside a structure, typically separated from the outside by exterior walls and from other rooms in the same structure by internal walls. A human-made structure used for sheltering or continuous occupancy. Site. A simple vocabulary for describing the rooms in a building. An agent that generally occupies the physical area of the subject resource. Having this property implies being a spatial object. Being the object of this property implies being an agent. Intended for use with buildings, rooms, desks, etc. Room. The object resource is physically and spatially contained in the subject resource. Being the subject or object of this property implies being a spatial object. Intended for use in the context of buildings, rooms, etc. A table used in a work or office setting, typically for reading, writing, or computer use. A named part of a floor of a building. Typically used to denote several rooms that are grouped together based on spatial arrangement or use. A level part of a building that has a permanent roof. A storey of a building. Occupant. An area of land with a designated purpose, such as a university Campus, a housing estate, or a building site.

Listing 1.2: Paragraph describing the rooms vocabulary, obtained with the preprocessing pipeline described in Section 3.

[4] `https://lov.linkeddata.es/dataset/lov/vocabs/rooms`

3.2 Machine Learning Models

As we cannot feed directly text paragraphs to the machine learning models, we applied a processing pipeline for transforming the texts into fixed-size vectors of attributes. For this purpose, we used several techniques described in [14] : we first apply a Bag-of-Words (BoW) transformation, mapping the texts to vectors containing the frequencies of each word and ngram made of 2 and 3 words in the documents which have a frequency value between 0.025 and 0.25. Then, a Term Frequency-Inverse Document Frequency (TF-IDF) is applied to normalize the frequencies of the words and ngrams by the length of each document. Finally, we apply a Latent Semantic Analysis (LSA) [3] which is a dimensionality reduction technique using a linear algebra method called truncated SVD, to map the space of word frequencies to a smaller space of concepts. Indeed, the dimension of the TF-IDF vectors is big, as it corresponds to the number of words used in the whole corpus plus the frequent ngrams (21,435). It is well-known in the literature that a high number of attributes often impact negatively a machine learning approach [2]. We tried different values of n representing the dimension of the vector space : 50, 150 and 300. These vectors of attributes are then used as input for the machine learning classifiers. The entire processing pipeline is summarized in Figure 3.

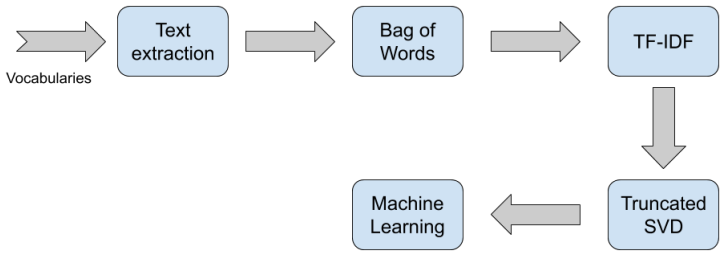


Fig. 3: Schematic view of the processing pipeline. From left to right, the diagram depicts the different steps: 1-Text extraction from Vocabulary dump; 2-BoW Transformation; 3-Normalization with TF-IDF; 4-Vector dimension reduction and finally the classifiers.

We then separated the data in two subsets composing of a training set (80% of the vocabularies) and a test set (the remaining 20%). In this paper, the dataset version of LOV used for the experiment is the snapshot as of May 7th, 2019 [5], containing 666 vocabularies. We claim that the approach described in this paper can be replicated to any type of machine learning multi-label task with a knowledge graph as input.

[5] `https://tinyurl.com/lovdataset`

As each vocabulary can have one to many tags, we tackle the problem as a multi-label classification task. A machine learning model is trained on the training set, trying to find relation between the attributes describing the graphs and their labels. The trained model is then applied to the test set. The predicted labels are finally compared to the ones tagged by human curators, and the micro precision, recall and f1-measure are computed, which are current supervised learning metrics [11]. We have tested several machine learning models with the python library scikit-learn [10], with an emphasis on the Support Vector Machine (SVM) and the Multi Layer Perceptron (MLP) which are ranked among the best classifiers for text classification task, mainly because they can handle large feature spaces [4, 12]. The K-Nearest-Neighbors (KNN) and the Random Forest (RF) classifiers have been tested as well, because they natively support multi-label classification, as well as the MLP.

However, we had to apply a One-vs-Rest strategy for the SVM [9], which consists in training a separate binary classifier for each label. The MLP had one hidden layer of size 100 with a Rectified Linear Unit (ReLU)[6] activation function. Similarly, we set the parameters $C = 10$, $gamma = 1$ for the SVM, with a radial basis function kernel (RBF kernel)[7] and weighting the classes uniformly. We chose $k = 7$ for the KNN model.

4 Results

The results of the classification for the 4 machine learning models, using $k = 50, 150, 300$ for the truncated SVD are presented in Table 1. The MLP and the SVM give the best micro F1-score respectively of 0.34 and 0.36, with $n = 150$.

	n = 50			n = 150			n = 300		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
SVM	0.22	**0.50**	0.31	0.39	0.32	**0.36**	0.47	0.23	0.31
RF	**0.74**	0.07	0.12	0.7	0.03	0.07	0.68	0.02	0.04
MLP	0.33	0.32	0.33	0.34	0.33	0.34	0.33	0.25	0.29
KNN	0.62	0.10	0.17	0.65	0.06	0.11	0.58	0.06	0.11

Table 1: Results of the classification on the test set for the 4 machine learning algorithms, with 3 values of the dimension of the feature space.

5 Evaluation and Discussion

In this section, we describe the evaluation of the classifier on newly submitted ontologies in LOV, and we discuss the results obtained comparing with manual assignment by two curators.

[6] The ReLU is the most used activation function in neural network. f(z) is zero when z is less than zero and f(z) is equal to z when z is above or equal to zero.

[7] `https://en.wikipedia.org/wiki/Radial_basis_function_kernel`

5.1 Evaluation

For evaluating our model, we took a list of 12 vocabularies in the back-end of LOV and asked two curators to assign domains to each of the vocabulary. Then, we passed the same vocabularies to the SVM classifier. The classifier's results is then compared with the human assignment tags as presented in Table 2.

Table 2: Comparison of tags suggested by the classifier and the curator. The underlined tags are the perfect match by both the human and the SVM classifier.

Vocabulary URI	Curator tag(s)	Classifier's tag(s)
`https://w3id.org/vir`	Multimedia	Support
`https://w3id.org/usability`	Support, Events	API
`https://www.w3.org/ns/solid/terms#`	Services, General & Upper	Services, General & Upper, RDF
`http://ns.inria.fr/munc/v2#`	Metadata	RDF
`https://w3id.org/arco/ontology/core`	Services, Society	Catalogs, Events, Government, Multimedia
`https://w3id.org/arco/ontology/catalogue`	Catalogs, society	Catalogs, Events, Government, Multimedia
`https://w3id.org/arco/ontology/context-description`	Support, General & Upper	Catalogs, Environment, Events, Government, Multimedia
`https://w3id.org/arco/ontology/denotative-description`	Support, General & Upper	Catalogs, Environement, Events, Government, Multimedia
`https://w3id.org/arco/ontology/cultural-event`	Events, society	Events, Catalogs, Goverement, Multimedia
`https://w3id.org/arco/ontology/location`	Geography, Geometry	Catalogs, Events, Government, Multimedia
`https://w3id.org/arco/ontology/arco`	General & Upper	Catalogs, Environment, Events, Government, Multimedia
`https://w3id.org/cocoon/v1.0`	Services, Contracts	Industry, Services

As the main goal of the system is to suggest recommendation to a curator, we compute a soft accuracy metric, corresponding to the number of graph with at least one match between one of the curator tags and the classifier suggestions, divided by the total number of tested vocabularies.

For a vocabulary i, its associated tags $y_i = \{y_{i1}, y_{i2}, ..., y_{il}\}$ and the prediction of the classifier $y_i^{pred} = \left\{y_{i1}^{pred}, y_{i2}^{pred}, ..., y_{im}^{pred}\right\}$, we say that the classifier is *softly accurate* for the vocabulary i if $\exists y_{ik}^{pred} \in y_i^{pred}$ such that $y_{ik}^{pred} \in y_i$. The soft accuracy is then computed by the ratio of the number p of outputs softly accurate on the total number n of inputs. We get a result of 0.33 for this evaluation.

5.2 Discussion

The results seem average regarding the precision in the detection from the classifier, compared to the curator. Their could be several explanations, like the disparity between the tags in the dataset (13 labels are used in less than 10 vocabularies), or the difference of subjects in vocabularies tagged by the same label. For example, the "geography" tag is used for the `rooms` and the `Postcode` [8] ontologies, whereas they both describe completely different things, thus we can expect different words usage and very different feature vectors.

Furthermore, multi-label classification for tagging recommendation is a hard task, especially when the number of possible tag is high (43) and the number of examples is low (666) [5] like in this particular setting . It has been demonstrated that SVM classifiers work well for text classification problem, however their performance decrease strongly as the number of labels increases [6]. The list of domains grows depending on the need and some have a more organizational function. For example, LOV curators introduced the IOT tag to group all the vocabularies related to the IoT domain. Historically, some of the tags are related to W3C vocabularies recommendations (W3C Rec).

6 Conclusion and Future Work

This paper addresses one main issue: build and evaluate a classifier based on the content of LOV catalog using machine learning technique. The final goal of this work is to help the human curator of vocabularies to have a list of recommendations for a new ontology submitted in the back-end. The classifier implemented gives a micro F1-score of 36%. Although this score seems low, the system will not be used without a human that validates or not the suggested tag. We do not intend to compare the system with the human curator. Instead, we want to have a system that reduce possible risk of bias when assigning domains to vocabularies and suggest tags to the curator. Future work includes ingesting the feedback from the curators into the classifier to learn from newly added vocabularies for a continuous learning workflow, and test deep learning models with a transfer learning strategy to overcome the low-frequency of training examples.

[8] `https://lov.linkeddata.es/dataset/lov/vocabs/postcode`

Indeed, deep learning approach can perform well on multi-label classification, but it needs a lot of training examples [8].

References

1. M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut. A brief survey of text mining: Classification, clustering and extraction techniques. *arXiv preprint arXiv:1707.02919*, 2017.
2. P. F. Evangelista, M. J. Embrechts, and B. K. Szymanski. Taming the curse of dimensionality in kernels and novelty detection. In *Applied soft computing technologies: The challenge of complexity*, pages 425–438. Springer, 2006.
3. N. Halko, P.-G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.
4. T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer, 1998.
5. I. Katakis, G. Tsoumakas, and I. Vlahavas. Multilabel text classification for automated tag suggestion. In *Proceedings of the ECML/PKDD*, volume 18, 2008.
6. T.-Y. Liu, Y. Yang, H. Wan, H.-J. Zeng, Z. Chen, and W.-Y. Ma. Support vector machines classification with a very large-scale taxonomy. *Acm Sigkdd Explorations Newsletter*, 7(1):36–43, 2005.
7. R. Meusel, B. Spahiu, C. Bizer, and H. Paulheim. Towards automatic topical classification of lod datasets. In *Workshop on Linked Data on the Web, LDOW-co-located with the 24th International World Wide Web Conference, WWW 19 may*, volume 1409, 2015.
8. J. Nam, J. Kim, E. L. Mencía, I. Gurevych, and J. Fürnkranz. Large-scale multi-label text classification—revisiting neural networks. In *Joint european conference on machine learning and knowledge discovery in databases*, pages 437–452. Springer, 2014.
9. M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008.
10. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
11. D. M. Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. 2011.
12. M. E. Ruiz and P. Srinivasan. Automatic text categorization using neural networks. In *Proceedings of the 8th ASIS SIG/CR Workshop on Classification Research*, pages 59–72, 1998.
13. F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.
14. F. Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.
15. B. Spahiu, A. Maurino, and R. Meusel. Topic profiling benchmarks in the linked open data cloud: Issues and lessons learned. *Semantic Web*, (Preprint):1–20, 2019.

16. P.-Y. Vandenbussche, G. A. Atemezing, M. Poveda-Villalón, and B. Vatant. Linked open vocabularies (lov): a gateway to reusable semantic vocabularies on the web. *Semantic Web*, 8(3):437–452, 2017.
17. Z. Wang, J. Zhang, J. Feng, and Z. Chen. Knowledge graph embedding by translating on hyperplanes. In *Twenty-Eighth AAAI conference on artificial intelligence*, 2014.

On The Role of Knowledge Graphs in Explainable AI

Freddy Lecue

CortAIx (Centre of Research & Technology in Artificial Intelligence eXpertise)
Montréal, Kanada
freddy.lecue@inria.fr

Abstract. The current hype of Artificial Intelligence (AI) mostly refers to the success of machine learning and its sub-domain of deep learning. However, AI is also about other areas, such as Knowledge Representation and Reasoning, or Distributed AI, i.e., areas that need to be combined to reach the level of intelligence initially envisioned in the 1950s. Explainable AI (XAI) now refers to the core backup for industry to apply AI in products at scale, particularly for industries operating with critical systems. XAI can not only be reviewed from a Machine Learning perspective, but also from the other AI research areas, such as AI Planning or Constraint Satisfaction and Search. We expose the XAI challenges of AI fields, their existing approaches, limitations and the great opportunities for Semantic Web Technologies and Knowledge Graphs to push the boundaries of XAI further.

Semantic Web Technologies for Explainable Machine Learning Models: A Literature Review

Arne Seeliger[1,2] (✉), Matthias Pfaff[1], and Helmut Krcmar[2]

[1] fortiss, Research Institute of the Free State of Bavaria associated with Technical University of Munich, Guerickestr. 25, 80805 Munich, Germany
seeliger@fortiss.org
[2] Technical University of Munich, Boltzmannstr. 3, 85748 Garching, Germany

Abstract. Due to their tremendous potential in predictive tasks, Machine Learning techniques such as Artificial Neural Networks have received great attention from both research and practice. However, often these models do not provide explainable outcomes which is a crucial requirement in many high stakes domains such as health care or transport. Regarding explainability, Semantic Web Technologies offer semantically interpretable tools which allow reasoning on knowledge bases. Hence, the question arises how Semantic Web Technologies and related concepts can facilitate explanations in Machine Learning systems. To address this topic, we present current approaches of combining Machine Learning with Semantic Web Technologies in the context of model explainability based on a systematic literature review. In doing so, we also highlight domains and applications driving the research field and discuss the ways in which explanations are given to the user. Drawing upon these insights, we suggest directions for further research on combining Semantic Web Technologies with Machine Learning.

Keywords: Semantic Web Technologies · Machine Learning · Explainability · XAI.

1 Introduction

Artificial Intelligence (AI) and Machine Learning (ML) techniques in particular have had tremendous success in various tasks including medical diagnosis, credit card fraud detection, or face recognition [11]. These systems, however, are often opaque and usually do not provide human-understandable explanations for their predictions [23]. This situation is problematic because it can adversely affect the understanding, trust, and management of ML algorithms [23]. While not every (benign) algorithmic decision needs to be explained in detail, explainability is necessary when dealing with incomplete problem statements including aspects of safety, ethics, or trade-offs [18]. Additionally, legal considerations of AI accountability add to the relevance of explainable decision systems [19].

The umbrella term *Explainable Artificial Intelligence* (XAI) is often used in academia to refer to a variety of approaches attempting to make ML methods explainable, transparent, interpretable, or comprehensible. Due to its relevance a plethora of research on XAI exists, including literature reviews of popular methods and techniques (see [2] or [22] for example). However, many of those approaches rely on a purely technical analysis of the black-box ML models. For such approaches Cherkassky and Dhar [14] argue that model explainability cannot be achieved. The authors further stipulate that explainability is highly dependent on the usage of domain knowledge and not data analysis alone. This idea has been adapted more recently by different authors arguing that the incorporation of Semantic Web Technologies might be a key to achieve truly explainable AI-systems [26, 27]. Since existing surveys on XAI have not explored this promising avenue of research in detail, we provide a literature-based overview of the usage of Semantic Web Technologies alongside ML methods in order to facilitate explainability. Specifically, we focus on addressing three research questions:

1. What combinations of Semantic Web Technologies and ML have been proposed to enhance model explainability?
2. Which domains of applications and tasks are especially important to this research field?
3. How are model explanations evaluated and presented to the user?

The remainder of this paper is organized as follows. Section 2 provides relevant background information pertaining to explainability of ML systems. Subsequently, Section 3 briefly describes the research design before presenting the main findings of this research. Based on these insights, implications for future research are presented in Section 4. Finally, Section 5 concludes this research.

2 Background and Scope of the Literature Review

Explainability of Artificial Intelligence is not a new stream of inquiry. Mueller et al. [39] analyzed the temporal development of XAI and showed that the topic has been intensively studied from the 1970s to the early 1990s within the context of Expert and Tutoring Systems. In the following two decades, only little research has been produced in the field. Recently, however, there has been a resurgence of the topic due to the interest in Machine Learning and Deep Learning [39].

Despite recent frequent publications on the topic of XAI there is no agreement upon a definition of explainability [34]. For the purpose of this survey, we follow Adadi and Berrada [2] in differentiating *interpretable* systems which allow users to study the (mathematical) mapping from inputs to outputs from *explainable* systems which provide understanding of the system's work logic. In this context, Doran et al. [17] postulate that truly explainable systems need to incorporate elements of reasoning which make use of knowledge bases in order to create human-understandablable, yet unbiased explanations. Furthermore, it is worth mentioning that interpretability or explainability not only depends on a specific model but also the knowledge and skills of its users [24].

Within the domain of ML a number of surveys address the topic of explainability and interpretability. For example, Biran and Cotton [10] review algorithmic and mathematical methods of interpretable ML models, Abdul et al. [1] focus on explanations from a human-centered perspective, and Adadi and Berrada [2] provide a holistic survey which also covers aspects of evaluation and perception. However, these studies often do not touch upon how tools such as Semantic Web Technologies might foster ML system explainability. In contrast, within the related field of Data Mining and Knowledge Discovery the interpretation of data patterns via Semantic Web and Linked Open Data has been described in a detailed survey by Ristoski and Paulheim [45]. While Data Mining, Knowledge Discovery, and ML certainly overlap in some areas, a clear overview of the combination of Semantic Technologies and Machine Learning is still missing. In this context it is worth mentioning that the scope of this review is on classical ML techniques as opposed to fields such as Inductive Logic Programming (ILP) [40]. ILP combines ideas from ML (learning from positive and negative examples) with logical programming in order to derive a set of interpretable logical rules. The interested reader can find a summary of how ontologies can be used in the ILP framework in [35]. While some researchers see ILP as a subcategory of ML (e.g. [50]), we follow Kazmi et al. [30] in differentiating the two fields and focus on more classical ML while touching upon ILP only briefly.

3 Explainable Machine Learning Models through Semantic Web Technologies

In this section we briefly lay out the research design of this survey before summarizing the insights of the conducted analysis. To answer the posed research questions we carried out an extensive literature review [58] by searching major academic databases including ACM Digital Library, SCOPUS, and peer-reviewed pre-prints on arXiv. The latter has been incorporated because XAI is a dynamically evolving field with a number of contributions stemming from ongoing work. We conducted a search based on keywords relating to three categories: Machine Learning, Semantic Web Technologies, and explainability.[1] The resulting list of papers was evaluated for relevance based on their abstracts and the remaining papers based on their full content. A forward and backward search [59] has been conducted to complement the list of relevant research articles.

To shed light on the first research question, we categorized the relevant models based on their usage of ML and Semantic Web Technologies. Specifically, we distinguished ML approaches along their learning rules (supervised, unsupervised, reinforcement learning) [48] and characterized the used Semantic Web Technologies by their semantic expressiveness. In doing so, we focused on the actually exploited knowledge rather than the underlying representation. For example, if a system incorporates an ontology but exclusively makes use of taxo-

[1] Search strings included but were not limited to: "machine learning" OR "deep learning" OR "data mining"; "explanation*" OR "interpret*" OR "transparen*"; "Semantic Web" OR "ontolog*" OR "background knowledge" OR "knowledge graph*"

nomical knowledge, it is categorized as a taxonomy. We followed Sarker et al. [47] in differentiating knowledge graphs from ontologies insofar that the former are usually a set of triples most often expressed using the Resource Description Framework (RDF) while the latter additionally posses type logics and are regularly expressed using Web Ontology Language (OWL). We addressed the second research question by observing the application domains and tasks of the analyzed systems. We provide answers to the third research question by describing in what form explanations are given to the user and how their quality is assessed.

3.1 Combining Semantic Web Technologies with Machine Learning

The results of categorizing the relevant literature along the dimensions laid out before are presented in Table 1. From a general point of view, one can observe that Semantic Web Technologies are used primarily to make two types of ML models explainable: supervised classification tasks using Neural Networks and unsupervised embedding tasks. The Semantic Web Technologies utilized alongside Neural Networks are quite diverse, while embedding methods usually incorporate knowledge graphs. Further, systems which attempt to enhance the explainability of ML systems agnostic of the underlying algorithms mainly harness ontologies and knowledge graphs. Table 1 also illustrates that only one of the reviewed articles covers reinforcement learning. In the following paragraphs we present more in-depth findings for each type of ML approach.

Concerning **supervised learning** (classification) techniques, Table 1 illustrates that Neural Networks are the dominant prediction model. The architectures proposed are manifold and include, among others, recurrent (e.g. [16, 57]) and convolutional (e.g. [13]) networks as well as autoencoders (e.g. [5, 6]). In combining these models with Semantic Web Technologies one approach is to map network inputs or neurons to classes of an ontology or entities of a knowledge graph. For example, Sarker et al. [47] map scene objects within images to classes of the Suggested Upper Merged Ontology. Based on the image classification outputted by the Neural Network, the authors run DL-Learner on the ontology to create class expressions that act as explanations. Similarly, in the work of [56], image contents are extracted as RDF triples and then matched to DBpedia via the predicate *same-concept*. In order to answer questions provided by the user about an image, the system translates each question into a SPARQL query which is run over the combined knowledge base. The results of this operation are then used to give an answer and substantiate it with further evidence that acts as an explanation. A related approach is used in [21] to explain image recognition on classes that have not been part of any training data (zero-shot learning). Furthermore, Selvaraju et al. [49] learn a mapping between individual neurons and domain knowledge. This enables the linking of a neuron's weight (importance) to semantically grounded domain knowledge. Another common approach within the supervised classification group is to utilize the taxonomical information of a knowledge base. These hierarchical relationships aid the explanation generation in different ways. For instance, Choi et al. [15] and Ma et al. [36] design attention mechanisms while authors such as Che et al. [12] and

Table 1. Overview of Reviewed Articles

Author	Machine Learning Technique						Semantic Expressiveness			
	Supervised *		Unsupervised		Reinforcement	Multiple				
	Neural Network	Other	Clustering	Embedding	MDP **		Ontology	Knowledge Graph	Taxonomy	Glossary/Lexicon
Aditya et al. [3]				x				x		
Ai et al. [4]				x				x		
Alirezaie et al. [5, 6]	x						x			
Batet et al. [7, 8]			x						x	
Bellini et al. [9]				x				x		
Che et al. [12]	x								x	
Chen et al. [13]	x						x			
Choi et al. [15]	x			x					x	
Clos et al. [16]	x									x
Geng et al. [21]	x						x			
Gusmão et al. [24]				x				x		
Huang et al. [28]				x				x		
Jiang et al. [29]		x							x	
Khan et al. [31]					x		x			
Krishnan et al. [32]						x	x			
Liao et al. [33]				x					x	
Ma et al. [36]	x			x					x	
Ma et al. [37]				x				x		
McGuinness et al. [38]						x	x			
Musto et al. [41]						x		x		
New et al. [42]						x	x			
Publio et al. [43]						x	x			
Racoceanu & Capron [44]						x	x			
Sarker et al. [47]	x						x			
Selvaraju et al. [49]	x									x
Tiddi et al. [50, 51]			x					x		
van Engelen et al. [52]						x		x		
Wan et al. [54]		x							x	
Wang et al. [55]				x				x		
Wang et al. [56]	x						x			
Wang et al. [57]	x							x		
Yan et al. [60]	x						x			
Zhang et al. [61]				x				x		

* Supervised learning comprises of classification approaches only because in this review regression models were only used in systems developed for multiple techniques.
** Markovian Decision Process (MDP)

Jiang et al. [29] employ model regularization based on this domain knowledge. It should be noted, however, that these systems focus more on interpretability than explainability. Since these approaches are often found in the health care domain they are more thoroughly discussed in Section 3.2.

Regarding **unsupervised learning**, we identified two groups within the reviewed literature. As shown in Table 1, a significant body of research aims at creating explainable embeddings of or with knowledge graphs. For the most part these approaches are part of some recommendation engine and are thus explained in more detail in Section 3.2. Apart from these, a smaller number of scholars strive to increase the level of interpretability or explainability for clustering algorithms. Batet et al. [7] use the taxonomical knowledge encoded in WordNet to derive a semantic similarity function which leads to more interpretable clusters. The authors present an extension to their work [8] which allows the incorporation and merging of multiple ontologies within their framework. However, no specific explanations are provided by the system as to how cluster membership of data points can be justified. Tiddi et al. [50, 51] go beyond semantic similarity functions and propose to explain clusters or data patterns (agnostic of the clustering algorithm) by traversing a knowledge graph to find commonalities among the clusters. The system, called Dedalo, uses ILP to generate candidate explanations based on the background knowledge and the given clusters. The former is built by dynamically following the URI links of the items in the data set. However, such a technique raises the question of explanation fidelity, thus asking whether the given explanation actually agrees with the underlying predictive model.

As stated above, only one reviewed system aims at explaining **reinforcement learning**. In this research [31] the authors utilize an ontology to incorporate domain knowledge into the explanation process of an MDP recommendation system. The ontology is used to provide information which is not available from the data alone and to perform inference to create rules which limit the number of actions recommended. Finally, Semantic Web Technologies such as ontologies can be used to aid explainability and interpretability from a more general and **model agnostic** point of view. Along these lines, Krishnan et al. [32] design an explainable personal assistant that uses an ontology to dynamically grow a knowledge base, interact with other modules, and perform reasoning. In addition, Racoceanu and Capron [44] design a medical imaging platform which provides decision reproducibility and traceability powered by an ontology. Even more general, some authors propose ontologies or interlingua to declaratively represent aspects and dimensions of explainability. For instance, McGuinness et al. [38] create three ontologies with concepts and relation about data provenance, trust, and justifications, thus offering an explanation infrastructure. Similarly, by constructing an ML schema, Publio et al. [43] aim at exposing the semantics of such systems which can positively affect model explainability.

Lastly, we want to highlight another insight relating to the performance of the explainable systems. It is worth noting that in using Semantic Web Technologies alongside ML algorithms, explainability is not raised at the cost of performance. Rather, the reviewed systems often achieve state-of-the-art performance in their respective tasks. This is particularly notable because these results exemplify how to overcome the often assumed trade-off between ML accuracy and interpretability by the means of structure and logic [46].

3.2 Domains and Applications

The combinations of ML algorithms and Semantic Web Technologies are also driven by the respective application domains and tasks to be accomplished. Table 2 provides an overview of the most frequent domains and tasks of the reviewed systems. Regarding the former, it becomes apparent that – while many systems are developed agnostic of a specific domain – health care is a strong driver for interpretable ML systems. Regarding the tasks of the reviewed systems, we found the recommendation task and image analysis to be of great importance. For brevity we limit the following paragraphs to the health care domain and the recommendation task.

Table 2. Selected Domains of Application and Tasks

	Tasks and Domains	Authors
Domains	General	[3], [16], [24], [28], [38], [43], [47], [50–52], [55, 56], [61]
	Health Care	[12], [15], [29], [36], [42], [44], [54], [60]
	Entertainment	[9], [41], [57]
	Commercial	[4], [33], [37]
Tasks	Recommendation	[4], [9], [28], [31], [37], [41], [55], [57]
	Image Annotation or Classification	[5, 6], [21], [44], [47], [49], [60]
	Transfer or Zero-Shot Learning	[13], [21], [49]
	Knowledge Base Completion	[24], [52], [61]
	Diagnosis Prediction	[12], [15], [36]
	Visual Question Answering	[3], [56]

Note: Multiple selections possible.

Systems in the domain of **health care** often combine classification tasks such as diagnosis prediction with taxonomical knowledge found in medical diagnosis codes or medical ontologies. For instance, Jiang et al. [29] use the hierarchical information of the International Classification of Diseases (ICD) to introduce a regularization penalty to their logistic regression which produces a sparse model where non-zero features tend to be localized within a limited number of subtrees instead of being scattered across the entire hierarchy. This kind of feature weighting might make the algorithmic prediction process more explicit (interpretability), but it does not provide explanations and justification for laymen (e.g. patients). Similarly, Chen et al. [12] incorporate hierarchical ICD knowledge in a Neural Network architecture to regularize the output layer of the network and learn clinically relevant features. Yan et al. [60] use hierarchical relationships within an ontology to expand a set of medical labels by inferring missing parent labels. For example, the label "right mid lung" is expanded to "right lung", "lung", and "chest". The authors also utilize exclusive relationships between labels to learn hard cases and improve accuracy. When making predictions on medical images, their system is able to provide input examples similar to the given model output as prediction evidence. Finally, KAME [36] is a diagnosis prediction system inspired by [15] which uses medical ontologies to learn (embedded) representations of medical codes and their parent codes. These are

then utilized to learn input representations of patient data which are fed into a Neural Network architecture. The authors exploit an attention mechanism which learns weights that allow to interpret the importance of different pieces of knowledge. Summing up, within the domain of health care many interpretable ML models have been proposed. These mainly use taxonomical knowledge to aid performance and interpretability. The reason for the relative abundance of such systems in the health care domain stems from the high stakes characteristics of the field as well as the existence of different medical ontologies.

Due to their extensive use of knowledge graphs, **recommendation systems** are an important branch of research in the reviewed field. More specifically, these systems commonly combine embedding models with knowledge graphs. For example, Bellini et al. [9] inject the DBpedia knowledge graph into an autoencoder network which is constructed to mirror the structure of the knowledge base. After training such a system for each user, the learned weights map to explicit semantic concepts from the knowledge graph and user-specific explanations can be generated based on these insights. Another special case of embedding is RippleNet [55] where the triples of a constructed knowledge graph (based on Microsoft Satori) are iteratively compared to the embeddings and then propagated. This way the path from a user's history to a recommended item can be used as an explanation for the recommendation. Further, there are approaches which use Semantic Web Technologies agnostic of the underlying recommendation algorithm. One such system is ExpLOD [41] which makes use of the Linked Open Data paradigm. The framework first maps liked items and recommended items into a knowledge base such as DBpedia, then builds a graph, ranks the properties in this graph based on relevance, and finally creates a natural language explanation from the top properties retrieved. While being model agnostic, the issue of explanation fidelity can be raised again here because the given explanation might not correspond to the actual underlying model process. Finally, it is worth mentioning that explainability in recommender systems is mainly driven from a user-centric perspective with the aim to increase user satisfaction and acceptance.

3.3 Explanation Forms and Evaluation

The conducted analysis revealed that the presentation and form of the given explanations is highly diverse – even within similar domains or prediction tasks. For example, some scholars combine different types of explanations (e.g. visual and textual [49]) in order to increase explainability while others provide only minimal explanation towards the user (e.g. [7] or [52]). Moreover, only few authors present explanations in natural language. For instance, Musto et al. [41] incorporate a dedicated natural language generator into their recommendation algorithm. The authors utilize a template-based approach which is also used by other authors [4, 31]. A more frequently employed explanation form consists of textual (semi)-logical or rule-like notation. Further, explanations are usually designed to optimally justify correct model output. One deviation from this is the work of Alirezaie et al. [5, 6] where the *errors* of a Neural Network image classifier are explained by performing ontological reasoning upon objects of a

scene. To illustrate the range of explanation forms used, Table 3 provides selected examples of textual explanations encountered in this review. Apart from the ambiguity of the term explainability, one potential reason for this diversity includes the relevancy of an explanation for a given system: While in most reviewed cases, explainability is an explicit goal, in a subset of models, explainability is treated as a secondary goal and Semantic Web Technologies are used to primarily address other issues such as data sparseness (e.g. [15]).

Table 3. Examples of Textual Explanations

Author	Task	Example Explanation
Bellini et al. [9]	Recommendation of a movie	**Prediction:** Terminator 2 **Explanation:** We guess you would like to watch Terminator 2: Judgment Day (1991) more than Transformers: Revenge of the Fallen (2009) because you may prefer: • (subject) 1990s science fiction films [...] over: • (subject) Films set in Egypt [...]
Gusmão et al. [24]	Knowledge graph completion (triple prediction)	**Prediction:** Head: francis_ii_of_the_two_sicilies, Relation: RELIGION, Tail: roman_catholic_church **Explanation:** #1: parents, religion #2: spouse^{-1}, religion [...]
Selvaraju et al. [49]	Image classification of an animal	**Prediction:** Yellow-headed blackbird **Explanation:** has_eye_color = black, has_underparts_color = white, has_belly_color = white, has_breast_color = white, has_breast_pattern = solid
Zhang et al. [61]	Knowledge graph completion (link prediction)	**Prediction:** World War I – entity envolved – German Empire **Explanation:** World War I – commanders – Erich Ludendorff Erich Ludendorff – commands – German Empire Supported by: Falkland Wars – entities envolved – United Kingdom Falkland Wars – commanders – Margaret Thatcher Margaret Thatcher – commands – United Kingdom

Note: Some explanations have been shortened for legibility as indicated by square brackets.

Furthermore, we found most systems to offer rather static explanations without much user interaction. In this context, the work of Liao et al. [33] is an exception as the proposed recommendation system enables user-feedback on human-interpretable domain concepts. Moreover, looking into the future, Sarker et al. [47] envision their explanation tool for image classification to be used in an interactive human-in-the-loop system where a human monitor can correct algorithmic decisions based on the given explanations. On the whole, however, we notice a lack of user-adaptive or interactive explanation approaches in the reviewed literature.

Finally, when it comes to evaluating the goodness of the explanations, only few authors go beyond a subjective assessment of the proposed system. Bellini et al. [9], for instance, perform an evaluation of their knowledge-aware autoencoder recommendation system by conducting A/B testing with 892 volunteers. Similarly, Musto et al. [41] designed a user study in which 308 subjects filled out a questionnaire involving questions such as *"I understood why this movie was recommended to me"*. Through this evaluation, the authors gain further insights into different aspects of how their explanation system affects end users. Other authors propose more quantitative evaluation metrics to determine the goodness of the given explanations. Zhang et al. [61] explain their link predictions by finding patterns within a knowledge graph which are similar to the predicted ones (see Table 3) and measure explanation reliability by the number of similar patterns found. Further, Jiang et al. [29] measure the interpretability of their predictive system by quantifying the sparseness of their linear model while taking into account the taxonomical structure of their data. Overall, from these findings it becomes obvious that there is no accepted standard for evaluating explanations within XAI.

4 Trends for Future Research

Based on our review of the relevant literature we articulate opportunities and challenges for future research in the field. We generate these insights based on our analysis and comparison among all reviewed papers as well as on the basis of the challenges put forward within each of the articles.

4.1 Semantic Web Technologies for Explainability

The combination of Semantic Web Technologies and ML offers great potential for facilitating explainable models. We identified the matching of ML data with knowledge base entities – which has been called *knowledge matching* [21] – as one central challenge which needs to be overcome by future research. Specifically, automated and reliable methods for knowledge matching are required. In this context, Wang et al. [56] suggest string matching between identified objects and ontology classes and Liao et al. [33] propose to mine concepts and relationships automatically from online sources. Further research in this area as well as related fields like semantic annotation are needed to enable effective and efficient knowledge matching.

Moreover, we found a certain concentration on specific ML techniques and Semantic Web Technologies. More work needs to be conducted on explainable reinforcement learning and clustering. In this context, we also note that the work across different disciplines and tasks still remains somewhat isolated even though concepts like linked data provide the tools for integrating various domains. Some existing research acknowledges the need to extend the range of tasks performed by explainable systems [12] or their domains of application [32]. Other authors envision the use of more data [60] or more complex background knowledge [41,

42, 47]. Hence, the areas of ontology or knowledge graph learning as well as knowledge base matching play an important role in accomplishing this goal. Future work will therefore need to find ways to mitigate the potential lack of data interconnectedness and the increased complexity of such systems.

Finally, we highlight the need for future work to aim for truly explainable systems which incorporate reasoning and external knowledge that is human-understandable. To achieve this goal, future explanation systems need to ensure that the explanations given are truthful to the underlying ML algorithm. Further, such approaches should be able to explain not only how an output relates to some representation of interest but also how this representation has been obtained. For example, it is not enough to justify that a human face has been detected by stating that eyes, mouth, and nose were recognized and that these features are part of a human face (e.g. inferred via ontology). A truly explainable system should also be able to explain why these features have been recognized. This point relates to the question of user interaction, which is discussed below.

4.2 Human-Centric Explanations

Since explanations are forms of social interactions [2], their efficacy and quality depend to a large extent on their intelligibility and comprehensibility as perceived by the user. In other words, an explanation is only useful if the user is able to understand it. In this review we have shown that the form and appearance of explanations differs significantly among current systems and many of those do not provide explanations in natural language. Therefore, we believe that the field of Natural Language Processing (NLP) and Natural Language Generation (NLG) in particular offers a useful starting point. For example, Vougiouklis et al. [53] generate natural texts from Semantic Web triples using Neural Networks. Moreover, Ell et al. [20] translate SPARQL queries to English text that is understandable by non-experts. More generally, the field of (Visual) Question Answering can be a source of inspiration since questions and answers are usually given in natural language [56].

Additionally, we believe that explanations need to be adaptive and interactive in order to generate the greatest benefit for the user. Structured knowledge bases could allow users to scrutinize and interact with explanations in various forms. For example, user could browse among different possible explanations or drill down on a specific explanation to extract more specific reasons that contributed to a prediction. Khan et al. [31] envision a system that allows for such follow up questions. Similarly, Bellini et al. [9] plan to incorporate the possibility for users to correct their system in a continuous loop. As described above, Sarker et al. [47] also regard this course of action as an important task for future studies. However, there seems to be no consensus regarding the actual mode of interaction. In order to find optimal ways of presenting and interacting with explanations, future research needs to incorporate findings from a greater variety of research fields. Existing studies [1, 2] show that there is a growing body of diverse and interdisciplinary work addressing the question of human-understandable explanations that can be leveraged in this context.

4.3 Common Grounds for Evaluation

We believe that meaningful progress in the field of XAI is not only dependent on novel explanation algorithms but also on common grounds for model evaluation and comparison. In light of this, Doshi-Velez and Kim [18] put forward the need for a shared language relating to factors of ML explainability. We have shown that Semantic Web Technologies can help in creating such a common lingua. Future work, however, needs to proof how to utilize such constructs effectively in the context of explainability. Another way forward could be to develop and rely on standard design patterns for combining ML with Semantic Web Technologies. The work of van Harmelen and ten Teije [25] already provides a collection of patterns for such hybrid systems. Moreover, common evaluation criteria need to be established so that subjective assessments of model explainability can be replaced by more rigorous practices.

5 Conclusion

Explainability and interpretability have become an essential requirement for many ML systems. In this work, through an extensive literature review, we have shown that the connection between ML and Semantic Web Technologies can yield exciting opportunities regarding model explainability. We discussed the most prevalent approaches within supervised and unsupervised learning and highlighted how the domain of health care and the recommendation task are important drivers of the research field. The literature analysis further revealed that prediction performance is not reduced but often increased by incorporating background knowledge within the ML paradigm. Finally, we provided examples of specific forms of explanations including natural language and rule-like statements. At the same time, we highlighted that meaningful progress in the reviewed field also relies on advances in a number of research challenges. These include technical questions like automated ways of knowledge matching or progress in knowledge base learning. Other challenges concern the development of adaptive and interactive systems. Lastly, more rigorous evaluation strategies need to be devised by future research. We believe that tackling these questions and further exploring the combination of structured knowledge, reasoning, and Machine Learning can pave the way to truly explainable systems.

References

1. Abdul, A., Vermeulen, J., Wang, D., Lim, B.Y., Kankanhalli, M.: Trends and trajectories for explainable, accountable and intelligible systems: An HCI research agenda. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. pp. 582:1–582:18. CHI '18, ACM, New York, NY, USA (2018)
2. Adadi, A., Berrada, M.: Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). IEEE Access **6**, 52138–52160 (2018)

3. Aditya, S., Yang, Y., Baral, C.: Explicit reasoning over end-to-end neural architectures for visual question answering. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence. New Orleans, Louisiana, USA (2018)
4. Ai, Q., Azizi, V., Chen, X., Zhang, Y.: Learning heterogeneous knowledge base embeddings for explainable recommendation. Algorithms **11**(9), 137 (2018)
5. Alirezaie, M., Längkvist, M., Sioutis, M., Loutfi, A.: A symbolic approach for explaining errors in image classification tasks. In: IJCAI Workshop on Learning and Reasoning. Stockholm, Sweden (2018)
6. Alirezaie, M., Längkvist, M., Sioutis, M., Loutfi, A.: Semantic Referee: A neural-symbolic framework for enhancing geospatial semantic segmentation. Semantic Web Journal (2019)
7. Batet, M., Valls, A., Gibert, K.: Performance of ontology-based semantic similarities in clustering. In: Proceedings of the 10th International Conference on Artificial Intelligence and Soft Computing. pp. 281–288. Springer, Berlin, Heidelberg (2010)
8. Batet, M., Valls, A., Gibert, K., Sánchez, D.: Semantic clustering using multiple ontologies. In: Artificial Intelligence Research and Development - Proceedings of the 13th International Conference of the Catalan Association for Artificial Intelligence. pp. 207–216. IOS Press, Amsterdam, The Netherlands (2010)
9. Bellini, V., Schiavone, A., Di Noia, T., Ragone, A., Di Sciascio, E.: Knowledge-aware autoencoders for explainable recommender systems. In: Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems. pp. 24–31. DLRS 2018, ACM, New York, NY, USA (2018)
10. Biran, O., Cotton, C.: Explanation and justification in machine learning: A survey. In: Proceedings of the IJCAI-17 Workshop on Explainable AI (XAI). pp. 8–13. Melbourne, Australia (2017)
11. Brynjolfsson, E., Mitchell, T.: What can machine learning do? Workforce implications. Science **358**(6370), 1530–1534 (2017)
12. Che, Z., Kale, D., Li, W., Bahadori, M.T., Liu, Y.: Deep computational phenotyping. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 507–516. KDD '15, ACM, New York, NY, USA (2015)
13. Chen, J., Lecue, F., Pan, J.Z., Horrocks, I., Chen, H.: Knowledge-based transfer learning explanation. In: Sixteenth International Conference on Principles of Knowledge Representation and Reasoning. pp. 349–358. Tempe, AZ, USA (2018)
14. Cherkassky, V., Dhar, S.: Interpretation of black-box predictive models. In: Measures of Complexity, pp. 267–286. Springer, New York (2015)
15. Choi, E., Bahadori, M.T., Song, L., Stewart, W.F., Sun, J.: GRAM: Graph-based attention model for healthcare representation learning. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 787–795. KDD '17, ACM, New York, NY, USA (2017)
16. Clos, J., Wiratunga, N., Massie, S.: Towards explainable text classification by jointly learning lexicon and modifier terms. In: IJCAI-17 Workshop on Explainable AI (XAI). pp. 19–23. Melbourne, Australia (2017)
17. Doran, D., Schulz, S., Besold, T.R.: What does explainable AI really mean? A new conceptualization of perspectives. In: Proceedings of the First International Workshop on Comprehensibility and Explanation in AI and ML 2017 co-located with 16th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2017). Bari, Italy (2017)
18. Doshi-Velez, F., Kim, B.: Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608 (2017)

19. Doshi-Velez, F., Kortz, M., Budish, R., Bavitz, C., Gershman, S., O'Brien, D., Schieber, S., Waldo, J., Weinberger, D., Wood, A.: Accountability of AI under the law: The role of explanation. Berkman Center Research Publication Forthcoming; Harvard Public Law Working Paper No. 18-07 (2017)
20. Ell, B., Harth, A., Simperl, E.: SPARQL query verbalization for explaining semantic search engine queries. In: Presutti, V., d'Amato, C., Gandon, F., d'Aquin, M., Staab, S., Tordai, A. (eds.) The Semantic Web: Trends and Challenges, pp. 426–441. Springer, Cham (2014)
21. Geng, Y., Chen, J., Jimenez-Ruiz, E., Chen, H.: Human-centric transfer learning explanation via knowledge graph. In: AAAI Workshop on Network Interpretability for Deep Learning. Honolulu, HI, USA (2019)
22. Gilpin, L.H., Bau, D., Yuan, B.Z., Bajwa, A., Specter, M., Kagal, L.: Explaining explanations: An overview of interpretability of Machine Learning. In: 5th International Conference on Data Science and Advanced Analytics (DSAA). pp. 80–89. IEEE, Turin, Italy (2018)
23. Gunning, D.: Explainable artificial intelligence (XAI). Defense Advanced Research Projects Agency (DARPA) (2017)
24. Gusmão, A.C., Correia, A.H.C., De Bona, G., Cozman, F.G.: Interpreting embedding models of knowledge bases: A pedagogical approach. In: ICML Workshop on Human Interpretability in Machine Learning (WHI). Stockholm, Sweden (2018)
25. van Harmelen, F., ten Teije, A.: A boxology of design patterns for hybrid learning and reasoning systems. Journal of Web Engineering **18**(1), 97–124 (2019)
26. Holzinger, A., Biemann, C., Pattichis, C.S., Kell, D.B.: What do we need to build explainable AI systems for the medical domain? arXiv preprint arXiv:1712.09923 (2017)
27. Holzinger, A., Kieseberg, P., Weippl, E., Tjoa, A.M.: Current advances, trends and challenges of machine learning and knowledge extraction: From machine learning to explainable AI. In: Holzinger, A., Kieseberg, P., Tjoa, A.M., Weippl, E. (eds.) Machine Learning and Knowledge Extraction. pp. 1–8. Springer, Cham (2018)
28. Huang, J., Zhao, W.X., Dou, H., Wen, J.R., Chang, E.Y.: Improving sequential recommendation with knowledge-enhanced memory networks. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. pp. 505–514. SIGIR '18, ACM, New York, NY, USA (2018)
29. Jiang, J., Chandola, V., Hewner, S.: Tree-based regularization for interpretable readmission prediction. In: AAAI 2019 Spring Symposium on Combining Machine Learning with Knowledge Engineering (AAAI-MAKE). Palo Alto, CL, USA (2019)
30. Kazmi, M., Schller, P., Saygn, Y.: Improving scalability of inductive logic programming via pruning and best-effort optimisation. Expert Systems with Application **87**(C), 291–303 (2017)
31. Khan, O.Z., Poupart, P., Black, J.P.: Explaining recommendations generated by MDPs. In: Proceedings of the Third International Conference on Explanation-aware Computing. EXACT'08, vol. 391, pp. 13–24. CEUR-WS, Aachen, Germany (2008)
32. Krishnan, J., Coronado, P., Reed, T.: Seva: A systems engineer's virtual assistant. In: AAAI 2019 Spring Symposium on Combining Machine Learning with Knowledge Engineering (AAAI-MAKE). Palo Alto, CL, USA (2019)
33. Liao, L., He, X., Zhao, B., Ngo, C.W., Chua, T.S.: Interpretable multimodal retrieval for fashion products. In: Proceedings of the 26th ACM International Conference on Multimedia. pp. 1571–1579. MM '18, ACM, New York, NY, USA (2018)
34. Lipton, Z.C.: The mythos of model interpretability. Queue **16**(3), 30:31–30:57 (2018)

35. Lisi, F.A., Esposito, F.: On ontologies as prior conceptual knowledge in inductive logic programming. In: Berendt, B., Mladenič, D., de Gemmis, M., Semeraro, G., Spiliopoulou, M., Stumme, G., Svátek, V., Železný, F. (eds.) Knowledge Discovery Enhanced with Semantic and Social Information, pp. 3–17. Springer, Berlin, Heidelberg (2009)
36. Ma, F., You, Q., Xiao, H., Chitta, R., Zhou, J., Gao, J.: KAME: Knowledge-based attention model for diagnosis prediction in healthcare. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management. pp. 743–752. CIKM '18, ACM, New York, NY, USA (2018)
37. Ma, W., Zhang, M., Cao, Y., Jin, W., Wang, C., Liu, Y., Ma, S., Ren, X.: Jointly learning explainable rules for recommendation with knowledge graph. In: The World Wide Web Conference. pp. 1210–1221. WWW '19, ACM, New York, NY, USA (2019)
38. McGuinness, D.L., Ding, L., Da Silva, P.P., Chang, C.: PML 2: A modular explanation interlingua. In: AAAI 2007 Workshop on Explanation-aware Computing. pp. 49–55. Vancouver, Canada (2007)
39. Mueller, S.T., Hoffman, R.R., Clancey, W.J., Emrey, A., Klein, G.: Explanation in human-AI systems: A literature meta-review, synopsis of key ideas and publications, and bibliography for explainable AI. arXiv preprint arXiv:1902.01876 (2019)
40. Muggleton, S., Raedt, L.D.: Inductive logic programming: Theory and methods. Journal of Logic Programming **19**(20), 629–679 (1994)
41. Musto, C., Narducci, F., Lops, P., De Gemmis, M., Semeraro, G.: ExpLOD: A framework for explaining recommendations based on the Linked Open Data Cloud. In: Proceedings of the 10th ACM Conference on Recommender Systems. pp. 151–154. RecSys '16, ACM, New York, NY, USA (2016)
42. New, A., Rashid, S.M., Erickson, J.S., McGuinness, D.L., Bennett, K.P.: Semantically-aware population health risk analyses. In: Machine Learning for Health (ML4H) Workshop at NeurIPS. Montreal, Canada (2018)
43. Publio, G.C., Esteves, D., Ławrynowicz, A., Panov, P., Soldatova, L., Soru, T., Vanschoren, J., Zafar, H.: ML Schema: Exposing the semantics of machine learning with schemas and ontologies. In: ICML 2018 Workshop on Reproducibility in Machine Learning. Stockholm, Sweden (2018)
44. Racoceanu, D., Capron, F.: Towards semantic-driven high-content image analysis: An operational instantiation for mitosis detection in digital histopathology. Computerized Medical Imaging and Graphics **42**, 2–15 (2015)
45. Ristoski, P., Paulheim, H.: Semantic web in data mining and knowledge discovery. Web Semantics: Science, Services and Agents on the World Wide Web **36**(C), 1–22 (2016)
46. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence **1**(5), 206–215 (2019)
47. Sarker, M.K., Xie, N., Doran, D., Raymer, M., Hitzler, P.: Explaining trained neural networks with Semantic Web Technologies: First steps. In: Proceedings of the Twelfth International Workshop on Neural-Symbolic Learning and Reasoning (NeSy). London, UK (2017)
48. Sathya, R., Abraham, A.: Comparison of supervised and unsupervised learning algorithms for pattern classification. International Journal of Advanced Research in Artificial Intelligence **2**(2) (2013)

49. Selvaraju, R., Chattopadhyay, P., Elhoseiny, M., Sharma, T., Batra, D., Parikh, D., Lee, S.: Choose your neuron: Incorporating domain knowledge through neuron-importance. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. pp. 540–556. Springer, Cham (2018)
50. Tiddi, I., d'Aquin, M., Motta, E.: Dedalo: Looking for clusters explanations in a labyrinth of linked data. In: Presutti, V., d'Amato, C., Gandon, F., d'Aquin, M., Staab, S., Tordai, A. (eds.) The Semantic Web: Trends and Challenges. pp. 333–348. Springer, Cham (2014)
51. Tiddi, I., d'Aquin, M., Motta, E.: Data patterns explained with linked data. In: Bifet, A., May, M., Zadrozny, B., Gavalda, R., Pedreschi, D., Bonchi, F., Cardoso, J., Spiliopoulou, M. (eds.) Machine Learning and Knowledge Discovery in Databases. pp. 271–275. Springer, Cham (2015)
52. Van Engelen, J.E., Boekhout, H.D., Takes, F.W.: Explainable and efficient link prediction in real-world network data. In: Boström, H., Knobbe, A., Soares, C., Papapetrou, P. (eds.) Advances in Intelligent Data Analysis XV. pp. 295–307. Springer, Cham (2016)
53. Vougiouklis, P., Elsahar, H., Kaffee, L.A., Gravier, C., Laforest, F., Hare, J., Simperl, E.: Neural Wikipedian: Generating textual summaries from knowledge base triples. Journal of Web Semantics **52-53**, 1 – 15 (2018)
54. Wan, S., Mak, M.W., Kung, S.Y.: Mem-mEN: Predicting multi-functional types of membrane proteins by interpretable elastic nets. IEEE/ACM Transactions on Computational Biology and Bioinformatics **13**(4), 706–718 (2016)
55. Wang, H., Zhang, F., Wang, J., Zhao, M., Li, W., Xie, X., Guo, M.: Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In: Proceedings of the 27th International Conference on Information and Knowledge Management. pp. 417–426. CIKM '18, ACM, New York, NY, USA (2018)
56. Wang, P., Wu, Q., Shen, C., Dick, A., Van Den Henge, A.: Explicit knowledge-based reasoning for visual question answering. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence. pp. 1290–1296. IJCAI'17, AAAI Press (2017)
57. Wang, X., Wang, D., Xu, C., He, X., Cao, Y., Chua, T.S.: Explainable reasoning over knowledge graphs for recommendation. AAAI (2019)
58. Webster, J., Watson, R.T.: Analyzing the past to prepare for the future: Writing a literature review. MIS Quarterly pp. xiii–xxiii (2002)
59. Wohlin, C.: Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering. pp. 38:1–38:10. EASE '14, ACM, New York, NY, USA (2014)
60. Yan, K., Peng, Y., Sandfort, V., Bagheri, M., Lu, Z., Summers, R.M.: Holistic and comprehensive annotation of clinically significant findings on diverse ct images: Learning from radiology reports and label ontology. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, CA, USA (2019)
61. Zhang, W., Paudel, B., Zhang, W., Bernstein, A., Chen, H.: Interaction embeddings for prediction and explanation in knowledge graphs. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining. pp. 96–104. WSDM '19, ACM, New York, NY, USA (2019)

Persuasive Explanation of Reasoning Inferences on Dietary Data

Ivan Donadello[1 [0000−0002−0701−5729]], Mauro Dragoni[1 [0000−0003−0380−6571]], and Claudio Eccher[1 [0000−0001−9643−0088]]

Fondazione Bruno Kessler, Via Sommarive 18, I-38123, Trento, Italy
{donadello,dragoni,cleccher}@fbk.eu

Abstract. Explainable AI aims at building intelligent systems that are able to provide a clear, and human understandable, justification of their decisions. This holds for both rule-based and data-driven methods. In management of chronic diseases, the users of such systems are patients that follow strict dietary rules to manage such diseases. After receiving the input of the intake food, the system performs reasoning to understand whether the users follow an unhealthy behaviour. Successively, the system has to communicate the results in a clear and effective way, that is, the output message has to persuade users to follow the right dietary rules. In this paper, we address the main challenges to build such systems: i) the natural language generation of messages that explain the reasoner inconsistency; ii) the effectiveness of such messages at persuading the users. Results prove that the persuasive explanations are able to reduce the unhealthy users' behaviours.

Keywords: Explainable AI · Explainable Reasoning · Natural Language Generation · mHealth · Ontologies

1 Introduction

Explainable Artificial Intelligence (XAI) aims at explaining the algorithmic decisions of AI solutions with non-technical terms in order to make these decision trusted and easily understandable by humans [1]. This is of great interest for both Machine Learning (ML) methods and symbolic reasoning in rule engines. The explanation of a reasoning process can be very difficult, especially when a system is based on a set of complex logical axioms whose logical inferences are performed with, for example, tableau algorithms [3]. Indeed, inconsistencies in logical axioms may be not well understood by users if the system limits to just report the violated axioms. Indeed, users are generally skilled to understand neither formal languages nor the behaviour of a whole system. This is crucial for some applications, such as a power plant system where a warning message to the user must be clear and concise to avoid catastrophic consequences.

An interesting domain for XAI is healthcare, in particular the management of chronic diseases such as heart disease, cancer and diabetes. These are responsible for approximately 70% of deaths in Europe and U.S. each year and they account for about 75%

of the health spending[1]. Such chronic diseases can be largely preventable by eating healthy, exercising regularly, avoiding smoking, and receiving preventive services. Prevention would help people stay healthy, avoid or delay the onset of diseases, and keep diseases they already have far from becoming worse or debilitating; it would also help people lead productive lives and reduce the costs of public health. The challenges of an explainable system that supports users in following an healthy behaviour are: i) the ability of providing a clear and comprehensible message regarding user's behaviour, and ii) the effectiveness of the message to *persuade* the user at adopting an healthy lifestyle. This is fundamental as often people do not know the importance of following diet rules, hence they may not be sufficiently motivated to adopt healthy behaviors. Differently from the case of the power system, here the message must be persuasive and personalized in order to keep people engaged in using the system.

In this paper we present a XAI system based on logical reasoning that supports the monitoring of users' behaviors and persuades them to follow healthy lifestyles[2]. The concepts and rules of healthy behaviors are formalized as a Tbox of the HeLiS ontology [7]. This ontology is one of the most updated conceptual models formalizing dietary and physical activity domains. The axioms in HeLiS encode the Mediterranean diet rules that can be associated with user profiles. The user data about her/his dietary behavior are acquired through a user's dietary diary with the help of a smartphone application. This information populates the HeLiS Abox with logical individuals. A reasoner module (Section 3) combines knowledge and user's data (Tbox and Abox) to infer the user behavior and generates inconsistencies if the user does not follow the rules of a healthy lifestyle. Once an inconsistency, i.e., an unhealthy user behaviour, is detected the system shows the user a natural language message explaining the wrong behaviour and its consequences. This translation from a logic language to plain text comprehensible by humans leverages a computational persuasion framework [2] and Natural Language Generation (NLG) techniques [10]. The latter exploit dynamic and smart templates able to adapt to every persuasion strategy. The proposed system has been integrated into the HORUS.AI platform [8] and it has been validated with a mobile application within the pilot project *Key To Health* run into our institution. Results compare the persuasive explanations with simple notifications of inconsistencies and show that the former are able to support users in improving their adherence to dietary rules. To the best of our knowledge this is the first work that joins reasoning explanations with persuasive messages.

The rest of the paper follows with Section 2 that provides a state-of-the-art of techniques for generating explanations from reasoning inferences. Section 3 shows the reasoning process that checks if a user has a healthy dietary behaviour. Section 4 describes the developed template system for the automatic generation of natural language persuasive explanations. Section 5 presents the *Key To Health* project in which we deployed the system, whereas Section 6 shows its evaluation. Section 7 concludes the paper.

[1] http://www.who.int/nmh/publications/ncd_report_full_en.pdf

[2] This work is compliant with good research practice standards. More details at:
http://ec.europa.eu/research/participants/data/ref/fp7/89888/ethics-for-researchers_en.pdf
http://www.who.int/medicines/areas/quality_safety/safety_efficacy/gcp1.pdf

2 Related Work

Explainable Artificial Intelligence (XAI) generally relates to strategies able to provide human-understandable descriptions of learning algorithms usually perceived as black boxes by users [1]. Here, we focused on applying XAI to the results of inference processes. Within the whole XAI research area, our aim is to generate natural language explanations of logic inferences for supporting end-users in understanding the recommendations provided by intelligent systems.

One of the first user studies dealing with explanations for entailments of OWL ontologies was performed by [13]. The study investigated the effectiveness of different types of explanation for explaining unsatisfiable classes in OWL ontologies. The authors found that the subjects receiving full debugging support performed best (i.e. fastest) on the task, and that users approved of the debugging facilities. Similarly, [15] performed a user study to evaluate an explanation tool, but did not carry out any detailed analysis of the difficulty users had with understanding these explanations. While, [4] presents a user study evaluating a model exploration based approach to explanation in OWL ontologies. The study revealed that the majority of participants could solve specific tasks with the help of the developed model exploration tool, however, there was no detailed analysis of which aspects of the ontology the subjects struggled with and how they used the tool.

In order to gain an understanding of how OWL users interact with ontology axioms and constructors, the work proposed in [18] compiled a set of OWL "antipatterns". These logical and non-logical "antipatterns" correspond to the errors users frequently make in the use of OWL constructors, for example, by mis-interpreting the meaning of constructors, leading to unwanted effects (or non-effects) in the ontology. Our study of justification patterns is based on a similar idea of naturally occurring patterns in OWL ontologies, but rather than finding common errors, our aim is to identify potential aids in the ontology development process.

Besides justifications, formal proofs are considered to be the most prevalent alternative form of explanation for logic-based knowledge bases. In [17] the authors present an approach to providing proof-based explanations for entailments of the CLASSIC system. The system omits intermediate steps and provides further filtering strategies in order to generate short and simple explanations. The work proposed in [5] first introduced a proof-based explanation system for knowledge bases in the Description Logic ALC. The system generates sequent calculus style proofs using an extension of a tableaux reasoning algorithm, which are then enriched to create natural language explanations. However, there exist no user studies to explore the effectiveness of these proofs. In [14] the authors proposed several graph-based visualizations of defeasible logic proofs and present a user study in order to evaluate the impact of the different approaches. The study, testing 17 participants from a postgraduate course and research staff, is based on similar task-oriented principles as the Experiments 2 to 4 presented in this paper.

Finally, as ontologies are often considered to be technical artifacts akin to software, we may regard ontology and justification comprehension as analogous to software comprehension. There has been a significant amount of work on predicting the complexity of understanding and the ease of maintaining software. In particular, seminal work described in [16], which devised a complexity metric known as cyclomatic complexity

was based on the control flow paths through software. In [11] the author uses various syntactic measures such as program vocabulary and program length to calculate volume and difficulty of understanding of a program. The concept of a complexity model for OWL justifications builds upon the general idea of measuring software complexity; however, due to the difference in syntax and semantics, software complexity metrics are not directly applicable to OWL justifications.

In summary, there has been a wide range of approaches to explanation in the areas of ontologies, logics, and software comprehension, with some user studies that aim at evaluating the effects of supporting techniques. However, to date there have been no studies dealing directly with the impact on users' behaviors of explanations from OWL ontologies such as the one presented in this paper.

3 The KB-based Explainable Model

The explainable model implemented within the HORUS.AI platform relies on two main components: the HeLiS ontology [7] and the RDFpro [6] reasoner. The HeLiS ontology provides three main kinds of information:

Domain knowledge defines in the Tbox the concepts modeling the domain of interest. In particular, the HeLiS ontology contains knowledge about the dietary (i.e. taxonomy of food categories and food compositions) and physical activities (i.e. effort needed for accomplishing a specific activity) domains.

Monitoring knowledge defines in the Tbox the set of rules enabling the monitoring task and the detection of undesired behaviors (hereafter called "violations").

User knowledge defines in the Abox the concepts describing user profiles and the data populating the knowledge base, i.e., food consumed and activities performed by users.

An undesired behaviour given by the union of Tbox and populated Abox will trigger a logical inconsistency of the monitoring knowledge that has to be explained. In this paper, we do not present the full modeling process and the content of HeLiS. The reader can refer to [7] for a complete presentation of the ontology engineering process and of the concepts involved in the conceptualization of user's profile and of the monitoring tasks. For each food category, the HeLiS ontology defines both its associated positive and negative aspects. Such aspects are exploited by the Natural Language Generator module as described in Section 4.

The second component is the reasoner. Reasoning in HORUS.AI has the goal of verifying if user's dietary actions are consistent with the monitoring rules defined by domain experts, detecting and possibly materializing violations in the knowledge base, upon which further actions may be taken. Reasoning is triggered each time a user's profile or associated data are added or modified in the system, and also at specific points in time such as the end of a day or week, to check a user's behavior in such timespans. We implement reasoning in HORUS.AI using RDFpro [6], a tool that allows us to provide out-of-the-box OWL 2 RL reasoning, supporting the fixed point evaluation of `INSERT... WHERE...` SPARQL-like entailment rules that leverage the full

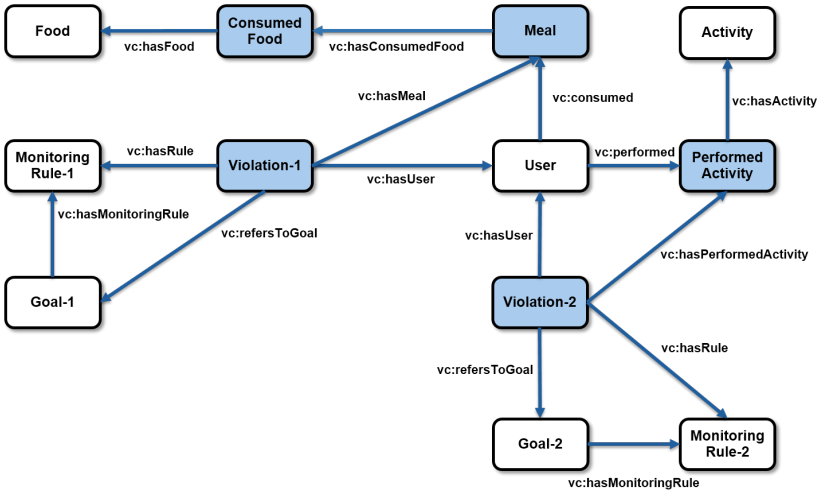


Fig. 1. Example describing how a violation individual is linked with the other HeLiS entities.

expressivity of SPARQL (e.g., `GROUP BY` aggregation, negation via `FILTER NOT EXISTS`, derivation of RDF nodes via `BIND`).

Figure 1 summarizes the knowledge graph generated by the reasoner. In order to understand the remaining of the section, individuals of type `Violation` contain all information about the unhealthy behaviors detected by the reasoner from user's data. While, individuals of type `MonitoringRule` contain information about the recommendations that users should follow. The descriptions of the other concepts are in [7].

We organize the reasoning in two phases: *offline* and *online*. The *offline* phase consists in an one-time processing of the *static* part of the ontology (monitoring rules, food, and nutrients). This is performed to materialize the ontology deductive closure, based on OWL 2 RL and some additional pre-processing rules that identify the most specific types of each `Nutrient` individual (this information greatly helps in aggregating their amounts). Whereas, during the *online* phase, each time the reasoning is triggered (e.g., a new meal is entered), the user data is merged with the closed ontology and the deductive closure of the rules is computed. This process can be performed both on a per-user basis or globally on the whole knowledge base. The resulting `Violation` individuals and their RDF descriptions are then stored back in the knowledge base. The generation of each `Violation` individual is performed in two steps. First, information inferred by aggregating the domain, monitoring, and user knowledge is used for instantiating the `Violation` individual. Second, accessory information is integrated into the `Violation` individual for supporting the Natural Language Generation module in the generation of the explanation concerning the detected violation. Accessory information includes, for example, references to other individuals of the ontology enabling the access to the positive and negative aspects associated with the food category, or the number of times that the specific rule has been violated. This kind of information

can be used for deciding the enforcement level of the persuasion contained within the generated messages.

The result of the reasoning activity is a set of structured packages containing information about the detected unhealthy behaviors. By considering as example the dietary domain, each package contains: (i) the list of meals that contributed to generate the violation; (ii) the actual quantity, for a specific food category, provided by the user; (iii) the expected quantity for the same food category; (iv) the violation level (this value gives a dimension of the violation, the higher the gap between the actual and the expected values is, the higher the value of the violation level parameter will be); and, (v) the violation history: the reasoner computes this value in order to provide a recidivism index about how a user is inclined to violate specific rules. This information, together with the identifiers of the violated rule and user, the rule priority, and the reference of the food (or food category, or nutrient) violated by the user, is sent to the persuasive explanation component that elaborates these packages and decides which information to use for generating the feedback sent to the user. An example of violation instance represented by using the JSON format is shown in Figure 2.

```
violation: [ userId: fb267
             violationId: violation_fb267
             ruleId: MR-MEDITERRANEAN-028-QB
             meal: MEAL-58ccf3cbfd110f24e59eeced
             history: 1
             expectedQuantity: 200
             quantity: 300
             unit: ml
             level: 1
             timestamp: 1491063927420
             priority: 1
             rule: MR-MEDITERRANEAN-001-GWEEK
             entity: SweetBeveragesAndJuices
             entityType: FOODCATEGORY
             startTime: 1491043927420
             endTime: 1491063927420
             constraint: less
             goal: MEDITERRANEAN-GOAL-D-190
           ]
```

Fig. 2. Example of the violation bean produced by the reasoner in consequence of the violation of a rule that limits the consumption of fruit juice to 200 ml.

4 Explaining Logical Inconsistencies with Natural Language

Here we present a method that performs a linguistic realization of the violation beans of Figure 2 that is useful as motivational message. This realization has to be human understandable and convince users to avoid undesired behaviours that trigger such inconsistencies. Therefore, we need i) a persuasive framework that helps users in conduct a good dietary behaviour (Section 4.1); ii) an effective natural language generator method that translates the logical language of the reasoning results (Section 4.2). Both

components need the HeLiS ontology to retrieve the necessary data. Figure 3 shows the architecture of our method. The core part relies on templates (a grammar) that encode

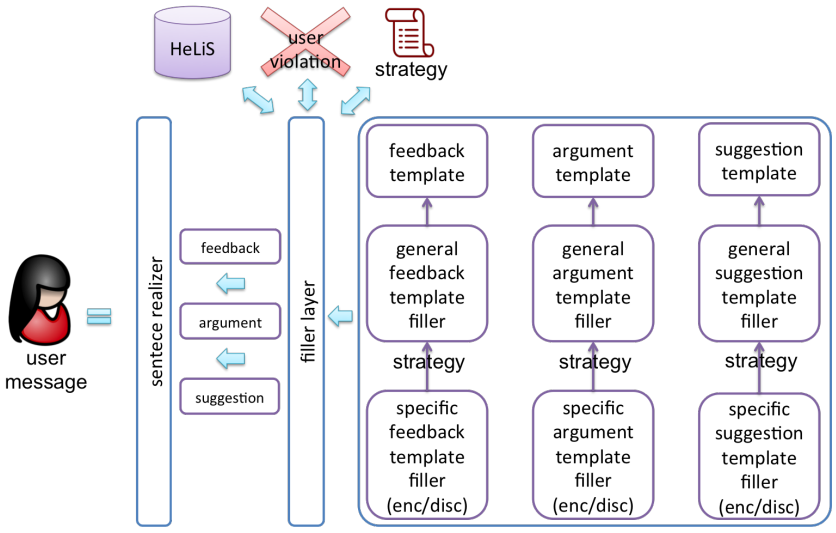


Fig. 3. Architecture our method: the templates are a grammar that translates a logical language into a natural one. They are organized according to persuasion strategies.

the several parts (feedback, arguments and suggestion) of a persuasion message. The terminal symbols of these templates are organized according to a hierarchy where the most specific terms are related to specific persuasion strategies. A filler layer manages the filling of the terminal symbols into the templates. Once the templates are filled, a sentence realizer generates natural language sentences that respect the grammatical rules of a target language (here Italian).

4.1 The Persuasive Framework

We inspired our work from the theoretical framework in [2] for encoding real-time tailored messages in behavior change applications that can be adapted to different generation strategies ranging from canned text to deep generation. The framework is based on four basic properties: *timing*, *intention*, *content* and *representation*. Timing and intention are related to the *persuasion strategy* whereas the others involve the *persuasive content* of the message. We choose this framework as it is a good balance between a "vertical" approach, deeply focused on the domain but with poor generalization properties, and a "horizontal" one that is not bounded to a specific domain but it is limited to be only at a theoretical/conceptual level.

Persuasion Strategy The violation bean of Figure 2 contains all the information explaining the inconsistency of the user's dietary behaviour with respect to the HeLiS

ontology. In addition, at the end of a day/week many of this beans can be generated. However, a long list of these beans is understandable mainly by the domain experts and, most of all, it does not prevent the user to avoid such an erroneous behaviour. A persuasion strategy addresses this challenge by considering the right *timing* for sending the bean, the *choice of the violation* bean to send to the user (not covered in [2]) and the *intention* the system wants to communicate to the user.

The **timing** represents the event prompting the creation of a new message. Message generation can be triggered by specific events (e.g., the generation of a new violation bean) or by temporal events. In particular, our system works with three kinds of events:

- events related to user's habits and behavior (i.e., the generated violations);
- time scheduling: the need to send particular information to the user at specific time of the day or of the week;
- localization: the third event triggering the generation of a message after recognizing that the user is in a specific place (e.g., near a vending machine).

The first kind of events is directly triggered by the detection (through the logical reasoning process of Section 3) of a violation; hence, those information are used for generating the persuasive explanation. The second and third kinds of events, instead, generate persuasive explanations by starting from a pool of past violations.

Once a list of violation beans has been generated, a **choice of the violation** is performed to avoid annoying the user with too many and repetitive messages. If the list of violations is empty, the system infers that the user adopted a healthy behavior so it sends messages with "positive" reinforcing feedback. If such list is not empty, the system sends a message regarding only one violation to provide the user with varied content about different aspects of a correct behavior. The violation is chosen according to (i) its priority, (ii) the number of times it was committed (see the history parameter in Figure 2), and (iii) the number of times the same violation was the object of a message. For example, if a message discouraging to drink sweet beverages has already been sent in the last 4 days, the next highest priority violation bean not sent recently is chosen.

Once a violation bean is selected, a persuasion strategy computes the **intention** (or aim) the persuasive message should convey. According to [2], the intention is composed by a *feedback* on user's activity, an *argument* about the consequences of user's behaviour and a *suggestion* to follow a healthy behaviour. We consider two kinds of intentions: to *encourage* or *discourage* the user to follow a healthy or unhealthy behaviour. In the example of Figure 2, the user drank too much sweet beverages, thus the intention is to discourage this behaviour.

Persuasion Content The **content** of the message is the information the message has to convey to the user. The content generation is the filling of the feedback, argument, suggestion components:

Feedback is the part of the message that informs the user about the unhealthy behavior. Feedback is generated considering data included in the selected violation: the entity of the violation represents the object of the feedback, whereas the level of violation (the deviation between the expected food quantity and the actually one) is used to

represent the severity of the incorrect behavior. Feedback contains also information about timing to report the moment in which violation was committed.

Argument is the part of the message that informs the user about the possible consequences of a behavior. For example, in the case of diet recommendations, the argument consists of two parts: i) information about nutrients contained in the food intake that caused the violation and ii) information about consequences that nutrients have on health. Consequences imply the positive or negative aspects of nutrients according to the encourage or discourage intention, respectively.

Suggestion this part is the solution proposed to the user in order to motivate him/her to change his behavior. This suggestion informs the user about the alternative and healthy behavior that he/she can adopt.

The **representation** regards the format of the content to present to the users. We focus on a natural language representation, however, the persuasive framework deals also with audio or visual formats, for example we can use hGraphs (`http://hgraph.org/`).

4.2 Linguistic Realization of the Persuasive Content

We describe the process of generating the persuasive explanation starting from the received violation bean, the chosen strategy (here encourage or discourage) and HeLiS. As shown in Figure 3, the natural language generation of the content is performed with templates. This is due to the fact that it is very difficult to build a big and tailored dataset of persuasion sentences to perform the linguistic realization with deep learning techniques. In addition, we need the total control on the generated output as wrong indications could lead to serious problems in the healthcare domain. Morevoer, our template system is devised to allow the dynamic construction of tailored sentences thus avoiding standard canned texts. Here, we encode the feedback, argument and suggestion components with some templates, i.e., a grammar with nonterminal/terminal symbols and production rules. The terminal symbols are selected in the filler layer module to fill the nonterminal ones according to the violation, the strategy and HeLiS. Once the templates are filled, they are sent to a sentence realizer that adjusts the raw sentence according to the syntax rules of the selected natural language, here Italian.

The Template System The template system is the organization of the templates according to the presence of nonterminal/terminal symbols and the persuasion strategy. They are organized in layers. The first is the structure of the feedback, argument and suggestion components. It is encoded as a set of production rules between generic nonterminal symbols, Table 1. The second layer consists of production rules between nonterminal and terminal symbols about the domain. This regards the content of the templates, see Table 2. The third layer contains rules between nonterminal and more specific terminal symbols related to the chosen persuasion strategy, Table 3. This decoupling of the templates structure from their content allows the portability of the templates. Indeed, the first layer could be adapted in other domains with other languages with very low effort. Indeed, our target language is Italian but the templates are the same for English and we here just translate the terminal symbols. On the other hand, if a different persuasion strategy needs to be adapted this reflects only the last layer.

1) Structure of the feedback template:

```
feedback := temporal_adv + feed_verb + adj + quantity + food_entity
```

2) Structure of the argument template:

```
argument := intro + food_ent_category + verb_adj + food_property + conseq_verb +
            consequence
```

3) Structure of the suggestion template:

```
suggestion := intro + food_entity + alternative
```

Table 1. First layer of the template system regarding the structure of the templates.

Table 1 shows the structure of the feedback, argument and suggestion components. This is the concatenation (symbol +) of some nonterminal symbols that are filled with the terminal ones of tables 2 and 3. The filling can be direct (see `intro` symbol of Table 2) or dependent from other data such as the violation or HeLiS. This dependency needs to be computed by the filler layer module and it can be just a query to HeLiS or could require more complex operations. For example, the symbols `food_entity` or `food_ent_category` are filled with the corresponding HeLiS labels retrieved by using the field `entity` of Figure 2. Some nonterminal symbols (e.g., the `feed_verb`) can be dependent from the verb and its tense: e.g., beverages imply the use of the verb "to drink" while for solid food we used "to eat". To increase the variety of the message the verbs "to consume" and "to intake" are also used. Simple past tense is used when violation is related to specific moments ("Today you did not eat enough vegetables"), while simple present continuous is used when the violation is related to a period of time not yet ended ("This week you are drinking a lot of fruit juice"). The filling of

1) Terminal symbols for the feedback template:

```
temporal_adv := ["today"|"in the last seven days"]violation
feed_verb := ["to eat"|"to consume"|"to intake"|"to drink"]violation, tense
food_entity := []violation, HeLiS
```

2) Terminal symbols for the argument template:

```
intro := "do you know that"
food_ent_category := []violation, HeLiS
```

Table 2. Second layer of the template system regarding the content of the templates.

other symbols can require more complex operations as long as we are processing the most specific layers of the template system. Indeed, the symbols of Table 3 needs the computation of the strategy. This is given by the field `constraint` in the violation bean: a "less" constraint (fruitjuice $<=$ 200ml) refers to an excess of this food and this behaviour has to be discouraged. A "greater" constraints (vegetables $>=$ 200g) implies an insufficient amount of this food and this behaviour has to be encouraged. Therefore,

a "less" constraint will trigger a discourage strategy, whereas a "greater" constraint will trigger an encourage strategy with the consequent choice of the right terminal symbols in the third template layer. Other template filling could require meta-reasoning strategies

Encourage	Discourage
1) Specific terminal symbols for the feedback template:	
`adj := ["not enough"| "too little"]`$_{\texttt{violation}}$	`adj := ["a lot of"| "too much"]`$_{\texttt{violation}}$
`quantity := ["({} of at least {})"]`$_{\texttt{violation}}$	`quantity := ["({} of maximum {})"]`$_{\texttt{violation}}$
2) Specific terminal symbols for the argument template:	
`verb_adj := ["to be rich of"]`	`verb_adj := ["to contain a lot"]`
`food_property := []`$_{\texttt{HeLiS, violation}}$	`food_property := []`$_{\texttt{HeLiS, violation}}$
`conseq_verb := ["that help to"]`	`conseq_verb := ["that can cause"| "that may contribute to"]`
`consequence := []`	`consequence := []`
3) Specific terminal symbols for the suggestion template:	
`intro := ["next time try to alternate"]`	`intro := ["next time try with"]`
`food_entity := []`$_{\texttt{violation}}$	
`alternative := "with" + []`$_{\texttt{HeLiS}}$	`alternative := []`$_{\texttt{HeLiS}}$

Table 3. Third layer of the template system regarding the strategy/content of the templates.

to identify the appropriate content that can depend on qualitative properties of food, user profile, other specific violations, and the history of messages sent. This can be noticed in the choice of alternative foods for the suggestion template. HeLiS provides foods that are valid alternatives to the consumed food (e.g., similar-taste relation, list of nutrients, consequences on user health). Then, these alternatives are filtered according to the user profile: even if fish is an alternative to legumes it will not be proposed to vegetarians. Moreover, foods that can cause a violation of "less" or "equal" constraints cannot be suggested, e.g., meat cannot be recommended as alternative to cheese if the user has already eaten its maximum quantity. Finally, control on messages history is performed to avoid the repetitiveness of the message content.

The Sentence Realizer Our system creates the message directly in the desired language through the Sentence Realizer (SR). The SR takes in as input the filled templates for the feedback, argument and suggestion components and generates a complex and well-formed sentence according to the grammar rules of the target language, putting spaces, capitol letters and choosing the correct inflected forms of the lemmas. In particular, the Italian language is morphologically richer than English and it entails additional linguistic resources management to harmonize the various parts of the sentences. To this end, the SR implements a morphological engine based on Morph-it!, a morphological resource for the Italian language [19] with a lexicon of inflected forms with their base lemmas and morphological features: gender and number for nouns and articles; gender, number and positive, comparative, superlative for adjectives; tense, person and number for verbs; number, gender, person for pronouns, etc. The Morph-it! version used in the system contains about 35,000 lemmas and 500,000 entries. The SR invokes the morphological engine to compose the basic lemmas and to agree verbs, articles, articulated

propositions and adjectives with the nouns according to the different roles that the noun plays in a sentence (subject, object, possessive form, etc.) according to the Italian grammar rules. Regarding our example of Figure 2, the final persuasive message is: "Today you have drunk too much (300 ml of maximum 200 ml) fruit juice [feedback]. Do you know that sweet beverages contain a lot of sugars that can cause diabetes [argument]? Next time try with a fresh fruit [suggestion]".

5 Use Case: The *Key to Health* Project

Systems for personalized healthy lifestyle recommendations fall in the broad area of decision support. The goal of these systems is to help and guide users in taking healthy-informed decisions about their lifestyle, on aspects such as food consumption. Such systems have to take a decision (e.g., suggesting conscious and healthy food consumption), similarly as a human expert would do, based on available data (e.g., nutrients ingested in the last meals, user health conditions), and to communicate these decisions to the users according to their preferred means and modalities.

As a specific case study, the presented system has been implemented into our HORUS.AI platform and deployed and evaluated in the context of the project *Key to Health* in workplace health promotion (WHP) inside our institution (Fondazione Bruno Kessler, FBK). WHP, defined as *the combined efforts of employers, employees, and society to improve the mental and physical health and well-being of people at work*[3], aims at preventing the onset of chronic diseases related to an incorrect lifestyle through organizational interventions directed to workers. Actions concern the promotion of correct diet, physical activity, and social and individual well-being, as well as the discouragement of bad habits, such as smoking and alcohol consumption. Within the Key to Health project, HORUS.AI has been used by 120 FBK's workers (both researchers and employers) as a tool to persuade and motivate them to follow WHP dietary recommendations. Table 4 shows main demographic information concerning the users involved in the performed evaluation campaign. All users were in good health. Indeed, in this first pilot we decided to not involve people affected by chronic diseases or other diseases.

6 Evaluation

In this Section, we report the evaluation activities we performed within our use case by adopting the HORUS.AI platform. The evaluation we propose is twofold. First, we present the validation performed by the domain experts with respect to the correctness and appropriateness of the generated messages (Section 6.1). This validation aims to verify that the explanations provided by the system are coherent with respect to the detected unhealthy behaviors. Second, we discuss the effectiveness of generated explanations on users' behaviors (Section 6.2) by showing how the use of explanations resulted more helpful with respect to a control group of users received punctual feedback without any detail. The evaluation of reasoning performance is out of scope of this paper. The reader may find these details in [9].

[3] Luxembourg Declaration on workplace health promotion in the European Union, 1997.

Dimension	Property	Value
Gender	Male	57%
	Female	43%
Age	25-35	12%
	36-45	58%
	46-55	30%
Education	Master Degree	42%
	Ph.D. Degree	58%
Occupation	Ph.D. Student	8%
	Administration	28%
	Researcher	64%

Table 4. Distribution of demographic information of the users involved in the evaluation.

6.1 Domain Experts Evaluation

The first validation of our approach concerns the correctness and appropriateness of the explanations generated by the system for supporting the interactions with users. Thus, we present below the procedure for defining and validating: (i) the structure of explanation templates and (ii) the appropriateness of the generated explanations with respect to the detected violations.

Explanation Templates Validation. Three experts [4] have been involved for modeling the templates adopted for generating the explanations. As it has been explained in Section 3, explanations are generated by starting from a finite set of templates that are combined together according to the information contained in the violation packages created by the reasoner. For example, given the category contained in the violation and the violation level, templates concerning the positive or negative properties of the specific food category are connected with verbs and adjectives for shaping the final message. The set of message templates has been validated by the experts that verified the grammatical and content correctness of each template.

Appropriateness of Explanations. The second validation task, where experts were involved, concerned the appropriateness of the messages generated with respect to the violations detected by the reasoner. In order to perform this validation, we performed the following steps:

1. we built data packages representing combinations of meals that should trigger, for each rule contained in the system, the detection of the corresponding violation;
2. we verified that the reasoner correctly detected the violation associated with a given data package;
3. we checked, together with the experts, the appropriateness of the explanation generated with respect to each detected violation.

The analysis of the pairs violation-explanation triggered slight revisions of the linguistic fragments. In particular, some verbs and adjectives used in the fragments were changed to better contextualize the messages.

[4] All experts are dietitians and well-being coaches of our local healthcare department.

6.2 Effectiveness of Explanation

The second evaluation concerned the effectiveness analysis of generated explanations on the user study designed within the *Key to Health* project. The user study consisted in providing to a group of users a mobile application we created based on the services included into the HORUS.AI platform. We analyzed the usage of a mobile application connected with our platform for seven weeks by monitoring the information provided by the users and the associated violations. Our goal was to measure the effectiveness of the explanations generated by our platform by observing the evolution of the number of detected violations. The 120 users involved in the *Key to Health* project have been split in two groups. A first group of 92 users received the whole persuasive messages generated by using the template system. Whereas a second group of 28 users, that was our control group, did not receive any composition of feedback, argument and suggestion, but only canned text messages notifying when a rule was violated. The expectation was to find a higher decrease in the number of violations through the time by the users receiving persuasive messages.

Results concerning the evolution of the violation numbers are presented in Figure 4. We considered three different kinds of dietary rules:

- QB-Rules: these rules define the right amount of a specific food category that should be consumed in a meal.
- DAY-Rules: these rules define the maximum (or minimum) amount (or portion) of a specific food category that can be consumed during a single day.
- WEEK-Rules: these rules define the maximum (or minimum) amount (or portion) of a specific food category that can be consumed during a week.

The three graphs show the average number of violations per user related to the QB-Rules, DAY-Rules, and WEEK-Rules sets respectively. The blue line represents the number of violations, while the red line the average standard deviation observed for each single event. Then, the green line represents the average number of violations generated by the control group and the orange one the associated standard deviation. As mentioned earlier, QB-Rules are verified every time a user stores a meal within the platform; DAY-Rules are verified at the end of the day; while WEEK-Rules are verified at the end of each week. The increasing trend of the gap between the blue and green lines demonstrates the positive impact of the persuasive messages sent to users. We can observe how for the QB-Rules the average number of violations is below 1.0 after the first 7 weeks of the project. This means that some users started to follow all the guidelines about what to consume during a single meal. A positive result has been obtained also for the DAY-Rules and the WEEK-Rules. In particular, for what concerns DAY-Rules the average number of violations per user at the end of the observed period is acceptable by considering that it drops of about 67%. For the WEEK-Rules, however, the drop remained limited. By considering the standard deviation lines, we can appreciate how both lines remain contained within low bounds and after a more in depth analysis of the data, we did not observe the presence of outliers.

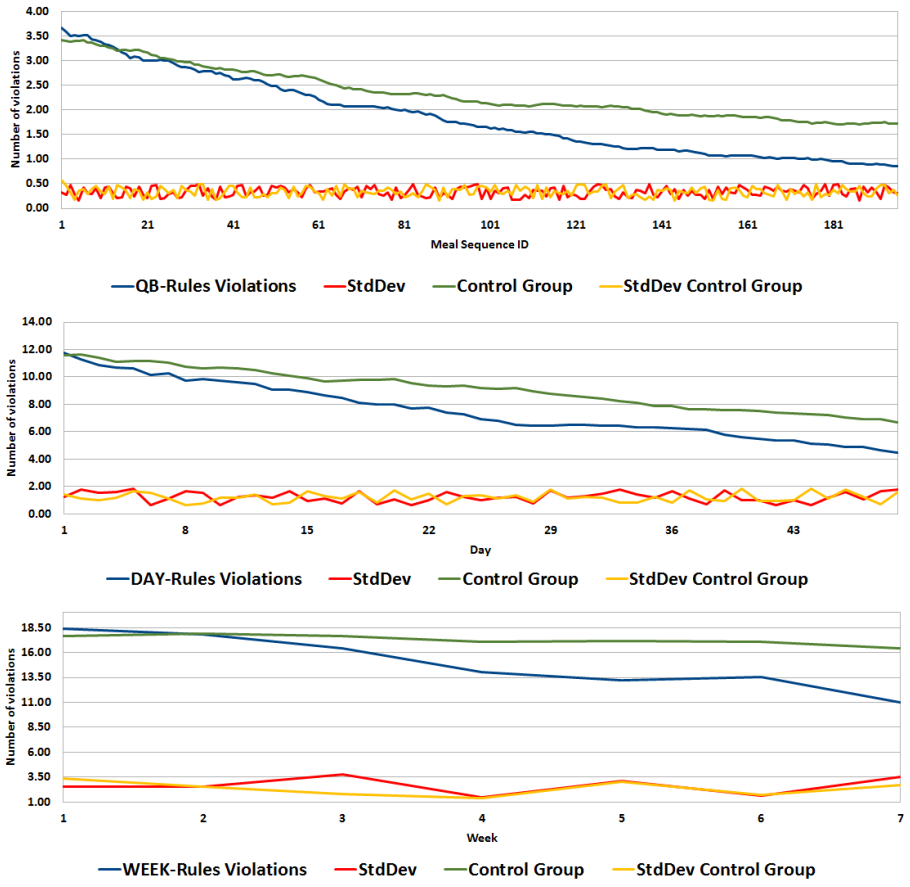


Fig. 4. Variation of the number of detected violations within the *Key To Health* time span.

7 Conclusions

We presented an explainable AI system supporting the users in following an healthy diet. The system checks the presence of unhealthy behaviours based on the food consumed by users. We discussed in particular the role of the natural language generation component and how it exploits information inferred by the reasoner for generating contextual effective explanations. We evaluated our system in a real-world context by discussing the effectiveness of using persuasive explanations with respect to canned texts. Results demonstrated how persuasive explanations allows the user to follow an healthy dietary behaviour. Moreover, the modular template systems allows the dynamic construction of natural language sentences and the templates portability in other domains. As future work, the persuasive explanations of user' behavior will be used in a Computational Persuasion framework [12] to develop a chatbot that understands the user's needs and difficulties to better persuade him/her at following healthy lifestyles.

References

1. Adadi, A., Berrada, M.: Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). IEEE Access **6**, 52138–52160 (2018)
2. op den Akker, H., Cabrita, M., op den Akker, R., Jones, V.M., Hermens, H.: Tailored motivational message generation: A model and practical framework for real-time physical activity coaching. Journal of Biomedical Informatics **55**, 104–115 (2015)
3. Baader, F., Horrocks, I., Sattler, U.: Description logics. In: Handbook of Knowledge Representation, Foundations of Artificial Intelligence, vol. 3, pp. 135–179. Elsevier (2008)
4. Bauer, J., Sattler, U., Parsia, B.: Explaining by example: Model exploration for ontology comprehension. In: Description Logics. CEUR Workshop Proceedings, vol. 477. CEUR-WS.org (2009)
5. Borgida, A., Franconi, E., Horrocks, I.: Explaining ALC subsumption. In: Horn, W. (ed.) ECAI 2000, Proceedings of the 14th European Conference on Artificial Intelligence, Berlin, Germany, August 20-25, 2000. pp. 209–213. IOS Press (2000)
6. Corcoglioniti, F., Rospocher, M., Mostarda, M., Amadori, M.: Processing billions of RDF triples on a single machine using streaming and sorting. In: ACM SAC. pp. 368–375 (2015)
7. Dragoni, M., Bailoni, T., Maimone, R., Eccher, C.: Helis: An ontology for supporting healthy lifestyles. In: International Semantic Web Conference (2). Lecture Notes in Computer Science, vol. 11137, pp. 53–69. Springer (2018)
8. Dragoni, M., Bailoni, T., Maimone, R., Marchesoni, M., Eccher, C.: HORUS.AI - A knowledge-based solution supporting health persuasive self-monitoring. In: International Semantic Web Conference (P&D/Industry/BlueSky). CEUR Workshop Proceedings, vol. 2180. CEUR-WS.org (2018)
9. Dragoni, M., Rospocher, M., Bailoni, T., Maimone, R., Eccher, C.: Semantic technologies for healthy lifestyle monitoring. In: International Semantic Web Conference (2). Lecture Notes in Computer Science, vol. 11137, pp. 307–324. Springer (2018)
10. Gatt, A., Krahmer, E.: Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. J. Artif. Intell. Res. **61**, 65–170 (2018)
11. Halstead, M.H.: Elements of Software Science (Operating and Programming Systems Series). Elsevier Science Inc., New York, NY, USA (1977)
12. Hunter, A.: Towards a framework for computational persuasion with applications in behaviour change. Argument & Computation **9**(1), 15–40 (2018)
13. Kalyanpur, A., Parsia, B., Sirin, E., Hendler, J.A.: Debugging unsatisfiable classes in OWL ontologies. J. Web Semant. **3**(4), 268–293 (2005)
14. Kontopoulos, E., Bassiliades, N., Antoniou, G.: Visualizing semantic web proofs of defeasible logic in the DR-DEVICE system. Knowl.-Based Syst. **24**(3), 406–419 (2011)
15. Lam, J.S.C.: Methods for resolving inconsistencies in ontologies. Ph.D. thesis, University of Aberdeen, UK (2007)
16. McCabe, T.J.: A complexity measure. IEEE Trans. Software Eng. **2**(4), 308–320 (1976)
17. McGuinness, D.L., Borgida, A.: Explaining subsumption in description logics. In: IJCAI (1). pp. 816–821. Morgan Kaufmann (1995)
18. Roussey, C., Corcho, Ó., Blázquez, L.M.V.: A catalogue of OWL ontology antipatterns. In: K-CAP. pp. 205–206. ACM (2009)
19. Zanchetta, E., Baroni, M.: Morph-it! a free corpus-based morphological resource for the italian language. In: IN: PROCEEDINGS OF CORPUS LINGUISTICS, HTTP://DEV.SSLMIT.UNIBO.IT/LINGUISTICS/MORPH-IT.PHP (2005)

Towards Explaining Natural Language Arguments with Background Knowledge

Ioana Hulpuş[1], Jonathan Kobbe[1], Maria Becker[2], Juri Opitz[2], Graeme Hirst[3], Christian Meilicke[1], Vivi Nastase[2], Heiner Stuckenschmidt[1], and Anette Frank[2]

[1] Data and Web Science Group, University of Mannheim, Germany
{firstname}@informatik.uni-mannheim.de
[2] Institute for Computational Linguistics, Heidelberg University, Germany
{mbecker, opitz, nastase, frank}@cl.uni-heidelberg.de
[3] University of Toronto, Toronto, Canada
gh@cs.toronto.edu

Abstract. In this paper, we propose the task of argument explicitation, a task that makes the structure of a natural language argument explicit, as well as the background knowledge the argument is built on, in the form of implicit premises or contextual knowledge. The purpose of argument explicitation is to support the understanding of an argument by providing users with an end-to-end analysis that offers a critical assessment of arguments including identification of argument weaknesses. Besides, the results of the argument explicitation process can be used by machines to retrieve similar arguments as well as counter-arguments. We propose a framework for argument explicitation that joins a variety of AI and NLP-based argumentation mining sub-tasks that by now have mostly been treated separately in the literature. We identify the challenges this task entails, while at the same time highlighting the opportunities brought by the recent development of structured, external knowledge sources.

1 Introduction

The analysis and use of Argumentation in natural language texts is an active field of research in Artificial Intelligence. Common lines of work include the identification of argumentation units [32, 44, 50, 52] and relations [11, 36, 40, 50], the measurement of argument quality [24,57] and the synthesis of argumentative texts [56]. While many tasks in natural language processing (NLP) can be solved with surprising accuracy using only surface features, tasks relating to argumentation often require a deeper understanding of the reasoning behind a line of argumentation.

In this paper, we discuss the *problem of providing explanations for arguments*, giving an account of the opportunities and challenges this involves. We define the task of *explicitation of arguments* whose purpose is to support the understanding

of a given argument by providing either end users or a computational system that tries to interpret an argument, with a structured and semantically founded analysis of the argument and to enrich it, if necessary, with explanations of otherwise implicit information that is crucial for the logics and understanding of the argument. This task brings together multiple research directions, some of which have already been investigated in the literature – however mostly in theoretical, as opposed to computational approaches. Indeed, we emphasize that while many of the challenges have been long debated in philosophy and logics communities, there are very few accounts of them in the NLP and modern AI communities, where these questions are now only starting to be addressed.

Argument explicitation is important in order to support end-users to critically judge natural language arguments. The need for systems that are able to perform argument explicitation has become particularly critical in the light of the current wave of references to "fake news". Explicitation of how the stated premises support or attack a given conclusion, as well as the provision of a full-fledged argument structure can shed light on both *validity* (does the conclusion follow logically from the premises?) and *soundness* (are the premises true?) of arguments. Beyond a purely logical account of argumentation, as one end of the extreme, or recourse to fact checking to corroborate the truth of premises on the other, argument explicitation aims at making explicit any background knowledge relevant for the understanding of the argument, either in the form of implicit premises, or relevant facts, states of affairs, or relations that connect explicitly stated argument components in a meaningful way.

In this paper, we discuss notions of explanations known in other contexts and motivate a new kind of explanation that is targeted to the explicitation of natural language arguments that makes the knowledge and the mechanisms of an argument explicit (Section II). We will distinguish different facets of argument explicitation and what specific kinds of knowledge are required for them (Section III). In Section IV, we discuss different types of argument explicitation and what kinds of explanations we can expect from them, in view of a content-based assessment of the validity, strength and quality of an argument. Section V summarizes our findings and concludes with an outlook on promising first steps towards a computational account of argument explicitation.

2 Explaining Arguments

2.1 Explaining Arguments with Deductive Reasoning

Researchers in the field of Logics consider arguments as logical formulas: the truth of the conclusion is a logical consequence of the truth of the premises. In this setting, the logical proof that establishes the entailment or inconsistency serves as an explanation of the respective relation. Consider the following example inspired from Walton and Reed (2005) [59]:

Example 1. Drastic decline in natural life is cause for alarm. Toads and frogs are forms of natural life and there is a drastic decline in toads and frogs. Hence, there is a cause for alarm.

Premise 1 $\forall x, natural\_life(x) \wedge drastic\_decline(x) \Rightarrow alarm(x)$
Premise 2 $natural\_life(toads\_and\_frogs)$
Premise 3 $drastic\_decline(toads\_and\_frogs)$
Conclusion $alarm(toads\_and\_frogs)$

Fig. 1. Example of formal logics-based explicitation of the argument in Example 1.

The example shows a syllogistic argument whose formalization is available in Figure 1. Given the formalization, an automated reasoner such as a Prolog reasoner can validate the argument. However, looking at this argument from the perspective of an everyday argument, it is straightforward to recognize several problems that reach beyond its deductive validity.

First, the text of the exemplified argument is rather unnatural, as the statement *toads and frogs are forms of natural life* is very unlikely mentioned in an everyday argument but it is most often implied. However, without it, the argument becomes deductively invalid, since it would miss *Premise 2* in Fig. 1. Most everyday arguments would face this problem. Arguments with unstated premises are called enthymemes [60] and we get back to them in the following sections.

Second, the argument's soundness is not beyond doubt. While the second premise would appear to be true to the majority of people, the truth of the first and third premises pertains to a higher level of subjectivity (when is decline *drastic*?). Indeed, in informal reasoning, counter-arguments question the validity of arguments as well as their soundness.

Thus, everyday arguments cannot be modeled in a deductive framework [59]. These arguments, whose conclusion can be defeated by either defeating the premises, or by adding new premises, are called *defeasible arguments*. In the following, we focus particularly on types of explicitations suitable for them.

2.2 Explaining Arguments with Informal Reasoning

In the informal reasoning literature, we identify several types of explanations each fulfilling a particular role, in different contexts:

Explanation as a discursive act has the function of providing reasons in support of an accepted conclusion [9,34,38]. In this regard, an explanation differs from an argument, as the explanation does not aim to prove the validity of the conclusion (which is the role of an argument), but rather considers the conclusion as being valid, and tries to provide the reasons for the occurrence of the event or state of affairs expressed by the conclusion.

Explanation as hypothesis occurs particularly in the context of abduction - the method of creating new hypotheses to explain unexpected observations, e.g. in the context of scientific literature [28, 53, 55].

Explanation for transparency is applied to enrich automatic systems with an output functionality that aims to inform the end-user with all the knowledge and processes used by the system for producing its primary output. This is the most common type of explanation in artificial intelligence [2, 43, 46].

In this paper, we discuss a new type of explanations, called *argument explicitation*: the explanation of an argument with the specific purpose of making the knowledge and mechanism of the argument explicit. The recent advances in natural language understanding and the availability of structured knowledge bases bear many opportunities to tackle some of the hard problems that this task entails.

3 Argument Explicitation

Broadly, the task of argument explicitation that we address consists of two subtasks. The first task – *argument analysis* – is concerned with analyzing the text in order to identify the *argument components* (e.g., premises and conclusion) and the overall structure of the argument. The second task – *argument reconstruction* – is concerned with making explicit any unstated, but implicit premises, as well as implicit connections between concepts mentioned in argument components, in terms of background knowledge. Most of the AI and particularly computational linguistics research in argumentation focuses on the first subtask [35], [39], [18], [32, 50], [3]. The second sub-task has by now been mainly addressed from a theoretical, or philosophical perspective by Walton and Reed (2005) [59], who reconstruct enthymemes (arguments with unstated premises) with argumentation schemes.

In the area of the argument analysis task, three very recent contributions outline the need for understanding argumentation on a deeper level. One investigation [37] shows that predictions of a state-of-the-art argumentative relation classification system are mostly driven by contextual shallow discourse features, while the model pays only little attention to the actual content of an argument. The need for deeper understanding of the content of the argumentative text has also been acknowledged with respect to the argumentative reading comprehension task (ARC)[4] [8]. The approach of Kobbe et al. (2019) [27], takes a step in this direction, but their knowledge-augmented model only marginally outperforms the linguistic baseline. Deeper understanding of arguments is even more crucial for the task of argument reconstruction, and as long as argument analysis is only achieved at a shallow level, there is very little hope for successful argument reconstruction on top of it. In light of these observations, we point out the kind of knowledge that such a system must access, model and integrate.

Knowledge about natural language is by far the most exploited type of knowledge in the literature with respect to argument mining. However, such knowledge has many facets, but it is by now only captured by relatively shallow features, such as discourse markers that indicate argument components (see e.g. [40]), or implicitly captured through training feature-based classifiers and recently, neural models (cf. [33, 49, 51]).

Knowledge about argumentation has been extensively researched, mostly in the philosophical literature. Here, multiple ways of modelling arguments have been proposed, including patterns of defeasible reasoning [14].

[4] SemEval-2018: https://competitions.codalab.org/competitions/17327

Background knowledge has probably been the most neglected type of knowledge in the current state of the art of argument analysis. Early argument comprehension systems [1, 5] made heavy use of hard coded, very precise domain knowledge. At the same time, in philosophy we encounter Schank's scripts [45] as the most referenced representation of domain knowledge for both argument comprehension and reconstruction [60]. Nonetheless, apart from very recent work of Botschen et al. [8] and Kobbe et al. [27], little progress has been made in using domain knowledge for argument comprehension and reconstruction. Recent work investigated the reconstruction of implicit knowledge in argumentative texts by way of manual annotation [4,7], but computational reconstruction approaches are still out of sight.

We claim that automated argument explicitation must model and reason with all of these complementary types of knowledge. In the following, we detail some of the sub-tasks of argument explicitation, focusing particularly on the challenges that can be addressed by, or that require exploiting background knowledge. We think that advances in the availability of large-scale knowledge bases bring significant opportunities in this direction.

3.1 Model-based Explicitation

In order to understand how and why defeasible arguments work, multiple argument models have been proposed. Generally these models aim to classify argumentative units on a more granular level than the generic *premise/conclusion* classification. In the following, we describe two of the most popular such models, and illustrate how we envisage argument explicitation based on them. However, we do not exclude the explicitation based on other models, such as the seven-step argument analysis approach of Scriven [47].

Toulmin Model-based Explicitation In research on argument analysis, one of the most well-known models for arguments is the Toulmin model [54]. It was defined particularly for legal arguments, but has since proven its suitability for a wide range of arguments [26]. This model defines five types of argument components, whose identification facilitates argument understanding.

claim is the statement that the argument intends to prove, and is analogous to the conclusion in other argumentation models;
datum is a statement of a fact, or evidence that supports the claim;
warrant is a statement that provides the connection between claim and datum, facilitating the datum to support the claim;
backing is a statement that justifies why the warrant holds;
qualifier is a statement that indicates the strength of the warrant;
rebuttal is a statement of an exceptional case whose occurrence would remove the authority of the warrant.

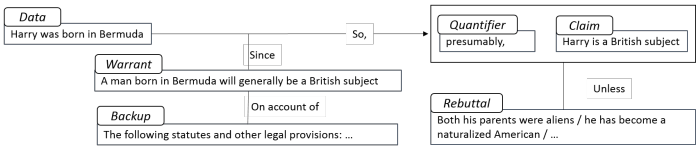


Fig. 2. Example of an argument structure following Toulmin's model

Figure 2 shows a classical example [54] of an argument modelled with the Toulmin scheme. An important challenge for explicitating natural language arguments with the Toulmin model is that most often, not all the components are present in the text. Consequently, a legitimate goal of argumentation explicitation can be to (i) *signal the lack* of specific argument components to the end user, to support her judgment of the validity of the argument, or (ii) to *identify and provide* such missing argument components from Toulmin's scheme, such as Data, Warrants or Backups, to complete the full understanding of the argument. We will come back to this discussion in Section 3.2.

As it can be seen in the example of Figure 2, and as discussed in detail by Freeman [19], warrants often take the form of generalization rules, that are often not explicitly stated. For instance, the argument in Figure 2 would most likely be encountered in everyday argumentation as *Harry is presumably a British subject because he was born in Bermuda*. The availability of background knowledge such as encyclopedic knowledge (i.e., DBpedia) can be exploited in order to suggest such potential warrants. For example, even if omitted from the text, the warrant that *A man born in Bermuda will generally be a British subject* can be reconstructed by noticing (for instance, in DBpedia) that a big fraction of people born in Bermuda are British citizens. The bigger challenge is how to deal with commonsense knowledge, or more specifically, what Feeman (2008) [19] names *empirical warrants* which *ordinarily* hold, for example *Given that x has mailed the letter, one may take it that x wants the addressee to receive it* or *If X ignited a fuse connected to a bomb, X intended to explode the bomb.*

Walton Schemes-based Explicitation Walton proposed about 50 argumentation schemes [59] organized in a hierarchy. These schemes represent common patterns of everyday reasoning, and Fig. 3 shows two of them.

There have already been a number of attempts to classify natural language arguments into Walton argumentation schemes, as well as their components, some of which are purely theoretical [10, 58] while others implement feature-based supervised classification models [17, 29, 30]. An example of an argument from verbal classification, originally published by Lawrence and Reed (2016) [30] is shown in Fig. 4. In this example, the argument text is annotated with the two premises and the conclusion. We use this example to pinpoint two important challenges, besides the actual classification of the arguments based on their Walton scheme.

Argument from Analogy
Premise 1: Generally, case C1 is similar to case C2.
Premise 2: A is true (or false) in case C1.
Conclusion: A is true (or false) in case C2.

Argument from Verbal Classification
Premise 1: a has property P
Premise 2: for all x, if x has property P, x can be classified as having property G .
Conclusion: a has property G.

Fig. 3. Example of two Walton schemes: *Argument from Analogy* and *Argument from Verbal Classification*

First, the logical conclusion following from the two premises, is *The PowerShot SX510 has great image stabilization.* For the conclusion in Fig. 4 to be logically entailed, we must assume the further premise *Cameras with great image stabilization are fantastic.*, which is implied by the text, but is not stated.

Example. *The PowerShot SX510 is a fantastic camera. It is made by Canon and all Canon cameras have great image stabilisation.*

Premise 1: It is made by Canon
Premise 2: all Canon cameras have great image stabilisation
Conclusion: The PowerShot SX510 is a fantastic camera.

Fig. 4. Example of an instance of Argument from Verbal Classification.

Second, in real life, the above argument would likely omit ***Premise 1:*** *It is made by Canon*, and the text would sound closer to *The PowerShot SX510 is a fantastic camera as Canon cameras have great image stabilization.* This adds another level of complexity and challenge to correctly classify the argument as an Argument from Verbal Classification.

A thorough explicitation of this argument that addresses both challenges is illustrated in Fig. 5. As illustrated, the argumentative text that contains only two explicit statements (Premise 2 and the Conclusion), actually implies a chain of two arguments, where the conclusion of the first serves as a premise to the second. In order to obtain such explicitations automatically, it is not sufficient to classify arguments into their corresponding Walton scheme. In addition, the classification of the components (premises and conclusions) is required, and even more challenging, the classification of the schema variables. Given the Argument from Verbal Classification scheme in Fig. 3, the classification of variables for the text *The PowerShot XS510 is a fantastic camera as all Canon cameras have great image stabilization* would be: {*The PowerShot XS510*: **a**, *fantastic camera*: **G**, *Canon cameras*: **P**, *great image stabilization*: **G**}. This classification, would then clarify which are the bits of knowledge that are needed for reconstructing the argument in such a way that it follows the Argument from Verbal

Example. *The PowerShot SX510 is a fantastic camera as all Canon cameras have great image stabilisation.*

First Argument
Unstated Premise 1: *The PowerShot SX510 is a Canon camera*
Premise 2: all Canon cameras have great image stabilisation
Unstated Conclusion: *The PowerShot SX510 has great image stabilisation.*
Second Argument
Unstated Premise 1: *The PowerShot SX510 is a camera and has great image stabilisation.*
Unstated Premise 2: *Cameras with great image stabilisation are fantastic.*
Conclusion: The PowerShot SX510 is a fantastic camera.

Fig. 5. Example of explicitation that includes analysis as well as reconstruction of an instance of Argument from Verbal Classification. The reconstruction makes explicit two arguments following the same scheme of Argument from Verbal Classification. One premise of the second argument is the conjunction of a premise and the conclusion of the first argument. The unstated components are written in Italics.

Classification scheme. Specifically, that **a** (*The PowerShot XS510*) must have property **P** (*Canon cameras*), resulting into Unstated Premise 1. We highlight here the opportunity for using structured knowledge bases that are available on the Web of Data to fill in such generalizing premises. Next, having two distinct strings serving the same role of **G** (*great image stabilization* and *fantastic camera*) can indicate that the author of the argument implies that there is a logical entailment between the two strings, leading to Unstated Premise 2. In the following, we discuss explicitations whose role is to fill in unstated premises.

3.2 Explicitation based on Enthymeme Reconstruction

Arguments with omitted premises are called *enthymemes*. They have been debated in philosophical literature since Aristotle [16,21,22,25,31,59,60]. Regarding our task of argument explicitation, dealing with enthymemes is one of the core challenges. Although explicitation based on Toulmin's model or Walton schemes may be regarded as a tangible aim as long as the problem of implied premises is ignored, we argue that most (informal) natural language arguments are enthymemes, and their explicitation, which includes reconstruction, should not be neglected. In Section 3.1, we provided some hints on how Walton schemes might be used to explicitate enthymemes, while in Section 3.1 we discussed Freeman's (2008) [19] claim that when modelling arguments with the Toulmin model, it is very common that the warrant is implied and omitted. We therefore consider explicitation based on enthymeme reconstruction as a form of explicitation that complements and deepens other types of explicitation proposed above.

The problem of enthymeme reconstruction is arguably an AI complete problem. Broadly, a system tackling enthymeme reconstruction – called an enthymeme machine [59] – must be able to answer three questions: (i) is the analyzed argument an enthymeme? (ii) which are the gaps that need to be filled? (iii) which are

the missing premises? Approaches for addressing questions (i) and (ii) depend on the chosen argument model (e.g., Walton scheme or Toulmin model). Addressing question (iii) is more challenging and actually brings us to the question of the actual purpose or use cases of the task. If the purpose of enthymeme reconstruction is to support the user in judging arguments, we can relax the requirement of stating *the missing premise*. We may instead just ask the system to present a *possible* premise. For instance, reconsidering the example in Fig. 5, instead of generating Unstated Premise 1 *The PowerShot SX510 is a Canon camera* and Unstated Premise 2 *Cameras with great image stabilization are fantastic*, the system would draw the attention of the user to consider some highlighted piece of inserted information that *could* form a coherent argument, e.g., (i) **The Powershot SX510** has the property **Canon camera** and (ii) **great image stabilization** implies **fantastic camera**. This way, it is the user's responsibility to validate the argument, while the system guides this process.

If, however, the purpose of the system is to provide a *true and valid* missing premise, the system must be able to check whether these premises state true facts, e.g., they may be validated against a knowledge base, or they can be flagged as subjective statements. In our example from Fig. 5, the system would search for relations holding between *The Powershot SX510* and *Canon cameras* in a knowledge base, and judge whether the found relation is similar to the relation required by the argument scheme: ***The Powershot SX510*** *has the property* ***Canon camera***. Validating the second unstated premise in our example, by contrast, should be impossible, since it is a subjective statement, not a fact. In such a case, the system might reconstruct a possible premise (***great image stabilization*** *implies* ***fantastic camera***), and flag it as subjective.

We conclude that the system must be able to distinguish between missing premises that are *subjective* as opposed to those that are *facts*. While subjective ones can be flagged as such, using state of the art opinion detection tools, reconstructing facts involves fact checking. This can only be achieved with respect to real-world knowledge available to the system. Such real-world knowledge can be: (i) encyclopedic (e.g., *The Powershot SX510 is made by Canon*) which is available online through Wikipedia and related structured knowledge bases such as DBpedia, Wikidata, Yago; (ii) ontological (e.g., *frogs and toads are animal life*) which is available for instance through taxonomies and lexicons such as WordNet, as well as Wikipedia-based knowledge bases; (iii) common sense knowledge (e.g., *dogs usually bark when strangers enter their space*), which is much harder to source and (iv) contextual, such as the purpose of the document, the author, the time, etc. While the first two types of real-world knowledge can be accessed with state-of-the-art entity linking tools, the last two types of knowledge are more challenging, and in general much less researched. Regarding commonsense knowledge, the recent study of Becker et. al (2016) [4] finds that a large majority of commonsense relations captured by implicit unstated statements in arguments can be mapped to ConceptNet [48] relations.

With respect to contextual knowledge, Green (2010) [23] provides evidence that knowledge needed for explicitating enthymemes can often be found in the surrounding context, meta-data about authors and the targeted audience, etc.

3.3 Acceptability-based Explicitation

The previously proposed types of argument explicitation focus solely on the internal structure of the argument. However, everyday arguments rarely occur in isolation or remain unchallenged. A defining property of everyday arguments is precisely their defeasible nature, i.e., their vulnerability to being attacked by other arguments. The ability of arguments to resist such counterarguments has been named *acceptability* [15].

Acceptability-based explicitation aims to expose the relations holding between the targeted argument and other arguments, weaving a macro structure of argumentation. This type of argumentation analysis, whose target are the relations between arguments, has been researched within the context of abstract argumentation frameworks. One of the first and best studied abstract argumentation frameworks was introduced by Dung (1995) [15]. It defines only one type of relation between arguments, that of *attack* or *defeat*. Dung [15] defines a set of arguments as *acceptable* (by a rational agent), if it can defend itself against all attacks on it. More recent lines of work on argumentation frameworks extend Dung's framework by defining two types of relations between arguments, *attack* and *support* [12, 13]. Drawing inspiration from these frameworks, much of the recent computational linguistic analysis of arguments has focused on automated support/attack relation classification between pairs of arguments [6, 11, 20].

Much of the research on argument analysis considers attack and support relations to exist within a single argumentative text [36, 40, 41, 50, 51]. This is often the case in everyday argumentation, in a rhetorical technique for displaying the argument's ability to defend itself against predictable counter arguments. In order to disentangle the argumentative text in such a way as to explicitate the acceptability of its arguments, one challenge is to identify and extract the *atomic arguments*: (i) *the main argument* - the one whose conclusion is the main conclusion of the text, (ii) *the supporting arguments* - sub-arguments whose conclusions act as premises to the main argument and (iii) (anticipated) *counterarguments* - arguments that attack the main argument. Our intuition is that counterarguments are indicated by what seems like attack relations between premises of the same argument.

Fig. 6 illustrates an explicitation of an argumentative text adapted from the Microtexts of Peldszus and Stede(2015) [40], by isolating two atomic arguments – the main argument and the anticipated counterargument. As shown in Fig. 6, a counterargument can be anticipated and defeated, hence increasing the acceptability of the main argument. In our example, the premise of the main argument attacks the ability of the counterargument's premise to entail the implicit conclusion (since reported relief of complaints is not a scientific proof).

We envisage two levels of acceptability-based explicitation: (i) a shallow explicitation in which an attack or support relation is indicated between pairs

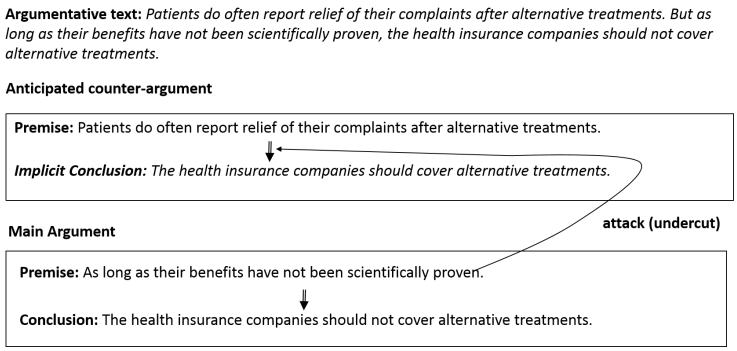


Fig. 6. Example of argumentative text containing attacking statements that are shown to belong to two different arguments.

of arguments and (ii) a deep explicitation in which the particular components (statements) participating in the relation are highlighted. Pollock [42] identifies two common types of attack relations: *rebuttals*, which directly attack the conclusion of an argument, and *undercuts*, which attack the logical entailment of the conclusion given the premise. From this perspective, in Fig. 6, the attack relation between the premise of the main argument and the anticipated counter-argument is an undercut. Acceptability-based explicitation is complementary to the previously defined types of explicitation: the identified individual arguments can be further explicitated with other types of explicitation.

3.4 Knowledge Enhancement-based Explicitation

The last type of explanation that we propose is knowledge enhancement-based explicitation, which provides additional background information about the entities and concepts mentioned in the argument's text, as well as the relations between them. The idea is to activate knowledge which is needed to understand the content of the argument components and how they are linked semantically. Consider the following argumentative text example: *Acetylsalicylic acid helps in case of a myocardial infarct as it reduces the platelet adhesion.*

A potential explicitation of this example for the lay person would be to add background knowledge in the form of additional statements such as *Acetylsalicylic acid is the active ingredient in Aspirin.*, or *Myocardial infarct is another term for Heart Attack.*, or *Reducing the platelet adhesion prevents blood clotting.* A medical doctor would most likely not benefit from this type of explicitation. Instead, they may be interested to know why the prior doctor has preferred *Acetylsalicylic acid* over alternative treatments, etc. Therefore, the challenge for this type of explicitation is to determine what information should be added. This type of explicitation therefore lends itself most naturally to personalization.

Knowledge enhancement-based explicitation bares some similarities to enthymeme reconstruction, but differs from it in that the provided knowledge

statements do not need to be premises. Thus, this type of explicitation does not require any argumentation knowledge. Nonetheless, we expect the extracted knowledge to oftentimes contain the premises required for enthymeme reconstruction and hence provide satisfactory explanations for the end-user. Still, we want to underline the less constrained nature of the knowledge presented in knowledge enhancement-based explicitation, and that while this step might help the user make sense of the argument, it does not reveal how the reasoning behind the argument works.

4 A Framework for Argument Explicitation

In this section, we propose a framework for argument explicitation that considers the presented explicitation facets, as well as how they relate to each other. The framework is illustrated in Figure 7. Given an argumentative text, the first steps towards its explicitation are (i) to enhance it with background knowledge (step K), by retrieving entities and relations that are relevant to the argument from external knowledge bases, and (ii) the identification of the atomic arguments and counterarguments (step A). The extracted background knowledge can assist the acceptability-based explicitation of the argument. For instance, recent work in Kobbe et al. [27] uses DBpedia and ConceptNet in order to classify support/attack relations between argumentative statements.

Once the atomic arguments are identified, the argument explicitation system can proceed to explicitate the argument based on the model(s) of choice. The first and minimal step in this direction is to detect the argumentative units and classify them as premise or conclusion. A more elaborate explicitation is to identify the Toulmin model elements in each argument, as well as their Walton scheme. These two tasks can support each other since in some Walton schemes, the premises can be mapped to either data or warrant elements in the Toulmin model. Furthermore, as discussed earlier, the relevant background knowledge can provide valuable insights for the classification of Walton schemes or Toulmin model elements. Lastly, after each identified argument has been explicitated based on the chosen model(s), the explicitation machine can proceed with enthymeme reconstruction (step E). This step brings further detail into the model-based explicitations by filling in the blank slots of the identified models, and can further explicitate the acceptability of the main argument.

5 Discussion and Implications

In this paper, we introduce the notion of *argument explicitation* as an overarching task that makes the reasoning mechanisms required for understanding natural language arguments explicit to the end-user. The perspective we take in this work is to analyze the very diverse research directions in argumentation from the same viewpoint: that of *explaining arguments*, and to integrate these different research contributions in a common Framework of Argumentation Explicitation.

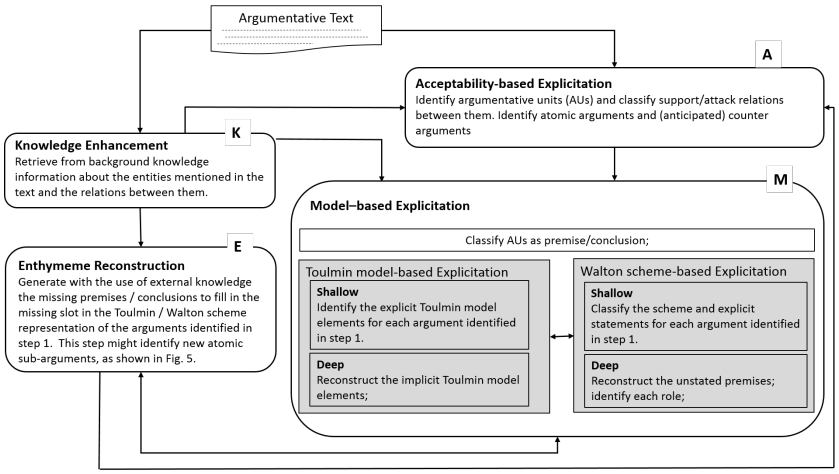


Fig. 7. Proposed Framework for Argument Explicitation

In doing so, we are able to identify the research challenges and opportunities that lie ahead. We are summarizing the most important implications of our analysis:

(i.) For uncovering the reasoning behind arguments, it is of great importance to apply and improve formal argument structure analysis, following *detailed, content-driven argument schemes* such as Toulmin's [54] or Walton's [59] schemes.

(ii.) Throughout the paper we stress and exemplify the importance of extending argument analysis with *enthymeme reconstruction*, by completing arguments with implicit argument components. This requires access to different types of knowledge that may support and validate a given argument in terms of linguistic, encyclopedic or commonsense knowledge. Clearly, this is a challenging aim. Strong NLP and AI capabilities are required in order to fully assess the explicit meaning of a given argument. Strong *reasoning capacities* are needed to be able to *select appropriate knowledge* and to verify the enriched argument to ensure its *validity and soundness* – or else to uncover inconsistencies that are revealed by assuming further information.

(iii.) Besides *appropriate repositories of background or domain knowledge*, alternative ways of identifying relevant knowledge need to be considered, such as *link prediction methods* and *on-the-fly knowledge retrieval* from textual sources, to make implicit assumptions in the NL argument (structure)s explicit.

(iv.) To support this process, *machine reasoning techniques* should be used to enforce high-level constraints over argumentation models, as well as for detecting inconsistencies in content or argument structures.

(v.) Real life arguments are rarely isolated, as they are most often part of debates. In this context, arguments should be treated as belonging to sets of arguments (following Dung). The retrieval of an *assembly of further supporting or defeating arguments* from additional sources should be considered, to facilitate

the judgement of the validity or generality of an argument from a more global perspective.

(vi.) Since the reconstruction of argument components can be highly subjective, the *explicitation of reconstructed knowledge* can be realized e.g. by way of *natural language generation techniques*, to allow end users identify what additional assumptions have been made to support the conclusion. This is especially relevant for argumentation machines, but may also serve humans to fully understand the logics and possible background assumptions of an argument.

While most of the above considerations have been discussed in the theoretical literature, they constitute true challenges to computational treatments of argumentation and need to be addressed in a step-wise fashion.

Acknowledgments

This work has been funded by the Deutsche Forschungsgemeinschaft (DFG) within the project ExpLAIN, Grant Number STU 266/14-1 and FR 1707/-4-1, as part of the Priority Program "Robust Argumentation Machines (RATIO)" (SPP-1999) as well as by the Leibniz Science Campus Empirical Linguistics & Computational Language Modeling, supported by Leibniz Association grant no. SAS2015-IDS-LWC and by the Ministry of Science, Research, and Art of Baden-Wurttemberg.

References

1. Alvarado, S., Dyer, M.G., Flowers, M.: Editorial comprehension in oped through argument units. UCLA Computer Science Department (1986)
2. Amgoud, L., Prade, H.: Using arguments for making and explaining decisions. Artificial Intelligence **173**(3-4), 413–436 (2009)
3. Becker, M., Palmer, A., Frank, A.: Clause types and argumentative texts. In: ArgMining Workshop (2016)
4. Becker, M., Staniek, M., Nastase, V., Frank, A.: Enriching Argumentative Texts with Implicit Knowledge. In: NLDB, LNCS. pp. 21–30 (2016)
5. Birnbaum, L., Flowers, M., McGuire, R.: Towards an ai model of argumentation. In: AAAI. pp. 313–315. AAAI Press (1980)
6. Boltuzic, F., Snajder, J.: Back up your stance: Recognizing arguments in online discussions. In: ArgMining@ACL. pp. 49–58 (2014)
7. Boltuzic, F., Šnajder, J.: Fill the gap! analyzing implicit premises between claims from online debates. In: ArgMining Workshop. pp. 124–133 (2016)
8. Botschen, T., Sorokin, D., Gurevych, I.: Frame-and entity-based knowledge for common-sense argumentative reasoning. In: ArgMining Workshop. pp. 90–96 (2018)
9. Brem, S.K., Rips, L.J.: Explanation and evidence in informal argument. Cognitive science **24**(4), 573–604 (2000)
10. Cabrio, E., Tonelli, S., Villata, S.: From discourse analysis to argumentation schemes and back: Relations and differences. In: International Workshop on Computational Logic in Multi-Agent Systems. pp. 1–17. Springer (2013)

11. Cabrio, E., Villata, S.: Combining textual entailment and argumentation theory for supporting online debates interactions. In: ACL. pp. 208–212 (2012)
12. Cayrol, C., Lagasquie-Schiex, M.C.: On the acceptability of arguments in bipolar argumentation frameworks. In: ECSQARU, Barcelona. pp. 378–389 (2005)
13. Cayrol, C., Lagasquie-Schiex, M.C.: Bipolarity in argumentation graphs: Towards a better understanding. International Journal of Approximate Reasoning **54**(7), pp–876 (2013)
14. Chesñevar, C.I., Maguitman, A.G., Loui, R.P.: Logical models of argument. ACM Computing Surveys (CSUR) **32**(4), 337–383 (2000)
15. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial intelligence **77**(2), 321–357 (1995)
16. Ennis, R.H.: Identifying implicit assumptions. Synthese **51**(1), 61–86 (1982)
17. Feng, V.W., Hirst, G.: Classifying arguments by scheme. In: ACL. pp. 987–996 (2011)
18. Florou, E., Konstantopoulos, S., Kukurikos, A., Karampiperis, P.: Argument extraction for supporting public policy formulation. In: LaTeCH (2013)
19. Freeman, J.B.: Argument strength, the toulmin model, and ampliative probability. Informal Logic **26**(1), 25–40 (2008)
20. Ghosh, D., Muresan, S., Wacholder, N., Aakhus, M., Mitsui, M.: Analyzing argumentative discourse units in online interactions. In: ArgMining Workshop. pp. 39–48 (2014)
21. Gough, J., Tindale, C.: 'missing' premises. Informal Logic **7**(2), 99 (1985)
22. Govier, T.: Analogies and missing premises. Informal Logic **11**(3) (1989)
23. Green, N.L.: Representation of argumentation in text with rhetorical structure theory. Argumentation **24**(2), 181–196 (2010)
24. Habernal, I., Gurevych, I.: What makes a convincing argument? empirical analysis and detecting attributes of convincingness in web argumentation. In: EMNLP. pp. 1214–1223 (2016)
25. Hitchcock, D.: Enthymematic arguments. In: Argumentation: Across the lines of discipline. Proceedings of the Conference on Argumentation. pp. 289–298 (1987)
26. Hitchcock, D., Verheij, B.: Arguing on the Toulmin model. Springer (2006)
27. Kobbe, J., Opitz, J., Becker, M., Hulpus, I., Stuckenschmidt, H., Frank, A.: Exploiting Background Knowledge for Argumentative Relation Classification. In: LDK. vol. 70, pp. 8:1–8:14. Dagstuhl, Germany (2019)
28. Kuhn, D., Pearsall, S.: Developmental origins of scientific thinking. Journal of cognition and Development **1**(1), 113–129 (2000)
29. Lawrence, J., Reed, C.: Combining argument mining techniques. In: ArgMining Workshop. pp. 127–136 (2015)
30. Lawrence, J., Reed, C.: Argument mining using argumentation scheme structures. In: COMMA. pp. 379–390 (2016)
31. Levi, D.S.: The case of the missing premise. Informal Logic **17**(1) (1995)
32. Levy, R., Bilu, Y., Hershcovich, D., Aharoni, E., Slonim, N.: Context dependent claim detection. In: COLING. pp. 1489–1500 (2014)
33. Lugini, L., Litman, D.: Argument component classification for classroom discussions. In: Workshop on Argument Mining. pp. 57–67 (2018)
34. Mayes, G.R.: Argument explanation complementarity and the structure of informal reasoning. Informal Logic **30**(1), 92–111 (2010)
35. Moens, M.F., Boiy, E., Palau, R.M., Reed, C.: Automatic detection of arguments in legal texts. In: Proceedings of ICAIL 2007 (2007)

36. Nguyen, H.N., Litman, D.J.: Context-aware argumentative relation mining. In: ACL. pp. 1127–1137 (2016)
37. Opitz, J., Frank, A.: Dissecting content and context in argumentative relation analysis. arXiv preprint arXiv:1906.03338 (2019)
38. Osborne, J.F., Patterson, A.: Scientific argument and explanation: A necessary distinction? Science Education **95**(4), 627–638 (2011)
39. Peldszus, A., Stede, M.: From argument diagrams to argumentation mining in texts: A survey. International Journal of Cognitive Informatics and Natural Intelligence (IJCINI) **7**, 1–31 (2013)
40. Peldszus, A., Stede, M.: Joint prediction in mst-style discourse parsing for argumentation mining. In: EMNLP. pp. 938–948 (2015)
41. Persing, I., Ng, V.: End-to-end argumentation mining in student essays. In: HLT-NAACL. pp. 1384–1394 (2016)
42. Pollock, J.L.: Defeasible reasoning. Cognitive science **11**(4), 481–518 (1987)
43. Rago, A., Cocarascu, O., Toni, F.: Argumentation-based recommendations: Fantastic explanations and how to find them. In: IJCAI. pp. 1949–1955 (2018)
44. Rinott, R., Dankin, L., Perez, C.A., Khapra, M.M., Aharoni, E., Slonim, N.: Show me your evidence - an automatic method for context dependent evidence detection. In: EMNLP. pp. 440–450 (2015)
45. Schank, R.C., Abelson, R.P.: Scripts, plans, and knowledge. In: IJCAI. pp. 151–157 (1975)
46. Schank, R.: Explanation patterns: Understanding mechanically and creatively. Psychology Press (2013)
47. Scriven, M.: Reasoning (1976)
48. Speer, R., Chin, J., Havasi, C.: ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In: AAAI (2017)
49. Stab, C., Gurevych, I.: Annotating argument components and relations in persuasive essays. In: COLING. pp. 1501–1510 (2014)
50. Stab, C., Gurevych, I.: Identifying argumentative discourse structures in persuasive essays. In: EMNLP. pp. 46–56 (2014)
51. Stab, C., Gurevych, I.: Parsing argumentation structures in persuasive essays. Computational Linguistics **43**, 619–659 (2017)
52. Teufel, S.: Argumentative zoning : Information extraction from scientific text (1999), doctoral dissertation, University of Edinburgh
53. Thagard, P.: Explanatory coherence. Behavioral and brain sciences **12**(3), 435–467 (1989)
54. Toulmin, S.E.: The uses of argument. Cambridge university press (2003)
55. Vreeswijk, G.: Reasoning with defeasible arguments: Examples and applications. In: JELIA. pp. 189–211. Springer (1992)
56. Wachsmuth, H., Stede, M., El Baff, R., Al Khatib, K., Skeppstedt, M., Stein, B.: Argumentation synthesis following rhetorical strategies. In: COLING. pp. 3753–3765 (2018)
57. Wachsmuth, H., Stein, B., Hirst, G., Prabhakaran, V., Bilu, Y., Hou, Y., Naderi, N., Thijm, T.A.: Computational argumentation quality assessment in natural language. In: EACL. pp. 176–187 (2017)
58. Walton, D.: Argument mining by applying argumentation schemes (2012)
59. Walton, D., Reed, C.A.: Argumentation schemes and enthymemes. Synthese **145**(3), 339–370 (2005)
60. Walton, D.N.: Enthymemes, common knowledge, and plausible inference. Philosophy & rhetoric **34**(2), 93–112 (2001)

Interactive Causal Discovery in Knowledge Graphs

Melanie MUNCH[1], Juliette DIBIE[1], Pierre-Henri WUILLEMIN[2], and Cristina MANFREDOTTI[1]

[1] UMR MIA-Paris, AgroParisTech, INRA, Paris-Saclay University, 75005 Paris, France

[2] Sorbonne University, UPMC, Univ Paris 06, CNRS UMR 7606, LIP6, 75005 Paris, France

Abstract. Being able to provide explanations about a domain is a hard task that requires from a probabilistic reasoning's viewpoint a causal knowledge about the domain variables, allowing one to predict how they can influence each others. However, causal discovery from data alone remains a challenging question. In this article, we introduce a way to tackle this question by presenting an interactive method to build a probabilistic relational model from any given relevant domain represented by a knowledge graph. Combining both ontological and expert knowledge, we define a set of constraints translated into a so-called relational schema. Such a relational schema can then be used to learn a probabilistic relational model, which allows causal discovery.

Keywords: Causal discovery, Probabilistic Relational Models, Knowledge Graph

1 Introduction

Probabilistic models such as Bayesian networks (BNs) are a good approach to represent complex domains, as they allow to express probabilistic links between variables. However, correlation does not imply causality, and thus these models lack explainability. Yet it could be useful when studying a disease to identify the cause (the actual illness) and the consequence (the symptoms). Uncovering causal relations from data alone is a difficult task: previous works have presented the use of interventions to construct causal models [21], but these interventions require to be able to change certain variables while keeping other constant, which is not always easily doable. Assessing for instance the impact of one's genotype and cigarettes smoking habits on lung cancer would theoretically require to intervene on both of these criteria. If controlling whether one is smoking or not

is possible (yet not really ethical), it is however impossible to directly control the genotype. As a consequence, for practical, ethical and economical reasons, direct interventions are often not available to learn causal relations. In this article, we present an interactive method that offers to introduce ontological and expert knowledge into the learning of a probabilistic model from a given knowledge graph (KG) [12], in order to discover causal knowledge. This causality helps to better explain a domains by allowing to reason on higher levels: a complete causal graph can answer causal questions such as "If I take this drug, will I still be sick?"; or even answer counter factual questions as "Had I not taken this drug, would I still be sick?". We propose to achieve this by using probabilistic relational models (PRMs) [14]. PRMs are an object-oriented extension of BNs, thus allowing a better representation between the different attributes. However, their learning can be tricky due to this specificity. Using the semantic and structural information contained in a KG , it can be greatly eased and, thus, be guided toward a learned model close to the reality. However, many different probabilistic models can be deduced from a same KG depending on the user (a domain expert) expectation. We present in this paper an interactive method to help such a user to build a probabilistic reasoning model from a KG able to answer his/her questions. The first section of this paper presents the background and state of the art, especially on PRM and causal discovery. The second section presents our approach to learn a PRM guided by the ontology and the user's knowledge. The third section presents an application of our method on a portion of DBPedia. The last section concludes this paper.

2 Background and State of the Art

The main idea of our method is to learn a probabilistic model under causal constraints given both by a user and the ontology. From the learned model we then are able to extract causal knowledge.

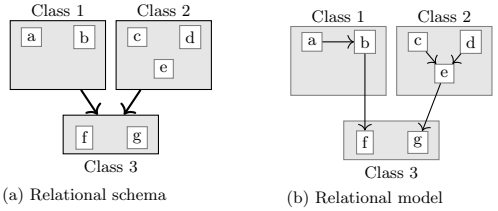
2.1 Probabilistic Models: BN and PRM

A BN is the representation of a joint probability over a set of random variables that uses a directed acyclic graph (DAG) to encode probabilistic relations between variables. Learning a BN requires learning both its structure and parameters. In our case, since learning is done under causal constraints, we need to express the conditional independence of this BN, which could give us new insight on the causality of this graph. Indeed, even if a correlation found between two variables of a BN does not prevail on the arc's orientation (explaining why causal discovery from data alone is difficult to achieve), some of these arcs also indicates conditional independence and are necessary to ensure the probabilistic information encoded in the BN. An essential graph (EG) [16] is a semi-directed graph associated to a BN. They both share the same skeleton, but the EG's edges' orientation depends on the BN's Markov equivalence class. If the edge's orientation is the same for all the equivalent BNs, then it means that its orientation

is necessary to keep the underlying probabilistic relations encoded in the graph: in this case, the edge is also oriented in the EG, and is called an **essential arc**. On the contrary, if the edge's orientation is not the same for all the equivalent BNs, then it means that its orientation can be both ways without changing the probabilistic relations, and it stays unoriented in the EG. Thus the EG expresses whether an orientation between two nodes can be reversed without modifying the probabilistic relations encoded in the graph: whenever the constraint given by an essential arc is violated, the conditional independence requirements are changed and the structure of the model itself has to be changed. With a BN learned under causal constraints such as in our method, the EG can then give us a new insight: if an arc is oriented, then it has to be kept if we want to conserve all the information we have provided during the learning.

However, our method also requires to use ontology's classes to group attributes by specific causal relations in order to learn them, and BNs lack such notion of modularity. As a consequence we turn to PRMs, that extend BNs' representation with the oriented-object notion of classes and instantiations. PRMs [14] are defined by two parts: a high-level, qualitative description of the structure of the domain that defines the classes and their attributes (i.e. the **relational schema** RS as shown Fig. 1 (a)), and a low-level, quantitative information given by the probability distribution over the different attributes (i.e. its **relational model** RM as shown in Fig. 1 (b)). Classes in the RS are linked together by so-called **relational slots**, that indicates the direction of probabilistic links. For instance, Fig. 1 has two classes 1 and 2 with a relational slot toward Class 3: it means that probabilistic links can exist between the attributes of class 1 and 2 with class 3's, and that they have to be oriented from the attributes of class 1 and 2 towards those of class 3. Using the RS structural constraints, each class can then be learned like a BN (in our case, we use the classical statistical methods Greedy Hill Climbing). As a consequence, a system of instantiated classes linked together is equivalent to a bigger BN composed of small repeated BNs, and thus can be associated to an EG.

Numerous related works have established that using constraints while learning BNs brings more efficient and accurate results, for parameters learning [9] or structure learning [10]. In case of smaller databases, constraining the learning can also greatly improve the accuracy of the model [19]. In this article we define structural constraints as an ordering between the different variables. The K2 algorithm [7], for instance, requires a complete ordering of the attributes before learning a BN, allowing the introduction of precedence constraints between the attributes. This particular algorithm needs a complete knowledge over all the different attributes precedences; however problems of learning with partial order have also been tackled [20]. In our case we will likewise transcribe incomplete knowledge as partial structural organization for the PRM's RS in order to discover new causal relations.



(a) Relational schema (b) Relational model

Fig. 1. The high (a) and low (b) level structures of a PRM

2.2 Causal Discovery

Causal models are DAGs allowing one to express causality between its different attributes [21]. Their construction is complex and requires interventions or controlled randomized interventions, which are often difficult or impossible to test. As a consequence the task of discovering causal relations using data, known as causal discovery, has been researched in various fields over the last few years. There are two types of methods for structure learning from data: independence-based ones, such as the PC algorithm [22], and score-based ones, such as Greedy Equivalent Search (GES) [6]. Usually independence-based methods give a better outlook on causality between the attributes by finding the *"true"* arc orientation, while the score-based ones offer a structure that maximizes the likelihood considering the data. Finally, other algorithms such as MIIC [23] use independence-based algorithms to obtain information considered as partially causal and thus allowing to discover latent variables. In this article we propose to explore if combining ontological and user's knowledge with BN learning score-based algorithms allows causal discovery. Other works have already proposed the use of EG: [15] for instance proposes two optimal strategies for suggesting interventions in order to learn causal models with score-based methods and the EG. Integrating knowledge in the learning has also been considered: [8] uses ontological causal knowledge to learn a BN and discover new causal relations with the EG; [4] offers a method to iterative causal discovery by integrating knowledge from beforehand designed ontologies to causal BN learning; [2] proposes two new scores for score-based algorithms using experts knowledge and their reliability; and [5] presents a tool combining ontological and causal knowledge in order to generate different argument and counterarguments in favor of different facts by defining enriched causal knowledge.

2.3 Ontology and Probabilistic Models

Using ontological knowledge in order to build probabilistic models has already been presented in numerous works. [13] uses the structure of an ontology to build and modify a BN by addressing three main tasks: the determination of the relevant variables, the determination of relevant properties and the computing of the probabilities. The learned model can then be used to reason on the domain. [1] presents a method for autonomic decision making combining BNs and

ontologies, using the framework BayesOWL [11]. This framework allow the expression of a BN using the OWL standardization, and offers a set of rules aiming to automate the translation from an ontology to a BN. [3] presents a method to generate Object Oriented Bayesian Networks from ontologies using a set of rules they have defined. While PRM offers a way to express and consider the expert knowledge in learning, to the best of our knowledge no causality learning method that combines ontological and user's knowledge has been proposed yet.

3 Causal Discovery Driven by an Ontology

In this article we present an interactive method aiming to build a *RS* from a KG relying on the ontological and user's knowledge. This *RS* presents the different PRM's classes, relational slots and attributes, and is used to learn a PRM under causal constraints, allowing the deduction of causal knowledge. This method is split into three parts: (1) building a first *RS* from the ontological knowledge; (2) helping the user improving the proposed *RS*; (3) learning a PRM from the *RS* from which causal knowledge can be deduced. In a previous work [17] we present a method to help the user to build the *RS* but without fully exploiting the ontological knowledge. In this article, we focus on the first and second parts.

3.1 Relevant KGs

In theory, a PRM can be learned from any knowledge graph. However, not all are interesting to do so and some **selection criteria (SC)** must be fulfilled in order to learn a relevant probabilistic reasoning model. As an illustration we define a simple ontology dedicated to an university representation (Fig.2). It is composed of three main classes: the University class, the Student class and the Course class. An university is defined by its name and its fees; a student is defined by his/her name, sex, social standing, mean note and his/her subject of interest; a course is defined by its subject and its difficulty.

SC1. **The domain the KG is dedicated to must contain causal information to be deduced.** Our model can be used to simply discover simple probabilistic relations. However, it best shines when it encompasses causal knowledge, as it allows a far better explainability of the represented domain. Therefore, the user must have a causal question or at least an idea to search for causality information. In our university example, one might be interested in studying the influence of a student's social standing with his/her choice of courses and university.

SC2. **The KG contains datatype properties (DPs) whose values can be discretized.** The PRM's learning is based on classical BNs learning methods, which uses statistical analysis to learn the probabilistic relations. Therefore, our method needs data, which is given by DPs: they define our model's attributes. As a consequence, they must be relevant for the domain and their values discretizable for the learning: a DP indicating a student's ID is not interesting, as it is different for each student.

SC3. **The classes of the KG are instantiated enough and there are not too many missing DPs.** As stated before, the learning is based on statistical methods. As a consequence the studied KG must have enough instantiations in order to study their variability. Since all instances of the same class are compared together using their DPs, each instance's missing DP is considered as a missing value: as a consequence, each missing DP can decrease the precision of the model. For example, a single student's instance would not be enough to study the relations between a student and his/her courses; likewise, if we have multiple student's instances, but only one of them has a DP about his/her social standing, then we will not be able to study the influence of social standing over other parameters.

In order to deal with the causality, we consider in this article that the KG is **complete** and **verified**: all important variables are present (no confounding factor possible), and the distribution of the different values is balanced (none is arbitrarily prominent over others). Confounding factors occur when a correlation is found between two attributes, but with no direct causal link, and that the explanatory variable is missing. A classical example is the study of the correlation between one's reading ability and shoes' size: while both are indeed correlated, it is arguably not due to the fact that one causes the other. In this example, one's age is a confounding example, as it explains both: the older we are, the better we can read and the bigger our shoes are. As a consequence, confounding factors can lead to false causal reasoning, and must be avoided. In the rest of this article, we will consider that it is possible to learn from our data the true causal model of the domain (or at least a part of it). In the case where those criteria cannot be satisfied, then the causal learning could not be guarantied.

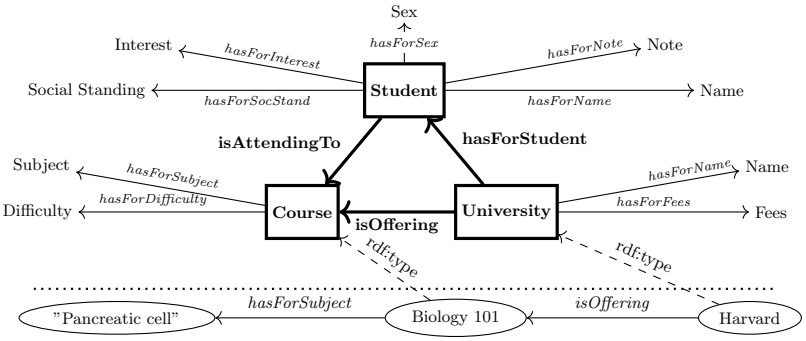


Fig. 2. Excerpt of a KG about students and universities

3.2 Interactive Building of a RS

From the ontological knowledge we automatically generate a first *RS* draft. The aim of this generation is to give the user a good preliminary overview of the

KG in order to help him/her build a probabilistic reasoning model. This transformation is done in three steps: (1) All ontology's classes become *RS* classes. With our university ontology we thus have three *RS* classes, University, Student and Course. (2) All ontology's DPs become attributes in the *RS* associated to their respective *RS* classes. In our example the University *RS* class owns two attributes, Name and Fees. (3) All ontology's object properties (OPs) become relational slots in the *RS*. In our example, the University *RS* class has two relational slots: one toward the Student *RS* class, and one toward the Course *RS* class. Before presenting to the user, we apply automatic **selection rules (SR)** based on the selection criteria presented above that directly modify the *RS*:

SR1. **The *RS* classes with too few instances are removed.**[3]

SR2. **The isolated *RS* classes are deleted.** If by applying **SR1** we break a path between two others *RS* classes, leading to the isolation of one of them (meaning there is no other relational slots linking this *RS* class), then the isolated *RS* class is also removed. We can illustrate this by adding a new OP in our example, *hasForTeacher*, taking for *domain* the Student class and for *range* a new Teacher class. In a regular situation, we would then be able to study the probabilistic relations between a teacher and a student, or a teacher and an university. However, if the student instances are not numerous enough to learn, then the Student *RS* class has to be removed, leaving the Teacher *RS* class isolated. As a consequence, it would not be possible anymore to study the probabilistic relations between a teacher and a university: the Teacher *RS* class has also to be removed.

SR3 **The attributes must be useful.** Since the learning of the PRM is based on statistical methods, problematic variables such as ones with too many missing data, values that do not repeat (for example IDs different for each instance) or that are not different (if we study a single university, its name is useless) are to be removed from the *RS*. In our example, if we had 50 students but only 3 with a DP about their social standing, then this corresponding attribute cannot be used to learn and is removed from the *RS*.

SR4 **The symmetric relational slots are deleted.** The PRM does not support cyclic relations, symmetric OPs cannot therefore be kept: as a consequence one of the corresponding relational slot in the RS must be discarded. In a first time, we automatically keep if possible the relational slot that corresponds to the most instantiated OP; if not, we randomly select one.

Once defined the *RS* is presented to the user who can intervene on different points. These **user modifications (UMs)** also directly modifies the *RS*:

UM1. **The choice of attributes.** Despite being instantiated enough, some selected DPs may be irrelevant according to the user, and thus their corresponding attributes need to be removed.

UM2. **The choice of relational slots.** The orientation of the relational slots has a great influence on the causal learning: if there is a relational slot from a

[3] The accepted missing values ratio is determined with the user.

class A to a class B, then all probabilistic links learned between attributes of class A and B have to be identically oriented. Broadly speaking it means that class A's attributes can explain class's B attributes, but not the contrary. However, not all ontology's OP are causal by default: as a consequence we need the user to validate when possible the orientation of the relational slots, or reverse it to express causality. He is also able to remove or add relations slots between classes if necessary.

UM3. **The choice of *RS* classes.** The orientation of the relational slots have a great influence over the learning of the causal knowledge. However, some *RS* classes' attributes might be intricate, meaning that two *RS* classes can be both explaining of and explained by a same other *RS* class. In our example, we can consider the relation between a student and his/her courses: the student's interest might explain his/her courses' subject; however, the courses' difficulty might explain the student's note. Fig. 3 (a) shows a first *RS*, in which both the interest and the note can explain the course's subject and difficulty. This is inconsistent with the idea that, on the contrary, the course's difficulty should explained the student's note. As a consequence, we offer the user a tool to split the *RS* classes in order to reflect this causal information. In Fig. 3 (b), the Student *RS* class has been split in two: a first *RS* class above with the interest attribute that can still explain the course's attributes, and a second below with the note attribute that can be explained by both the student's interest and the course's subject and difficulty.

UM4. **The choice of attributes.** As mentioned before the user can choose whether a DP can be kept or not in the *RS*. By default, a DP is directly translated into an attribute. However, when multiple identical DPs are involved it requires an intervention of the user: it can be the case when a single instance has several time the same DP (such as a Student who has multiple interests), or when a same *RS* class's instance can be explained (through a relational slot) by multiple instances of another *RS* class (e.g. a single course instance can be attended by many students). Here, the repeated DPs cannot be distinguished given the ontology: in these particular cases, we need to aggregate the given DP in order to allow a statistical learning. The aggregation can take many forms, depending on what the user wants (e.g. the mean value, the maximum value, if a certain value is present or not). For example, if we consider that a single course can have a variable number of different students, then it is not possible to learn a statistical model: some course will have 5 students, other 30, 12... No comparison is possible, and even if two courses had the same number of students, there is no way to distinguish one from another. As a consequence we need to transform these possible multiple attributes in the *RS* in an unique one, which is what aggregation allows us to do. For instance, instead of considering all the student's notes, we calculate the mean value: each course now have one attribute for the note, whether they had 1 or 100 students in the beginning. Aggregator must be defined by the user. If no aggregator can be found to characterize an aggregated attribute corresponding to a group of DPs, then this group of corresponding DPs attributes must be removed from the *RS*.

UM5. **The choice of instances in the KG.** Sometimes the user wants to be able to study only a particular part of the KG (e.g. students that are registered in at least one course). This UM allows some conditions to be defined in order to select the instances that are consistent with the building of the *RS*: if we have a relational slot from the University *RS* class to the Student *RS* class, then all student instances in the KG must be registered in an university.

Once the user has done all the modifications he deemed necessary, we can learn the probabilistic model using the *RS*.

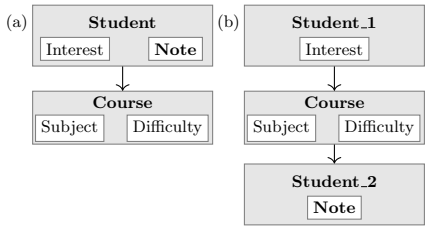


Fig. 3. Example of a *RS* class splitting in a PRM as detailed in **UM3**. The Student *RS* class of (a) has been split in two *RS* classes Student_1 and Student_2 in (b).

3.3 Model Validation

The *RS* has been defined using constraints from both the ontological and the user's knowledge. As a consequence the PRM learned using this *RS* has been learned under causal constraints, and then can be used to deduce causal knowledge. However, the *RS* are not good enough to discover new causal relations. Since it is easier for a user to criticize when confronted to mistakes, we have devised a method to validate the learned model [17].

First, the inter *RS* classes relations are presented to the user. Those relations flow directly from the relational slots defined during the *RS* building: their orientation has been fixed either by the ontology or by the user. They are thus easier to criticize for the user than if they have been built from scratch: if their orientation contradicts a piece of knowledge the user has about the domain, then the *RS* has been badly constructed, and has to be reconsidered. Then, the intra *RS* classes relations are presented. Their orientation is not ruled by the *RS*, so in order to criticize them we need to look at the EG. If this arc is not an essential arc, then it can be reversed without consequences; however, if it is not, then the *RS* has to be modified in order to reflect this change. Finally, if the user challenges a learned relation that should not exist (for instance, between two attributes he knows are independent), then it means that the KG is not balanced enough: for example, scientists that might have tested too much of an hypothesis and not enough of an other. In this case, we cannot continue, as our data is not robust enough to deduce causality.

3.4 Causal Knowledge Discovery

Once the *RS* has been built using the ontological and user's knowledge (Sec. 3.2) and the learned model validated by the user (Sec. 3.3), we can use it to discover causal knowledge. Causal knowledge can be validated by three means:

- **the Ontology**: the orientation of a learned relation between attributes from two different *RS* classes defined by the ontology (e.g. between a student and his/her university) has been constrained by its causal information.
- **the User**: During the *RS*'s interactive building, the user was able to inject causal knowledge with UMs. If a relation is learned between two attributes from two *RS* classes (or whose relational slot has been) defined by the user (e.g. between the classes Student_1 and Course in Fig. 3), then the learning has been constraint by the user who validates the causal knowledge discovery.
- **the EG**: Since the model has been learned under causal constraints given by the ontology and the user, the EG's essential arcs can give causal information. Indeed, an oriented arc in the EG is oriented for all of Markov's equivalence's classes of the learned BN, meaning that, if our model has been learned under right conditions (i.e. complete data set, good given constraints), then it is highly probably causal, allowing the discovery of causal knowledge between attributes of a same *RS* class (e.g. a student's Interest and his/her Note).

The interest of this discovery has two goals: first, it can help a user validate his/her hypothesis on a domain; second, it can suggest new experiments to conduct to test new hypothesis. For instance, using this method, [18] suggests a strong link between plausible control variables and some parameters of the studied cheese, whereas it also indicates that some other experiments had to be conducted to understand the whole process.

4 Application on DBpedia

We illustrate our method with a part of the DBpedia[4] KG dedicated to writers.

4.1 Dataset Presentation

The DBpedia database collects and organize all available information from the Wikipedia[5] encyclopedia. Since it describes 4.58 million things (including persons, places, ...), we have decided for our test to only study a small part of it, on a subject simple enough where we could easily play the role of an expert. As a consequence, we have restrained our study to a much smaller KG[6], dedicated to writers. During this first pre-selection, we have selected four classes to represent our domain: Writer, University, Country and Book. The selected KG is presented in Fig. 4. Considering all possible DPs for every instances of these classes, and also all OPs between them, we have a dataset of 2,966,073 triples.

[4] https://wiki.dbpedia.org/
[5] https://www.wikipedia.org/
[6] https://bit.ly/2X0eeCw

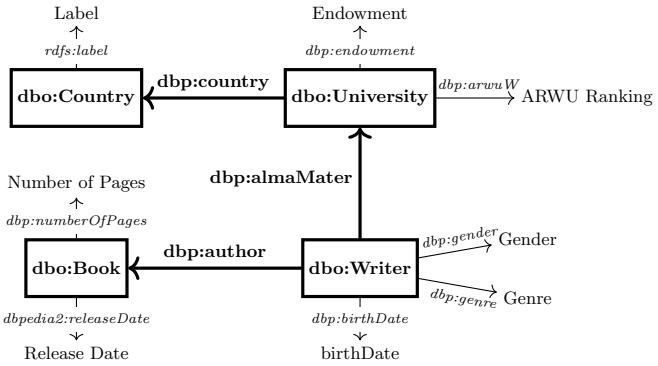


Fig. 4. Schema of the used excerpt of DBPedia with the DPs kept in the final *RS*.

4.2 Interactive Construction of the *RS*

First, we translate all the selected classes as new *RS*'s classes, and all DPs as new *RS*'s attributes. In our case, there is no symmetric OPs, so we keep the original ones present in DBpedia (as depicted in Fig. 4) to define the direction of the relational slots. By applying the selection rule **SR3**, a first automatic selection removes all attributes that correspond to DPs that are not represented enough: for instance, over the 32,511 instances of writers, only 12,188 have the DP *occupation*. This selection is coupled with the expert selection using **UM1** which removes attributes that correspond to uninteresting DPs. We also apply **UM5**, which filters some instances: for example, in our case, we want to study writers that have written books. However, on the whole database, only 6,028 writers instances are linked to at least one book instance. As a consequence, we remove authors with no books since they are out of the scope of our study. Then as a user we apply **UM2**. Since we consider that a country can explains the values of an university's variables, and not the contrary: we reverse the relational slot corresponding to the OP *dbp:country*. One country can have multiple universities, but one university can only have one country: reversing the relational slot removes the aggregation of universities and creates a simple linear relation, since now one university can be explained by at most one country. Moreover, we want to study the possible influence of an university over a writer's work, so we need to reverse the relational slot corresponding to the OP *dbp:almaMatter*. Since a person can register in one or more universities, then his/her attributes can be explained by a combination of his/her universities'. We apply **UM4** and create an aggregation from universities to writers. For each writer, we create two aggregated variables: the highest rank and the highest endowment among all of the universities he/she went to. But doing so break the relation between the Country *RS* class and the Writer *RS* class, since they are linked trough the University *RS* class. The only way to keep a relational slot between the country and the writer is to also aggregate the country's attributes. However, the only available country's attribute is the label, and there is no way of intelligently ag-

gregating it. As a consequence, with the aggregation of universities, we loose the information about countries for the writers and their books. In the end thanks to the rule **SR3**, only interesting attributes which have no missing values and are easily discretizable are kept. For each class, we keep the following attributes:

- **dbo:Country**: each country is only represented by its **label**. Since the majority of our writers are Anglo-Saxon, we distinguish five categories: USA, Canada, Great Britain, Europa and Asia.
- **dbo:University**: each university is represented by its **Academic Ranking of World Universities** (ARWU), and its **endowment**. The endowment is split by its median value. The ARWU ranking is split between the first hundred universities, and the rest.
- **dbo:Writer**: each writer is represented by his/her **gender**, his/her **genre** and his/her **birth date**. Genders are split between male and female, while genres are split between fiction and non-fiction. Birth dates are separated by their median, 1950. Two aggregated attributes have been also added: the **highest rank** among all universities he/she went to, and the **highest endowment** he/she went to, with the same discretization used before.
- **dbo:Book**: each book is represented by its **number of pages** and its **release date**. The number of pages is split between books with 250 pages or less and the others; the release date attribute is split between books published before 1980 and those published after.

In the end we have drastically dropped the number of instances to 6,908 triples and 185 writers. The final *RS* defined both by ontological and user's knowledge is presented in Fig. 5. The direction of relational slots indicates how the considered variables can influence each other: for instance, a writer's genre or highest university rank can influence the number of pages of his/her books.

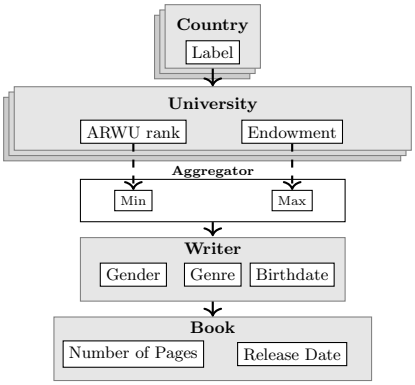


Fig. 5. Relation Schema defined from ontological and user's knowledge. Since a writer can have multiple universities, we introduced an aggregation between the two classes.

4.3 Results

Using the dataset and the RS, it is now possible to learn a PRM and study its EG (respectfully Fig. 6 (a) and (b)). We apply the discretization presented in Sec.4.2, and consider any missing data as a new category "Unknown".

Inter RS classes relations. We have three inter RS classes relations: one between Label and Endowment, one between the highest ARWU rank and the book's release date, and another one between the author's birth date and the book's release date. Since the RS classes was built from the ontology, and the relational slot's direction decided by the user, then we have a causal discovery validated by both the ontological and user's knowledge.

Intra RS class relations. Three relations are oriented in the EG (see Fig. 1 (b)), but only one is an intra RS class relation: from Release Date toward Number of Pages. Thus, the causality of this relation is validated by the EG. There is another intra RS class relation (between ARWU Rank and the Endowment), but it is not oriented in the EG: the given RS and dataset are not enough to assume the causality between those two attributes.

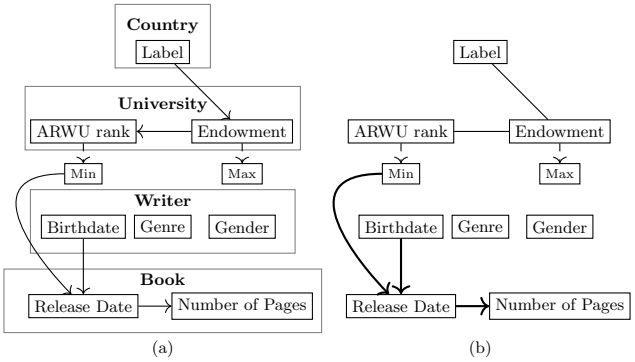


Fig. 6. (a) PRM learned. Plain arrows indicates probabilistic relations. (b) Associated EG. Plain arrows indicates essential arcs, unoriented ones indicate the edges. Dashed arrows only serve as a visual cue to indicate aggregation.

4.4 Discussion

Despite not being experts of the domain, most of our results appears to agree with common sense. For instance, it seems logical that an university's ARWU rank and its endowment are correlated, itself explained by its university's country. However our KG's representativeness casts doubts on other results. For instance, we find that a book's release date can be explained by both the highest rank of the university its author went to, and this author's birth date (the

joint probability is presented in Table. 1). Basically, authors born before 1950 tend to publish more before 1980 when they are from a top-tiers school. On an other hand, youngest authors tend to publish after 1980, which at first seems logical: writers born after 1980 would hardly be able to publish books prior to their birth. However, we have no instance in our dataset of books published before 1980 written by persons born after 1950, which explain why we learned this relation. This underlines the importance of a complete and verified KG: if our dataset is representative, then we acknowledge the fact youngest authors cannot publish before 1980. On another hand, if our dataset is not representative, then it means that our learned relation cannot be causal, as we are missing arguments. In the end, the main point of this example is to illustrate our method:

1. The *RS* construction from the KG is simplified thanks to selection rules that preemptively remove *RS* classes, attributes... that are not learnable. In our case, numerous attributes corresponding to DPs with not enough instantiations were removed (such as *dbp:occupation* for the writer).
2. The user introduced causal knowledge in the *RS* with UMs: **UM1** to remove attributes irrelevant for the problem (e.g. the wikipedia page ID), **UM2** to reverse relational slot to express causality (e.g. between a writer and his/her universities), **UM4** to formulate aggregations (e.g. since writers had a variable number of universities, we had to aggregate the universities' attributes), and **UM5** to remove instances that did not have certain properties (e.g. all writers with no book or no birth date).

writer.birthDate	**writer.min_arwu**	**releaseDate**	
		before_1980	*after_1980*
before_1950	*100_or_less*	0.58	0.42
after_1950	*100_or_less*	0.01	0.99
before_1950	*101_or_more*	0.44	0.56
after_1950	*101_or_more*	0.01	0.99

Table 1. Joint probability of the attribute **releaseDate** depending on the attributes **writer.birthDate** and **writer.min_arwu**. The low values 0.01 are an artefact of learning, and indicates that these combinations are not encountered in the dataset.

UM3 was not used here. However, should we have had a variable about an author's success, it would then have been possible to study the impact of an author's books on his/her success. To do so, we would have split the author *RS* class in two, to see how an aggregation of the books' attributes would have influenced this variable. Fig.7 presents the corresponding *RS*: we can see that since it is the same *RS* class split in two, both the writer's other attributes (genre, gender, birth date) and the aggregated book attributes (mean number of pages, oldest release date) can explain the writer's success.

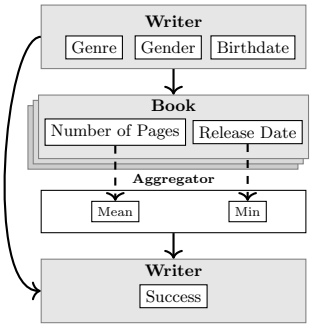


Fig. 7. Example of a *RS* class split with creation of an aggregation.

5 Conclusion

While causal knowledge can be useful for explaining a domain, causal discovery is a hard task, especially from data alone. In this paper, we present an interactive method aiming to allow a user to combine his/her knowledge with that of a KG in order to learn a probabilistic model from a KG able to help him/her uncover new causal explanations. The main idea is to combine the knowledge of both of these sources in order to interactively build a *RS* able to guide and causally constraint the learning of a PRM. This method is split into three parts: (1) automatic design of a first *RS* from the KG; (2) modification of this *RS* by the user; (3) learning of the PRM using the *RS*. This method is **interactive** (i.e. the user can interact with the algorithm to give his/her inputs and influence the learning) and **generic** (i.e. it can be applied on any KG as long as it is relevant for causal discovery). It is also dependant on the quality of the dataset: it has to be checked (i.e. no errors) and complete (i.e. no missing attributes or incomplete data). Our future work will focus on the explanation of the discovered causal relations in order to help the user to improve his/her knowledge (e.g. by enriching the ontology) and clarify his/her reasoning needs.

References

1. Aguilar, J., Torres, J., Aguilar, K.: Autonomie decision making based on bayesian networks and ontologies. pp. 3825–3832 (07 2016)
2. Amirkhani, H., Rahmati, M., Lucas, P.J.F., Hommersom, A.: Exploiting experts knowledge for structure learning of bayesian networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 39(11), 2154–2170 (Nov 2017)
3. Ben Ishak, M., Leray, P., Ben Amor, N.: Ontology-based generation of object oriented bayesian networks. vol. 818, pp. 9–17 (01 2011)
4. Ben Messaoud, M., Leray, P., Ben Amor, N.: Integrating ontological knowledge for iterative causal discovery and visualization. In: Sossai, C., Chemello, G. (eds.) Symbolic and Quantitative Approaches to Reasoning with Uncertainty. pp. 168–179 (2009)

5. Besnard, P., Cordier, M., Moinard, Y.: Arguments using ontological and causal knowledge. In: Foundations of Information and Knowledge Systems - 8th International Symposium, FoIKS 2014, Bordeaux, France, March 3-7, 2014. Proceedings. pp. 79–96 (2014)
6. Chickering, D.M.: Optimal structure identification with greedy search. J. Mach. Learn. Res. 3, 507–554 (Mar 2003)
7. Cooper, G.F., Herskovits, E.: A bayesian method for the induction of probabilistic networks from data. Machine Learning 9(4), 309–347 (Oct 1992)
8. Ćutić, D., Gini, G.: Creating causal representations from ontologies and bayesian networks (2014)
9. De Campos, C.P., Ji, Q.: Improving bayesian network parameter learning using constraints. In: 2008 19th International Conference on Pattern Recognition. pp. 1–4 (Dec 2008)
10. De Campos, C., Zhi, Z., Ji, Q.: Structure learning of bayesian networks using constraints. In: Proceedings of the 26th Annual International Conference on Machine Learning. pp. 113–120. ICML '09, ACM, New York, USA (2009)
11. Ding, Z., Peng, Y., Pan, R.: BayesOWL: Uncertainty Modeling in Semantic Web Ontologies, pp. 3–29. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
12. Ehrlinger, L., W, W.: Towards a definition of knowledge graphs (09 2016)
13. Fenz, S.: Exploiting experts knowledge for structure learning of bayesian networks. Data & Knowledge Engineering 73, 73 – 88 (2012)
14. Friedman, N., Getoor, L., Koller, D., Pfeffer, A.: Learning probabilistic relational models. In: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI 99, Stockholm, Sweden, July 31 - August 6, 1999. 2 Volumes, 1450 pages. pp. 1300–1309 (1999)
15. Hauser, A., Bühlmann, P.: Two optimal strategies for active learning of causal models from interventional data. Int. J. Approx. Reasoning 55, 926–939 (2014)
16. Madigan, D., Andersson, S.A., Perlman, M.D., Volinsky, C.T.: Bayesian model averaging and model selection for markov equivalence classes of acyclic digraphs. Communications in Statistics–Theory and Methods 25(11), 2493–2519 (1996)
17. Munch, M., Dibie, J., Wuillemin, P., Manfredotti, C.E.: Towards interactive causal relation discovery driven by an ontology. In: Proceedings of the Thirty-Second International Florida Artificial Intelligence Research Society Conference, Sarasota, Florida, USA, May 19-22 2019. [17], pp. 504–508
18. Munch, M., Wuillemin, P., Dibie, J., Manfredotti, C.E., Allard, T., Buchin, S., Guichard, E.: Identifying control parameters in cheese fabrication process using precedence constraints. In: Discovery Science - 21st International Conference, DS 2018, Limassol, Cyprus, October 29-31, 2018, Proceedings. pp. 421–434 (2018)
19. Munch, M., Wuillemin, P.H., Manfredotti, C., Dibie, J., Dervaux, S.: Learning probabilistic relational models using an ontology of transformation processes. In: On the Move to Meaningful Internet Systems. OTM 2017 Conferences. pp. 198–215 (2017)
20. Parviainen, P., Koivisto, M.: Finding optimal bayesian networks using precedence constraints. Journal of Machine Learning Research 14, 1387–1415 (2013)
21. Pearl, J.: Causality: Models, Reasoning and Inference. Cambridge University Press, New York, USA, 2nd edn. (2009)
22. Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction, and Search. MIT press, 2nd edn. (2000)
23. Verny, L., Sella, N., Affeldt, S., Singh, P.P., Isambert, H.: Learning causal networks with latent variables from multivariate information in genomic data. PLOS Computational Biology 13(10), e1005662 (2017)