WoLE2012
ISWC2012 Workshop

Web of Linked Entities

Workshop in conjunction with the
11th International
Semantic Web Conference
Boston USA, November 11th 2012
http://wole2012.eurecom.fr/

Edited by:
Giuseppe Rizzo
Pablo N. Mendes
Eric Charton
Sebastian Hellmann
Aditya Kalyanpur

Introduction

The WoLE 2012 workshop envisions the Semantic Web as a Web of Linked Entities (WoLE), which transparently connects the World Wide Web (WWW) and the Giant Global Graph (GGG) using methods from Information Retrieval (IR) and Natural Language Processing (NLP).

Topics of interest to WoLE 2012 include:

1. Improvements upon the state of the art in NLP using information in the Linked Open Data (LOD) space;
2. Knowledge extraction from text and HTML documents (or other structured and semi-structured documents) on the Web, with a special focus on scalability, evaluation of precision & recall and/or real-time systems;
3. Representation of NLP tool output and NLP resources as RDF/OWL and especially connections to linked data;
4. Novel applications to search and browse the WWW with the help of extracted knowledge and the Web of Data.

The focus of this workshop is to reconcile the communities of Information Retrieval, Semantic Web and NLP. The primary goal is to strengthen research techniques that provide access to textual information published on the Web to further improve the adoption of Semantic Web technology.

Motivation

Most of the knowledge available on the Web is present as natural language text enclosed in Web documents aimed at human consumption. A promising approach to have programmatic access to such knowledge uses information extraction techniques in order to reduce texts written in natural languages to machine readable structures, from which it is possible to retrieve entities and relations. The Natural Language Processing (NLP) community has been approaching this crucial task for the past few decades, with two major guidelines: establishing standards for various tasks, and metrics to evaluate the performance of algorithms. Scientific evaluation campaigns, starting in 2003 with CoNLL, ACE (2005, 2007), TAC (2009, 2010, 2011, 2012), and ETAPE in 2012 were proposed to involve and compare the performance of various systems in a rigorous and reproducible manner. Various techniques have been proposed along this period to recognize entities mentioned in text and to classify them according to a small set of entity types.

Recently, an increasing number of researchers have investigated information extraction techniques in the context of Semantic Web research. Working in the intersection with the NLP community, researchers have used fine grained ontologies to classify entities and proposed disambiguation techniques to map these pieces of information to real world entities. Therefore, the Web represents a vital lookup space where entities extracted from textual documents can be disambiguated. Moreover, it offers a broad range of relationships that already exist among entities. The landscape of available techniques vary on their approaches and performance, leading to new evaluation campaigns being proposed -- focusing on the extraction of richer information as compared to previous work. The final results of such information extraction tasks may potentially be consumed in the LOD cloud, as exemplified by efforts of the NLP2RDF/NIF community.

The focus of this workshop is to strengthen the connection between the communities of Information Retrieval, Semantic Web and NLP -- reconciling research and techniques that provide access to textual information published on the Web to further improve the adoption of Semantic Web technology.

Programme Committee

The following colleagues kindly served in the workshop’s program committee. Their joint expertise covers all of the questions addressed in the workshop, and they reflect the range of relevant scientific communities.

- Caroline Barriere, Centre de Recherche Informatique de Montréal, DETI, Canada

- Frédéric Béchet, Université d'Aix-Marseille, LIF, France
- Andreas Blumauer, Semantic Web Company
- Paul Buitelaar, DERI/National University of Ireland, Galway, Ireland
- Philipp Cimiano, CITEC, University of Bielefeld, Germany
- Eric de la Clergerie, INRIA, France
- Christian Dirschl, Wolters Kluwer, Germany
- Benoit Favre, Université d'Aix-Marseille, LIF, France
- Michel Gagnon, École Polytechnique de Montréal, Canada
- Daniel Gerber, AKSW, Universität Leipzig, Germany
- Claudio Giuliano, Fondazione Bruno Kessler, Italy
- Jiafeng Guo, Institute of Computing Technology, China
- Daniel Hladky, Ontos AG
- Guy Lapalme, Université de Montréal, RALI-DIRO, Canada
- Paul McNamee, Johns Hopkins University, USA
- Marie-Jean Meurs, Semantic Software Lab, CSFG, Concordia University, Canada
- Meenakshi Nagarajan, IBM Research, USA
- Axel-C Ngonga Ngomo, AKSW, Universität Leipzig, Germany
- Cartic Ramakrishnan, ISI, University of Southern California, USA
- Ganesh Ramakrishnan, IIT Bombay, India
- Harald Sack, HPI, University of Potsdam, Germany
- Benoit Sagot, INRIA, France
- Felix Sasaki, DFKI-LT, German Research Center for Artificial Intelligence (DFKI), Germany
- Tomas Steiner, Universitat de Catalunya, Spain
- Fabian Suchanek, Max-Planck Institute for Informatics, Saarbrücken, Germany
- Vojtech Svatek, University of Economics, Prague
- Krishnaprasad Thirunarayan , Wright State University, USA
- Andraz Tori, Zemanta
- Ruben Verborgh, IBBT, Ghent University, Belgium
- Mateja Verlic, Zemanta
- Wouter Weerkamp, University of Amsterdamm, Netherlands
- Gerhard Weikum, Max-Planck Institute for Informatics, Saarbrücken, Germany
- René Witte, University of Montréal, Canada

- Feiyu Xu, DFKI-LT, German Research Center for Artificial Intelligence (DFKI), Germany

Additional reviewers:

- Magnus Knuth, University of Potsdam, Germany
- James Mayfield, Johns Hopkins University, USA
- Nadine Steinmetz, University of Potsdam, Germany

Organizing Committee

The WoLE 2012 workshop was organised by:

- Giuseppe Rizzo, EURECOM, France
- Pablo N. Mendes, Freie Universität Berlin, Germany,
- Eric Charton, Centre de Recherche Informatique de Montréal, Canada
- Sebastian Hellmann, Universität Leipzig, Germany
- Aditya Kalyanpur, IBM, USA

Enabling Russian National Knowledge with Linked Open Data (LOD Russia)

Daniel Hladky, Victor Klintsov, Grigory Drobyazko

National Research University – Higher School of Economics (NRU HSE),
Moscow/Russia
`{dhladky, vklintsov, gdrobyazko}@hse.ru`

Abstract. The LOD Russia research project funded by the Ministry of Education aims to create a first Linked Open Data Set in Russia enabling scientists, researchers and commercial users to share, access, analyse and reuse knowledge related to scientific data. The position paper is highlighting challenges of the life-cycle management of LOD data, especially focuses on the process of entity linking and the creation of a unique identifier (UID) based on the concept of the Identification Knowledge Base (IKB).

The publication is prepared with the financial support of the Ministry of Education and Science of the Russian Federation within the framework of the State Contract № 07.524.11.4005 of October 20, 2011.

Keywords: Linked Data, NLP, RDF, UID disambiguation

1 Introduction

The World Wide Web has enabled the creation of a global information space compromising linked documents. Under the umbrella of the Russian National Knowledge several Russian ministries have enabled projects that pursuit the desire to access scientific data not currently available on the Web or bound up in hypertext documents. Linked Data provides a publishing paradigm in which not only documents, but also data, can be first class citizen of the Web, thereby enabling the extension of the Web with a global data space based on open standards promoted by the W3C[1]. Based on first Russian experiences of e-Arena[2] and DANTE[3] the LOD Russia project is designed to build use cases for scientific data related to nanotechnology and mathematics. The research project has a runtime of 600 man-days and the project will finish by June 2013. The goals are to semantize various scientific papers, patents, research projects and create Linked Data sets based on a domain knowledge driven vocabulary. The access to the data should not only provide a better search but the use case shall also support analytical functions in order to make better decisions. Various stakeholders like scientists, researchers and lawyers shall have the possibility to use the data

---

[1] http://www.w3.org/

[2] http://en.e-arena.ru/

[3] http://www.dante.net/

adfa, p. 1, 2011.

sets and have different reports and analysis according the individual needs. This position paper will focus mainly on the process of creating unique identifiers (UID) and the merging of the UID using the Identification Knowledge Base (IKB).

2 LOD project

The consortium consisting of NRU HSE, Avicomp Services[4] and AKSW University of Leipzig[5] have identified different tasks within the process of creating the linked data sets. The simplified view can be described as such:

- Crawl/Harvest sources and create plain text
- Large-scale information extraction from unstructured, semi-structured heterogeneous sources using rule based NLP, statistical methods, background knowledge and bootstrapping.
- Process objects using the Identification Knowledge Base (IKB) for named entity disambiguation and the creation of unique identifier (UID).
- Aggregating the data into a scalable RDF store applying methods for interlinking named entities with the LOD.

The size of the test corpus consists of 15’000 documents related to mathematics and 100’000 documents related to nanotechnology resulting in approximately 2.5 million triples for the RDF store. Based on the current ontology we expect to create links to another 20 data sets on the Web[6]. The focus of the position paper is on the automatic creation of an UID and the merging of same objects (named entities) as shown in Figure 1.

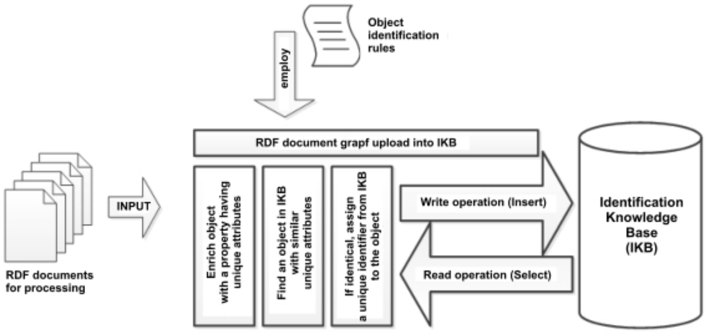

Object identification rules
employ
RDF document grapf upload into IKB
RDF documents for processing
INPUT
Enrich object with a property having unique attributes
Find an object in IKB with similar unique attributes
If identical, assign a unique identifier from IKB to the object
Write operation (Insert)
Read operation (Select)
Identification Knowledge Base (IKB)

Figure 1 – Simplified process of UID creation using the IKB

---

[4] http://avicomp.ru/

[5] http://aksw.org/About

[6] Datasets (excerpt) incl.: The Open Library, British National Bibliography, CiteSeer, DBLP, RISKA Digest, ACM and DBpedia

2.1 Disambiguation and Fusion using IKB

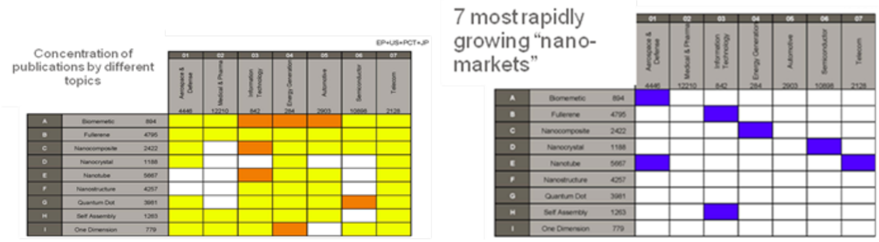
The IKB main purpose is to create an UID for a named entity (NE) [3] and therefore support the process of data cleaning [1] and data scrubbing [6]. Within an IKB an object is automatically populated from the Natural Language Process (NLP) with diverse attributes (e.g. person: gender, first/middle/last name, date of birth, etc.) and relations to other objects [4]. From the list of those attributes a set of keys can be created called Data Driven Identifiers (DDI). Those DDI are used for the creation and merging of UID.

2.2 IKB-based approach for identification

Native attributes of an object as well as a context surrounding it are characteristics enabling the identification of any specific object. A context may be either values of other objects interlinked with the one being identified, or text words surrounding a given object. Therefore, an object can be treated as a point in the multidimensional space of its characteristics [2,5]. To identify an instance from the whole set of object characteristics belonging to a certain class, subsets of identifying characteristics are selected that constitute a kind of “keys” for each instance. Objects obtained from different documents are considered “identical” subject to the “nearness” of one of the sets of identifying characteristics. The IKB process involves the indexing of identifying characteristics of an object. The resulting index is used to search exactly matching or “similar” objects. This information may be adapted and corrected by human interaction. The IKB object is used for automatically identifying objects similarity and hence creates an UID.

3 Use Case

The aim of the use case is to provide to various stakeholders an access to a knowledge portal which is connected to the LOD Russia data sets. Each of the stakeholders has different expectations on how to search for data and on how to analyze the data. For the most part a user will be able to find experts and leaders in a specific domain of their interest, as well as to look for shadow groups of people and institutions working in the domain, based on thesauri and objects of interest. The knowledge portal provides different views to the LOD sets and allows simple filtering using thesauri, sources and filtering by the extracted named entities and semantic relations. Besides the search another use case is related to provide analyzes over the data sets. This process allows for better decision making, especially under the point of view of a patent lawyer or an investment analyst. For example create a report of existing patents related to a domain or on the other hand identify trends of research and link that information to expert groups. Other examples of analysis could be to identify trends in a specific domain (see figure 2 right) or to investigate top publications by numbers in a specific environment (see figure 2 left).



Concentration of publications by different topics

EP+US+PCT+JP

			01 Aerospace & Defense 4446	02 Medical & Pharma 12210	03 Information Technology 842	04 Energy Generation 284	05 Automotive 2903	06 Semiconductor 10698	07 Telecom 2128
A	Biomemetic	894	🟨	🟨	🟧	🟧	🟧	🟨	🟨
B	Fullerene	4795	🟨	🟨	🟨	🟨	🟨	🟨	🟨
C	Nanocomposite	2422	🟨		🟧	🟨	🟨	🟨	🟨
D	Nanocrystal	1188	🟨			🟨			🟨
E	Nanotube	5667			🟧	🟨	🟨		🟨
F	Nanostructure	4257			🟨	🟨	🟨		🟨
G	Quantum Dot	3981	🟨		🟨	🟨	🟨	🟧	🟨
H	Self Assembly	1263	🟨	🟨	🟨	🟨	🟨		🟨
I	One Dimension	779	🟨	🟨	🟨	🟧		🟨	🟨

7 most rapidly growing "nano-markets"

			01 Aerospace & Defense 4446	02 Medical & Pharma 12210	03 Information Technology 842	04 Energy Generation 284	05 Automotive 2903	06 Semiconductor 10698	07 Telecom 2128
A	Biomemetic	894	🟦						
B	Fullerene	4795			🟦				
C	Nanocomposite	2422				🟦			
D	Nanocrystal	1188						🟦	
E	Nanotube	5667	🟦						🟦
F	Nanostructure	4257							
G	Quantum Dot	3981							
H	Self Assembly	1263			🟦				
I	One Dimension	779							

Figure 2 – Left: Top publications by domain; Right: Trend on growing markets

4 Discussion and Outlook

The approach sketched above for the UID creation and merging is a straight forward method comparing DDI Keys. In the remaining time of the project we envisage to enhance the existing approach using a weighting of DDI keys allowing the creation of relevancy of each DDI key.

5 Conclusion

At the end of the research project we expect to have the following impact:

- Better leverage of knowledge from unstructured sources applying new NLP methods such as rule based, statistical and background knowledge;
- Increase information sharing through cross-linking of knowledge using the fusion and data link to other LOD sets;
- Foster various stakeholders through use cases that allow better search and analysis of the data.

6 References

1. Do, H.H. et al., 2000. Data Cleaning : Problems and Current Approaches. *Informatica*, 23(4), pp.1-11.
2. Li, C., Jin, L., Mehrotra, S. Supporting record linkage in large data sets. *Proceeding of eighth International Conference on Database Systems for Advanced Applications (2003)*
3. Lim, E.-P. et al., 1993. Entity Identification in Database Integration R. Hutterer & W. W. Keil, eds. *Information Sciences*, 89(1), pp.294-301.
4. Maynard, D., Li, Y. & Peters, W., 2008. NLP Techniques for Term Extraction and Ontology Population. *Proceeding of the 2008 conference on Ontology Learning and Population Bridging the Gap between Text and Knowledge*, pp.107-127.
5. Tejada, D, Knoblock, C.A., Minton, S. Learning object identification rules for information integration. *Information Systems Journal (2001), pp.607-633*.
6. Widom, J., 1995. Research problems in data warehousing. Proceedings of the fourth international conference on Information and knowledge management CIKM 95, pp.25-30.

Classifying the Wikipedia Articles into the OpenCyc Taxonomy

Aleksander Pohl*

Jagiellonian University
Department of Computational Linguistics
ul. Łojasiewicza 4, 30-348 Kraków, Poland
`aleksander.pohl@uj.edu.pl`

Abstract. This article presents a method of classification of the Wikipedia articles into the taxonomy of OpenCyc. This method utilises several sources of the classification information, namely the Wikipedia category system, the infoboxes attached to the articles, the first sentences of the articles, treated as their definitions and the direct mapping between the articles and the Cyc symbols. The classification decision made using these methods are accommodated using the Cyc built-in inconsistency detection mechanism. The combination of the best classification methods yields 1.47 millions of classified articles and has a manually verified precision above 97%, while the combination of all of them yields 2.2 millions of articles with estimated precision of 93%.

1 Introduction

The primary goal of this paper is a description of a method for a classification of the Wikipedia articles into the OpenCyc taxonomy. This research is motivated by the fact that the proper classification of entities into types is indispensable for any Information Extraction (IE) system (c.f. Moens [11]).

The strength of IE systems versus traditional text processing might be easily illustrated with the Google Trends service[1]. It allows for a comparison of trends for terms that people enter into the Google search engine. Suppose a person wishes to compare two programming languages: *Ruby* and *Python*. If they are entered, a plot concerning them will be presented. But a quick survey of the results will show, that the comparison covers not only the programming languages, but, due to the ambiguity of Ruby and Python terms, also other meanings. What one could expect from such a system would be at least an option to select only the interesting meanings. In a more sophisticated version of the system the selection should be done automatically based on their shared type – that is a *programming language*.

To fulfil such requirements it is required that during the processing of the text, the terms are disambiguated against some reference resource providing

---

* This work is partially sponsored by the Faculty of Management and Social Communication of the Jagiellonian University.

[1] http://www.google.com/trends/

meaning for them. What is more, that resource should also provide fine grained types for the disambiguated terms, to allow for the realization of the second part of the scenario. Although we all know that there is such a resource – namely Wikipedia – and that there exists systems such as DBpedia Spotlight [9], AIDA [19] and Wikipedia Miner [10], that disambiguate unstructured text against it, the types that are determined for the Wikiepdia entities in resources such as DBpedia [8] and YAGO [16] are still not perfect. So the aim of this research is to provide better classification of the entities using the OpenCyc taxonomy as the reference resource.

2 Related work

The DBpedia [1, 2, 8] project concentrates on producing RDF triples[2] representing various facts about the Wikipedia entities, such as their categorisation, date of establishment or birth, nationality, sex, occupation and the like. These data are mostly extracted from Wikipedia infoboxes that describe the facts in a structured manner. It also provides its own ontology [2] used to classify the extracted entities. The classification is achieved via manual mapping of the infoboxes into the corresponding DBpedia ontology classes.

YAGO (Yet Another Great Ontology) [16] in its core is much similar to DBpedia – it converts Wikipedia to a knowledge base that may be queried for various facts using a sophisticated query language. The primary difference between these resources is the reference ontology used to categorise the entities. In the case of DBpedia it is its own hand-crafted, shallow ontology, in the case of YAGO these are WordNet [4] and SUMO (Suggested Upper Merged Ontology) [12].

The classification of Wikipedia entities into WordNet is done via the Wikipedia category system, which helps the Wikipedia users to discover related articles. YAGO exploits this system by syntactically parsing the category names and determining their syntactic heads. If the head is in plural, it is mapped to a corresponding WordNet synset. As a result the entity in question is supposed to be an instance of the concept that is represented by the synset.

A different approach is taken by Sarjant et al. in the experiment described in [15]. At the first stage the authors (following [7]) map the Wikipedia articles into symbols from the Cyc ontology [6] and in the next stage, some of the Wikipedia entities that lack corresponding Cyc symbols are classified into the Cyc taxonomy. The mapping is based on various transformations of the article names as well as transformations of the Cyc symbol names. Then a disambiguation is performed based on the semantic similarity measure described in [18]. In the next stage several heuristics (exploiting information encoded in infoboxs and introductory sentences) are used to determine the classification of the articles. At the last stage, the Cyc inconsistency detection mechanism is used to filter out false positives. The first stage yields 52 thousands of mapped entities, while

---

[2] http://www.w3.org/RDF/

the last 35 thousands of classified entities. As a result approx. 87 thousands of the Wikipedia articles are classified into the taxonomy of Cyc.

3 Current limitations

The short description of DBpedia Spotlight claims that the system is able to recognise 3.5 millions of things and classify them into 320 classes. However, only a half of the Wikipedia articles has an infobox[3] attached and as a result only 1.7 millions of articles are classified withing the DBpedia ontology.

The other thing which is assumed about DBpedia is its perfect classification precision. But this is true only to some extent. E.g. in DBpedia *Algol* is classified both as[4] a *dbpedia-owl:Writer* and a *yago:FlamsteedObjects*. From its description one may find that the entity is a star, but there is a *Writer* infobox in the contents of the article, so the DBpedia classification mechanism assigns a *dbpedia-owl:Writer* class and some other derived classes (such as *foaf:Person* and *dbpedia-owl:Person*).

The DBpedia ontology, with its 320 classes is definitely a small one. Even though a typical IE system defines only a few classes (such as *person*, *organisation*, *place*, etc. cf. [13] for a list of such types), when one wishes to perform moderately-sophisticated IE tasks, such as an automatic cleaning of the extracted data, that ontology is simply too shallow. What is more, the concepts defined in the ontology are not well balanced (e.g. *CelestialBody* has three subclasses: *Planet*, *Asteroid* and *Galaxy* but lacks *Star*).

YAGO seems to be on the opposite end of the ontology spectrum. The conversion of all Wikipedia categories with plural heads into YAGO classes yielded an ontology with 365 thousands of classes[5]. Although this is really an impressive number, most of the classes are over-specified. Consulting the entry for *Gertrude Stein* one will find the following results: *American autobiographers*, *American feminists*, *American poets*, *Feminist writers*, *Jewish American writers*, *Jewish feminists* and more. On the one hand many of the classes in the above example are overlapping, on the other the categories are not decomposed, so searching for *Jews* in YAGO will not yield Gertrude Stein. What is even worse, there is no such category in the ontology[6].

Further investigation into the class system of YAGO will also reveal that the category based classification is also error-prone. Although its authors used some heuristics devised for the removal of the contradicting classifications [3], such contradictions are still present in YAGO. For example *Gertrude Stain* has a type of *Works by Gertrude Stein* and via transitivity of the *type* relation she is classified

---

[3] The facts concerning Wikipedia are obtained using Wikipedia Miner fed with the Wikipedia dump from 22th of July, 2011, containing 3.6M articles. Statistics for the (latest) DBpedia might be different.

[4] http://dbpedia.org/page/Algol – accessed on 25th of July, 2012

[5] http://www.mpi-inf.mpg.de/yago-naga/yago/statistics.html

[6] Probably due to the fact, that in Wikipedia the *Jews* category includes only subcategories.

as *artifact*, *end product* and *oeuvre*. These are definitely wrong classifications. Even if the majority of the classifications are correct, such inconsistencies should be totally removed, since they introduce contradicting facts into the knowledge base.

To sum up – the available knowledge sources, that classify Wikipedia articles into ontologies still lack some features required from a fully-fledged IE systems. The classification of Wikipedia articles into Cyc was very limited, while the classification provided by DBpedia and YAGO could still be improved.

4 Solution

The proposed solution follows [15] – namely the goal is to classify as many of the Wikipedia articles into the Cyc taxonomy [6] and then use its inconsistency detection mechanism to filter out inconsistent classifications. The primary difference between them is that the first method covers less than 100 thousands of the Wikipedia articles, while the method presented in this paper yields more than 2 millions of classified entities.

The important feature of Cyc (compared to other ontologies like YAGO and DBpedia) is its efficient inferencing engine, which allows for querying the ontology for various sophisticated facts. This makes the development of any Cyc-based system simpler, since there are many built-in API calls, covering navigation through the taxonomy, indexing and inferencing, that would have to be otherwise implemented from scratch.

The contents of Cyc might be roughly divided into two ontological categories: *collections* and *individuals*. The entities from the second category might be instances of entities of the first category and might not have their own instances. They roughly correspond to the entities which are referred to by their proper names. On the other hand the entities of the first category have instances, but might be also instances of other collections. It might be assumed that the first order collections (whose instances are only individuals) correspond to classes (such as *books*, *people*, *numbers* etc.). In these terms Cyc contains approx. 71 thousands of classes[7].

The difference between the DBpedia ontology and Cyc is rather obvious – there are simply more classes (and relations) available. The difference between Cyc and YAGO is more subtle. Cyc also uses reification to a large extent, the feature that was criticised in YAGO, but the reification level is much lower in Cyc, than in YAGO, so none of the classes found in YAGO that describe *Gertrude Stein* will be found in Cyc. As a result the classification might be expressed in canonical form, where each component of the classification type is separated.

But that what makes Cyc particularly helpful for the classification task is the *disjointWith* relation with the corresponding *collections-disjoint?* API call[8]. The information encoded using this relation allows for straightforward detection of inconsistencies in object classification. Assuming that given object is classified

---

[7] The statistics are provided for OpenCyc version 4.0, released in June 2012.

[8] The description of the Cyc API is available at `http://opencyc.org/doc/opencycapi`

into several Cyc collections, by calling *any-disjoint-collection-pair* one can check if that classification is consistent. In the case that one of the classes is more trusted, the other classes might be accepted and rejected pair-wise.

5 Classification algorithm

5.1 Introduction

The goal of the classification algorithm is a consistent assignment of one or more first order Cyc collections to every Wikipedia article. The nature of the classification depends on the ontological status of the entity described in the article – if it is an object, the classification is interpreted as *instanceOf* relation, if it is a concept, the classification is interpreted as *subclassOf* relation. Telling apart objects from concepts is out of scope of this algorithm.

Unlike the other algorithms that were used to classify Wikipedia articles into an ontology using only one method, this algorithm tries to maximise its coverage by combining several classification methods: *category*-based, *infobox*-based, *definition*-based and *mapping*-based. It is assumed that the results of the methods will overlap, allowing for a reconciliation of the results using both the well developed Cyc taxonomy and the inconsistency detection mechanism.

5.2 Categories

The primary source of classification data is the category system of Wikipedia. The category names are split into segments and the first plural noun is detected. That noun, together with its preceding modifiers (if they exists) is assumed to be the name of a parent *semantic* category (i.e. a category that subsumes the category in question) of the category. Then the ancestor categories of the category are consulted and the category with the same name (if it exists) is selected as a parent semantic category.

Although inspired by YAGO classification algorithm, this method diverges from it in several places. First of all the name is not parsed using a link-grammar parser. The second difference is the more sophisticated semantic parent determination algorithm. It stems from the fact that the single-segment expressions used in YAGO are more ambiguous than the multi-segment expressions. The last difference concerns the inspection of parent/ancestor categories. Although not yet realized, in future this will allow for an extension of the Cyc ontology with meaningful categories defined in Wikipedia.

When the set of root semantic categories is determined, these categories are mapped semi-automatically into Cyc symbols. The name of the category is converted into singular form and then the methods of Wikipedia-Cyc names mapping described in [7] are applied. Usually this will lead to an ambiguous mappings. The Cyc symbols that are not first order collections are filtered. Still in most of the cases there is more than one candidate mapping. Although it is possible to create a method of automatic selection of the best candidate, since this mapping is the key element of the classification algorithm, the proper mapping is determined manually. This also allows for ignoring mappings that are valid but not meaningful – e.g. words such as *group*, *system* or *collection* have very broad meaning in Cyc and they are not mapped.

5.3 Infoboxes

The second source of article classification were the infoboxes. The author used the classification provided by DBpedia and the mapping between Cyc and DBpedia ontology that is available in the Cyc Semantic Web service[9]. It turned out that many of the DBpedia classes were not mapped into Cyc symbols, so the author manually mapped the remaining classes.

This classification procedure was augmented with a category-based heuristic used to identify people. All the articles that were lacking an infobox, but belonged to the *Living people* category or categories ending with *births* or *deaths* were classified as *Person*. This simple heuristic gave 500 thousands of classifications with confidence comparable to the original infobox method.

The infobox and people-related category classification heuristics have very high level of confidence, since the infoboxes, the categories and the infobox-class mappings are determined manually and the chance for a misclassification is low. This is the reason why the mapping between the Cyc symbols and Wikipedia categories was first tested against the results of this method.

5.4 Definitions

The third method that was used to classify the Wikipedia articles was inspired by methods used to extract *hyponymy* relation from machine-readable dictionaries. Following Aristotle, the definitions in such dictionaries are constructed by indication of *genus proximum* and *differentia specifica*, that is the closest type and the specific feature of the defined entity. This allows for a construction of patterns devised for the extraction of the type of the entity (cf. [5], [15]).

The method used to determine the location of the entitie's type is as follows: the first sentence of the short description of the article that is extracted using the DBpedia extraction framework is tagged using Stanford POS tagger [17]. Then a continuous sequence of adjectives, nouns, determiners and (optional) *of* preposition that follow the first occurrence of *to be* or *to refer* verb is marked as the probable location of the type name. This expression is disambiguated using the improved Wikipedia Miner disambiguation algorithm [14], taking as the disambiguation context all the articles that are linked from the source article.

This method does not follow [15] in using the existing links that are usually present in the first sentence of the definition, since first, there are many articles which lack a link to the article's type in the first sentence and second, the links not always indicate their type (e.g. only the type constituens like *life* and *system* in the *living system* type).

---

[9] http://sw.opencyc.org

After defining the type-articles, the articles which are not semantically related to any other article with the same type (i.e. their semantic relatedness measure [18] with each article is 0) and lacking Wikipedia categories that include the type name in their names are rejected as false entity-type mappings.

At the end of the procedure the type-articles are mapped to Cyc symbols. In the first step candidate Cyc symbols are generated with the *dentotation-mapper* Cyc API call. This call maps given string to all its interpretations in Cyc. It is called for the name of the article and if it does not succeed the names of the links that have the article as their target are used, in descending frequency order. Only the symbols that are first order collections are registered as candidates.

In the next step the articles that have a Cyc type assigned via the *infobox* classification method are used to order the candidate type-article mappings. In the first pass the equality and in the second pass the subsumption tests are performed. The symbol with the largest number of positive matches is selected.

As the last resort the mapping between the type and the Cyc symbols was determined on the basis of the generality of the Cyc collection (determined as the sum of subsumed collections and covered instances). If a collection was proposed for any of the type names, the most general was selected. If there was no such mapping, but for the covered articles there were any *infobox*-based collections determined, their most specific generalisation was selected.

As a final remark it should be noted that the definition-based classification was applied only to the articles that were not classified as *Person* in the infobox-based classification.

5.5 Cyc mappings

The Cyc-mapping based classification utilizes the direct mappings between Wikipedia articles and Cyc symbols obtained with the methods described in [15] (excluding the cross-validation step, which is performed using the category-based classification). The mapping assumes various types of transformations of the names of Cyc symbols and Wikipedia articles as well as disambiguation strategies. The author used the original results of Sarjant et al. so the reader is advised to consult [15] in order to check the details of the method.

5.6 Cross-validation

The cross-validation of the results generated by the different classification methods allows for a consistent assignment of the types to the articles. It assumes that they have different accuracy and it takes into account the fact, that the number of classified articles varies between the methods. What is more, the results obtained with the more accurate methods are reused by the weaker methods. The methods are cross-validated pair-wise and their order is as follows:

1. categories vs. infoboxes
2. categories vs. definitions
3. categories vs. Cyc mappings

In the first case it is assumed, that the infobox-based classification is more accurate than the category-based one. In the two remaining cases this assumption is inverted. The structure of the cross-validation is as follows:

1. selection of the Cyc symbol that is assigned as the type by the second method (i.e. not category-based) to the Wikipedia article; this is the *primary* Cyc type
2. selection of Cyc symbols assigned to the article by the category-based classification, which are *compatible* with the primary type
3. generalisation of the symbols that were compatible with the primary type; this is the *secondary* Cyc type
4. *compatibility-check* between the secondary type and the Cyc symbols that are assigned to the categories of the article

The selection of the primary Cyc type is usually straightforward – it is the type that was assigned by the method. If there are many such types, the first type of the most specific types is taken. Even though in some cases this leads to a lose of information, the problem is reduced by the generalisation step (3) and usually the category-based classification spans more types than the alternative methods.

The compatibility of the symbols in the second step is determined using subsumption and instantiation relations, via *genls?* and *isa?* Cyc API calls[10]. In the case of the subsumption relation the types are marked as compatible disregarding the fact which of the types is the subsumed and which is the subsuming.

The generalisation of the types that are compatible with the primary type is performed using the *min-ceiling-cols* Cyc API call, which computes the most specific generalisation of a set of collections. The results are filtered using a black-list of types such as *SolidTangibleThing* and *FunctionalSystem* that are too abstract for this task. The black list is created empirically to forbid generalisations that do not posses discriminative power.

The goal of the fourth step is to select the types that will be assigned to the article. This is performed using both the subsumption relation and the disjointness relation. If the category-based type is subsumed or subsumes the secondary type, it is marked as *compatible*. If it is disjoint with the type, it is marked as *incompatible*. Still is status might be *undetermined* if none of the situations occurred.

The side effect of the cross-validation of individual entities is validation of the mappings between the Cyc symbols and the Wikipedia categories. Although the mapping of the root categories was manual, the mapping of the other categories was automatic, thus it introduced errors. Thanks to the cross-validation such erroneous mappings were removed and not exploited in the next cross-validation scenario. Furthermore, the mappings that turned out to be positively verified were used as a sole source of classification for the entities that did not have any types assigned in any of the cross-validation scenarios.

---

[10] The second call is used only for direct Cyc mappings, since in all other cases the types are always collections.

6 Results

Each variant of the cross-validation procedure yielded a different number of types that were determined as compatible and incompatible for the respective concepts. Table 1 summarizes these numbers. The total number of concepts is the number of concepts for which given classification method assigned at least one type. The number of cross-validated concepts is the number of concepts that have the type determined by the method and at least one category-based type[11]. The classifications denoted as *valid*, were the classifications for which the cross-validation procedure found at least one compatible type and as *invalid* – the classifications that have only incompatible types determined.

The last column indicates the number of valid classifications that were produced by the method for concepts that were not classified by the previous methods. It also indicates the number of classified concepts that were incorporated in the final result.

Table 1. The number of classifications (in thousands) with the respective status produced by each variant of the cross-validation procedure. C_t – total number of classified concepts. C_c – number of classifications that were cross-validated. C_v – number of valid classifications. C_i – number of invalid classifications. Δ – number of classifications included in the final result.

Variant	C_t	C_c	C_v	C_i	Δ
Infoboxes	2188	1712	1471	67	**1471**
Definitions	406	247	154	60	**154**
Cyc mappings	35	25	14	5	**3**
Categories	2470	742	593	—	**593**
Total					**2221**

The results of the classification were verified by two subjects (excluding the author of the article) with some ontological and linguistic training (one being a PhD student of philosophy and the other a person with a bachelor degree in linguistics). Each variant was verified on a distinct set of 250 randomly selected cross-validated classifications with equal number of compatible and incompatible types. The subjects were presented with the names of the entities and their respective types, supplemented with their short descriptions – the first paragraph of the Wikipedia article in the case of the entities and the comment attached to the Cyc symbol in the case of the types.

The subjects had three answers to choose from when deciding if the classification is correct: *yes*, *no* and *not sure*. The third option was left for cases when it was hard to decide if the classification is correct, due to the mismatch of description accuracy level between Wikipedia and Cyc.

---

[11] In the case of category-only classification, these were the types that were recognized as valid in previous cross-validation scenarios

As a result the precision and recall measures are given separately for cases when both of the subject were confident about their choice and only one of them was confident. In the first case the answer was used to compute the precision and the recall only if both answers were the same, that is there was no adjudication procedure implemented. The precision and the recall were defined as follows:

$$P = \frac{c_{tp}}{(c_{tp} + c_{fp})} \qquad R = \frac{c_{tp}}{(c_{tp} + c_{fn})}$$

where:

- c_{tp} – the number of types determined as *compatible* by the cross-validation procedure and marked as *valid* by *both* of the subjects
- c_{fp} – the number of types determined as *compatible* by the cross-validation procedure and marked as *invalid* by both of the subjects
- c_{fn} – the number of types determined as *incompatible* by the cross-validation procedure and marked as *valid* by both of the subjects

Table 2. The results of the verification of the cross-validated classifications carried out by two subjects on 250 classifications (for each of the cross-validation variants). P – precision for classifications with agreed answer. R – recall for classification with agreed answer. $P_{1/2}$ – precision for classifications with one uncertain answer. $R_{1/2}$ – recall for classifications with one uncertain answer. A – agreement between the subjects. $C_{1/2}$ – percentage of classifications that were confusing for one of the subjects. C – percentage of classifications that were confusing for both of the subjects. # – number of classified concepts (in thousands).

Variant	P	R	$P_{1/2}$	$R_{1/2}$	A	$C_{1/2}$	C	#
Infoboxes	**97.8**	77.2	90.0	78.0	**92.5**	9.7	2.1	**1471**
Definitions	93.5	69.4	**93.9**	68.6	89.0	**5.2**	**0.0**	154
Cyc mappings	94.0	76.4	89.1	71.5	86.1	10.8	**0.0**	3
Categories	81.9	**80.4**	82.1	**78.7**	90.5	10.9	0.8	593
Overall (est.)	**93.3**	**77.5**	**88.2**	**77.5**	**91.7**	**9.7**	**1.6**	**2221**

The results of the verification are presented in Table 2. The testers agreed approx. in 90% of the answers, which means that the verification procedure was meaningful. In approx. 10% of the answers one of the subjects was confused with the classification. It shows that the ontology-based classification is not an easy task, especially if the reference resource is Cyc, making very strict and well defined distinctions, which are sometimes hard to accommodate with fuzzily defined Wikipedia entities.

Comparing the results of the classification to YAGO[12] shows that the combination of the best methods has almost the same precision as in YAGO. However,

---

[12] http://www.mpi-inf.mpg.de/yago-naga/yago/statistics.html

it should be noted that in the case of the presented algorithm there should be no inconsistent classifications and no compound types, which are both present in YAGO. What is more, Cyc collections are defined more strictly than WordNet synsets.

Comparing the results to DBpedia shows that with a moderately hight precision (93%) we can assign types to more than 2.2 millions of entities, going far beyond the infobox-based classification.

The comparison with the results of Sarjant et al. [15] is harder, since the evaluation procedure was more sophisticated in the second case. However, they reported that the classification was indicated as strictly correct by the majority of evaluators in 91% of the cases. Assuming this is a fair comparison, the presented method surpasses their results both in precision (93% vs. 91%) and coverage (2.2 millions of classified concepts vs. 87 thousands).

7 Conclusions

The precision of the Cyc-based method used to classify the Wikipedia articles depends strongly on the source of the classification information. It is apparent that it is possible to achieve very good classification results (with precision above 97%) for a large number (1.47 millions) of articles using the best method (infobox-based) and also, that with a moderately high precision (93%) we can extend the coverage of the classification.

The sample results of the classification together with the handcrafted mappings are available on the Internet: `https://github.com/apohllo/cyc-wikipedia`. The full result is available upon request. The results of the classification are incorporated into an Information Extraction system, that utilizes the improved Wikipedia Miner algorithm [14]. This system is available at `http://textplainer.com`.

As a final remark we can conclude that Cyc is well suited for the task of detecting the inconsistencies in the classification. The author is going to further utilize this feature in cross-linguistic classification of the Wikipedia articles and automatic, type-base validation of the information extraction results.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A nucleus for a web of open data. The Semantic Web pp. 722–735 (2007)
2. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia-A crystallization point for the Web of Data. Web Semantics: Science, Services and Agents on the World Wide Web 7(3), 154–165 (2009)
3. De Melo, G., Suchanek, F., Pease, A.: Integrating yago into the suggested upper merged ontology. In: Tools with Artificial Intelligence, 2008. ICTAI'08. 20th IEEE International Conference on. vol. 1, pp. 190–193. IEEE (2008)
4. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press (1998)

5. Kazama, J., Torisawa, K.: Exploiting wikipedia as external knowledge for named entity recognition. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL). pp. 698–707 (2007)
6. Lenat, D.B.: CYC: A large-scale investment in knowledge infrastructure. Communications of the ACM 38(11), 33–38 (1995)
7. Medelyan, O., Legg, C.: Integrating cyc and wikipedia: Folksonomy meets rigorously defined common-sense. In: Proceedings of the WIKI-AI: Wikipedia and AI Workshop at the AAAI. vol. 8 (2008)
8. Mendes, P., Jakob, M., Bizer, C.: Dbpedia for nlp: A multilingual cross-domain knowledge base. LREC-to appear (2012)
9. Mendes, P., Jakob, M., García-Silva, A., Bizer, C.: DBpedia Spotlight: shedding light on the web of documents. In: Proceedings of the 7th International Conference on Semantic Systems. pp. 1–8. ACM (2011)
10. Milne, D., Witten, I.: Learning to link with Wikipedia. In: Proceeding of the 17th ACM conference on Information and knowledge management. pp. 509–518. ACM (2008)
11. Moens, M.: Information extraction: algorithms and prospects in a retrieval context, vol. 21. Springer-Verlag New York Inc (2006)
12. Niles, I., Pease, A.: Towards a standard upper ontology. In: Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001. pp. 2–9. ACM (2001)
13. NIST: Automatic Content Extraction 2008 Evaluation Plan (ACE08) (2008), `http://www.itl.nist.gov/iad/mig/tests/ace/2008/doc/ace08-evalplan.v1.2d.pdf`
14. Pohl, A.: Improving the Wikipedia Miner Word Sense Disambiguation Algorithm. In: Proceedings of Federated Conference on Computer Science and Information Systems 2012. IEEE (to appear)
15. Sarjant, S., Legg, C., Robinson, M., Medelyan, O.: All you can eat ontology-building: Feeding wikipedia to cyc. In: Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 01. pp. 341–348. IEEE Computer Society (2009)
16. Suchanek, F., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proceedings of the 16th international conference on World Wide Web. pp. 697–706. ACM (2007)
17. Toutanova, K., Manning, C.: Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In: Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13. pp. 63–70. Association for Computational Linguistics (2000)
18. Witten, I., Milne, D.: An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In: Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI Press, Chicago, USA. pp. 25–30 (2008)
19. Yosef, M., Hoffart, J., Bordino, I., Spaniol, M., Weikum, G.: Aida: An online tool for accurate disambiguation of named entities in text and tables. Proceedings of the VLDB Endowment 4(12) (2011)

Finding Good URLs: Aligning Entities in Knowledge Bases with Public Web Document Representations

Christian Hachenberg and Thomas Gottron

Institute for Web Science and Technologies (WeST), University of Koblenz-Landau,
{`hachenberg,gottron`}`@uni-koblenz.de`

Abstract. In this paper we address the novel task of mapping entities from a knowledge base to public web documents. This task is of relevance for aligning structured data with web documents, e.g., for the purpose of providing equivalent human readable representations of entities or to detect and propagate changes on the web to the knowledge base. An alternative interpretation of the task is to find good public URLs for the entities in a knowledge base. In order to address the task, we adapt and investigate several approaches based on web search and link network analysis. We compare nine approaches including ordinary web search for the text label of an entity as well as link analysis strategies like HITS authority ranking or PageRank. We evaluate the approaches under the aspect of identifying URLs of documents which are good representations of a given entity. In general, our experiments show a significant advantage of label based web search over all other methods. Furthermore, we introduce a filtering technique leveraging semantic typings to boost the performance of virtually all methods.

1 Introduction

A knowledge base can be seen as a database where information is organized to be available for standardized access, retrieval or querying. One common approach to model knowledge bases are ontologies describing different types of objects, their corresponding instances (entities) and the various ways they are linked to each other (e.g. hierarchies, taxonomies or other semantic relations). In this paper we address the task of establishing a mapping from entities in a knowledge base to public web representations of these entities. These representations correspond to web documents and are identified by an URL. Hence, we seek a mapping from entities in a knowledge base to documents on the Web, i.e. providing URLs of web documents best representing a given entity.

There are several scenarios in which such a mapping is of relevance. One use case is to utilize the URLs of web documents as URIs for a direct public representation of an entity. This would enable to publish a proprietary knowledge base in a semantic web format. A second application is to render a knowledge base more accessible for human users. Here, the mapping from entities to web documents can be used as a human readable overlay for browsing knowledge bases and their entity descriptions. Finally, some of the information in a knowledge base, such as the type or properties of entities as well as links between entities, might become obsolete over time. While the task of updating the knowledge base can be pursued manually by an expert, this process becomes infeasible when the rate of change is high and/or the size of the knowledge base is large. Information extraction techniques which use a mapping from entities to

URLs can provide a solution here. They can operate on the web documents assigned to an entity, detect changes and propagate them back to the knowledge base.

Mapping and aligning text or web data to knowledge bases is a well-established field of research [17, 21]. Here, typical scenarios are the generation, extension or population of knowledge bases from unstructured or semi-structured data. We are interested in the opposite direction, though, of mapping entities from knowledge bases to the Web. To our best knowledge there is no work in this direction so far.

In this paper we investigate several approaches for finding mappings from knowledge base entities to public web documents. The approaches can be divided into three categories: keyword based web search using descriptive texts of entities, approaches making use of the link structure among web documents corresponding to the connections between related entities and a post-process filtering approach leveraging semantic typings. We evaluate the approaches regarding their effectiveness in identifying web documents that perfectly match the entities in a knowledge base. To this end, we first have constructed a test collection of 100 entities of different types and varying degree of being connected to other entities. Then, we evaluated for all approaches the quality of identified documents. In this way, we could identify the most effective methods and observed that especially the filtering based on semantic typings helps boosting virtually all methods.

The rest of the paper is structured as follows: we start with a formal definition of the task of finding public URLs to represent entities in a knowledge base in Section 2. We then present a collection of approaches to solve this task in Section 3. In Section 4 we develop an evaluation methodology and analyse the performance of the different approaches. Finally, after giving an overview of related work in Section 5 we discuss our results and conclude with an outlook on future work.

2 Task Definition

Given that we address a novel task, we start by providing a formalization of the task of finding good URL representations for entities. We also provide a short example to illustrate the setting.

2.1 Formal Definition of the Task

The task of finding good web documents representations for knowledge base entities can be formalized as finding a mapping between the entities in a knowledge base and URLs on the Web. Thus, the task is operating on two structures: a knowledge base and the Web as a hyperlink graph of documents.

Knowledge Base: We represent a knowledge base as a graph in which the entities form nodes and are connected by different types of edges. So, a knowledge base K is a tuple (E, C, L, P), where:

- E is a finite set of entities $E = \{e_1, \ldots, e_n\}$.
- $C \subset E$ is a finite set of types or classes $C = \{c_1, \ldots, c_l\}$.
- Λ is a finite set of literals $\Lambda = \{\lambda_1, \ldots, \lambda_m\}$.

- P is a set of properties $P = \{P_i\}$, I being a finite index set. Each property is a binary relation linking entities with other entities or literals: $P_i \subseteq E \times (E \cup \Lambda)$
- P contains a specific property $P_c \subseteq E \times C$ assigning semantic types to the entities.

We further assume that one of the properties linking entities to literals is used to attach *labels* to entities. This property provides a name or short description and we use the shorthand notation e_i*.label* to denote the literal attached to e_i via this property.

Web: We model the Web also as a graph, consisting of a set of documents represented by URLs and the hyperlink structure between these documents:

- H is a set of web documents $H = \{h_1, \ldots, h_k\}$. Each document can be represented by its URL.
- L is a binary relation representing hyperlinks between web documents $L \subseteq H \times H$

For the sake of completeness it remains to be said that each web document involves some content $c(h_i)$. Indirectly, we make use of this context for a keyword based search.

Mapping M_{web}*:* The task of finding good web documents representations for knowledge base entities can formally be seen as the task of finding a mapping from the entities E in the knowledge base to web documents H represented by URLs. This mapping can be defined by $M_{\text{web}} \subseteq E \times H$.

This definition provides a syntactic formalization. In order to fulfil the need for finding *good* URLs, the mapping M_{web} is required to map an entity to a web document which is a representation of the very same entity. This means the web document shows a clear and preferably complete or extensive embodiment of the entity. As there might be many representations of an entity on the Web, there might accordingly be several solutions for M_{web}.

2.2 Example

Assume a knowledge base about movies, actors and directors. For instance, now consider an entity representing the 1995 movie *Rob Roy* starring Liam Neeson and Jessica Lange which was directed by Michael Caton-Jones. A mapping M_{web} should assign this entity onto web documents that represent this movie. Suitable representation might cover the Wikipedia article about the movie, its IMDB entry or an official website of the movie itself. A review of the movie would not be suitable as the document rather represents a discussion about the movie than a manifestation or representation of it. Neither would a webpage of an online shop offering the movie for sale on DVD be suitable, since the document represents a DVD containing the movie. Obviously, neither a web document discussing the historic figure of Rob Roy nor one representing the novel by Sir Walter Scott would be good representations.

3 Approaches

We now present a total of nine different approaches for solving the task described above as well as a solution for post-process data source filtering. The approaches are built on top of each other and can be categorized into three types. The baseline approaches in

Section 3.1 make use of text labels attached to the entities which are used as queries for standard web search engines. The ranked lists returned by web search engines form the basis for further approaches in Section 3.2 which aim at optimizing the results by using the underlying link structure aligned to the context of an entity. A last category of approaches in Section 3.3 makes further use of the types of entities and benefits from different search results over the same type of entity. It actually makes use of and is itself applicable to the approaches in the previous two categories.

3.1 Keyword Based Web Search

The naive way for mapping entities to web documents is to use the entities' label as input for a web search. We considered two implementations for this approach.

Label Search It is a simple web search using the label $e_i.label$ of an entity e_i as query terms. If an entity has more than one label attached, a concatenation of the labels can serve as query. This approach entirely ignores the structure of the knowledge base as well as the context of an entity. For the sake of clarity, we consider this as pure and most intuitive way of search for a mapping of an entity to the Web.

All Linked Labels In order to extend the search and to use the context of an entity e_i, we consider all entities that are connected to it in the knowledge base. This means we extend the keywords used for web search of e_i by the labels of the entity set $E_j := \{e_j | (e_i, e_j) \in P_i \vee (e_j, e_i) \in P_i\}$. We denote the joint set $E_j \cup \{e_i\}$ as E_{c_i} in the following, the set which comprises the entity under investigation as well as all its connected entities. This is to evaluate the impact of using the graph structure in the knowledge base and so we further refer to this method as *All Linked Labels*.

3.2 Search Making Use of the Link Structure

The hyperlink structure on the Web represents relations between web documents. We leverage these (typically content motivated) relations as well as the semantic relations in the knowledge base in order to compute the mapping M_{web}. For this reason, we use again the set of connected entities E_{c_i} we introduced in the *All Linked Labels* approach. But, instead of formulating a single query we rather generate one query for each entity. As done for *Label Search* each entity's label is used to query a web search engine to obtain a ranked list of webpages. We keep track of which documents (and their URLs) were returned for which entity. As we retrieve several pages for each entity, the document collection obtained in this way will be by far larger than the original set of entities E_{c_i}. Subsequently, each webpage's contents is analysed for web links to other webpages. We then create an adjacency matrix for the hyperlink network of the web document collection. This provides us with a graph structure as depicted in Figure 1.

Using *SearchEngine*$(e_k.label)$ as function to provide us with the documents found when searching for the label of entity e_k, the approach can be formalized as follows:

1. Search for labels of all resources linked to e_k on the Web to obtain a collection of documents:
   $H_{e_i} := \{h_k | h_k \in H \wedge e_k \in E_{c_i} \wedge h_k \in \textit{SearchEngine}(e_k.label)\}$

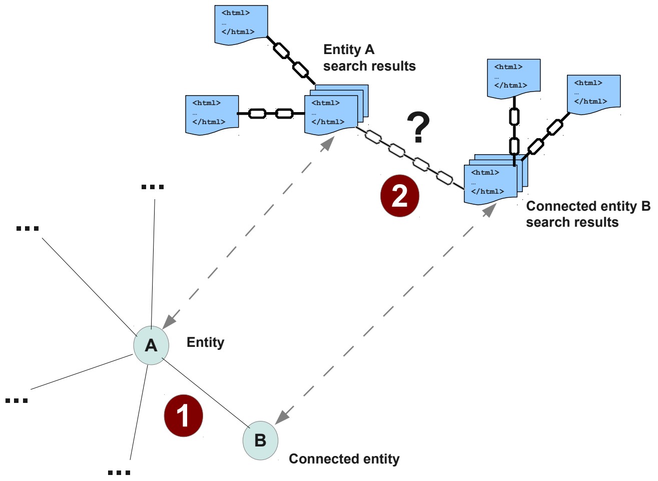


Fig. 1. Sketch of a typical link network originating from entity A with corresponding entities being connected in the knowledge base (e.g. entity B). **(1)** Entity A is connected to another entity B in the knowledge base. Their labels are used to find a result set of URLs via a web search engine **(2)** If there exists a HTML link from one URL (inside the webpage), e.g. coming from entity A web search, to an URL coming from entity B web search we keep this URL in our link network as node. Edges are represented by the existing HTML link.

2. Afterwards, for entity e_i: use all links present in documents H_{e_i} (denoted L_{e_i}) linking in between any documents $\in H_{e_i}$ to create a link network.

The hyperlink network we obtain in this way then serves as input for the approaches discussed in this section. All the approaches analyse this network for computation of a ranking of the documents. The aim is to rank higher those documents which are a better representation of the initially considered entity e_i.

PageRank We apply the original PageRank method [7] by Brin and Page with parameters α = 0.85 taken from the literature, e.g. [16] (and ϵ = 10^{-8} used with the power method for computation of G):

$$G = \alpha S + (1-\alpha)\frac{1}{n}ee^T = \alpha H + (\alpha a + (1-\alpha)e)\frac{1}{n}e^T \tag{1}$$

S is the stochastic matrix coming from normalizing the hyperlink matrix H so that it fulfils the stochastic property for a matrix, e is the unit vector, a the “dangling node”[1] vector having $a_i = 1$ if page a_i is a dangling node and 0 otherwise. In our case, the hyperlink matrix H stems directly from the link network L_{e_i} (for an entity e_i) which

---

[1] “Dangling” means a node is only accessible from other nodes and there is no way out to continue to other nodes again according to the “random surfer” model used in the PageRank algorithm.

is stored first as an adjacency matrix and where each entry is normalized afterwards in order to fulfil the constraints of H.

Topic PageRank We introduce also a modified version of PageRank where we change the first part of the convex combination from $\alpha\ S$ to $\alpha\ H$ and the second part for the "random surfer" from $(1-\alpha)\ \frac{1}{n}\ ee^T$ to $(1-\alpha)\ \frac{1}{n}\ V$. V is a personalization matrix according to the *Label Search* so that all entries in V representing links to URLs from the search engine result list of the corresponding entity are set to 1 and 0 otherwise.

Focussed PageRank All webpages from the web search results for the considered entity are looked up in the PageRank list as computed with the original method. Thus, we only consider pages retrieved for the entity's label and return them in descending order by their individual PageRank score. In this way, we get a relatively (re)ordered list of *Label Search* results according to the position in the complete link network ordered by *PageRank*.

HITS This covers the original method [14] by Kleinberg where only inbound links to a webpage are considered for ranking (authority ranking):

$$x^{(k)} = L^T y^{(k-1)} \tag{2}$$

HITS is computed using the iterative power method so that $x^{(l)}$ denotes the authority vector in iteration l we are interested in whereas $y^{(k-1)}$ is the hub vector from the previous iteration and L is the adjacency matrix (as for PageRank, $\epsilon = 10^{-8}$ is used for computation). The matrix L directly corresponds to our adjacency matrix L_{e_i} (for an entity e_i) from the link network.

Topic HITS Only authorities which are among the results of the *Label Search* result list are considered. Actually, the adjacency matrix L is changed so that all entries are set to zero which do not belong to one of the results from the *Label Search* method. This means in effect, all links are discarded which do not point from any webpage in the link network to one of the webpages in the list of the *Label Search* method.

Focussed HITS This works exactly like the *Focussed PageRank* method but uses *HITS* ranking instead of *PageRank*.

Focussed Link Count Along the lines of *Topic HITS* ranking, we simply count the number of inbound links for every webpage in the *Label Search* result list. The web documents are then ranked by decreasing number of incoming links.

3.3 Data Source Filtering Using Semantic Typing

The last category of approaches makes use of the semantic typing of entities in the knowledge base. The hypothesis for this approach is that entities of the same type are typically found together at the same location on the Web. Therefore, by querying the web for several entities of the same type we can observe web sources ranking repeatedly high for this type. Such knowledge can be used to filter results sets by removing web documents from the result list which did not appear repeatedly.

To this end, we implemented a variation of the method for Borda count result set fusion [1]. Instead of merging result lists of the same query from several search engines, we merge web sources in result sets of several queries from the same search engine. This means that we consider in the result set only the domain name in the URLs to represent a web data source. In a next step we generate a joint ranking of the data sources over several queries (i.e. entities) of the same semantic type. Finally, we take the top ranking data sources as a filter to apply to each individual result list. That means only data sources (i.e. domain names) accounting for at least 1% of the total sum of Borda counts per type are taken into consideration. Note, that this process is independent of the initial computation of the result list. It is a post-processing step that can be applied to all approaches we mentioned before.

4 Experiments and Evaluation

In order to compare the methods described above, we evaluated them in real world scenarios. The evaluation methodology follows the paradigms widely used in the information retrieval domain, as we are effectively dealing with a search task.

We utilize a selection of entities from DBPedia as knowledge base and use the above mentioned algorithms for retrieval of good web representations of those entities. In Section 4.1 we elaborate the details of how we chose these entities to have an unbiased evaluation data set. As web search engine we used BING[2] as it offers an unrestricted use via API calls. In general, the use of an external web search engine bears the risk of an uncontrolled bias in the data. However, given the lack of a controlled search index over the Web, this risk is equally immanent to all search engines. The search results for each label are cut at 50 results (i.e. 50 webpages). We ran all the approaches introduced in the previous section with these parameters and computed a ranking of good web document representations for the entities hereof. The resulting sets of URLs from each method were pooled and presented for graded relevance judgement to expert evaluators.

4.1 Selecting Entities for Evaluation

We used four domains of general purpose among datasets in the knowledge base DBPedia: Those are of type *company*, *city*, *movie* and *person*[3]. Per type (i.e. domain), we selected 25 entities hence 100 entities in total. All entities provided a label via a single `rdfs:label` property. In order to preserve the underlying distribution of entities mentioned frequently or rarely on the Web we first drew uniformly 1000 entities out of each domain. We stratified these 1000 entities into bins according to the $n^{th} tertile$ of the number of results for a certain entity. This number is generally returned by BING web search engine when the entity's corresponding label is put into, respectively. In the

---

[2] `http://www.bing.com`, search parameters are set to allow for only English web documents with all sorts of content filtering being deactivated

[3] `http://dbpedia.org/ontology/Company`, `http://dbpedia.org/ontology/City`, `http://schema.org/Movie` and `http://dbpedia.org/ontology/Person`

following, we then drew randomly the 25 entities per domain (100 in total) where 9 entities came from the first tertile, and 8 each from second and third tertiles. After having selected the entities, we extended our custom knowledge base by all connected entities, i.e. computing a 1-hop closure over the properties of each entity (see also Figure 1). The actual numbers ranged from 10 to over 3000 connections per entity.

4.2 Construction and Evaluation of the Web Document Collection

We fed our knowledge base into each of the methods above and computed the top 50 rankings for every entity[4]. In order to evaluate the results it was necessary to have human judgements, whether the found web documents actually were suitable representations of the entities. To this end, we applied pooling of result lists for each query, by taking only the top 5 (i.e. highest ranked) URLs of webpages of each of the analysis methods. We presented these web documents to human evaluators and asked for relevance judgements. To support the relevance decision we provided the evaluators with the entity's label, a short description taken from the `rdfs:comment` property in our knowledge base and a screen shot of the web document to ensure a consistent presentation of the documents to the evaluators.

The human experts were asked to judge each single document with respect to its degree of relevance [11] denoted by 0 (irrelevant), 1 (marginally relevant), 2 (fairly relevant) or 3 (highly relevant). The experts were given specific instructions to judge a document as highly relevant, if and only if it is solely about the entity and shows a clear and preferably complete or extensive embodiment of this entity.

Since we address a novel task in this evaluation we checked the agreement among the human evaluators. For this reason, we had each document judged also by a second evaluator. As the evaluators had to assign a document to one of the four possible categories on an ordinal scale we used Krippendorff's Alpha [15]. Table 1 shows the results of this analysis both in total and for each type of the entities. All values are above 0.667 which is considered the minimum threshold for a reasonable agreement [15] both for single domains and the total of all entities. Hence, the obtained relevance judgements are consistent and valid for evaluating the different approaches.

The relevance judgements of the human experts, the entities used for evaluation and the result lists of the algorithms were encoded in the TREC format[5]. This allowed us to employ the TREC evaluation tools[6].

4.3 Retrieval Performance of the Algorithms

In our setting we are mostly interested in retrieving one relevant URL (i.e. webpage). So, we would like to measure the performance of the methods at providing the first

---

[4] It is worth mentioning, that the Wikipedia pages that served as "ancestors" of the DBPedia entities in many cases did **not** appear as most relevant representation for any of the approaches.

[5] The list of document URLs, queries and relevance judgements we used in this experiment is publicly available at `http://west.uni-koblenz.de/Research/DataSets/FindingURLs` under a Creative Commons license.

[6] The TREC evaluation tool `trec_eval` can be found at `http://trec.nist.gov/trec_eval/`

Table 1. Krippendorff's α – overall and per domain

	Krippendorff's α
Movies	0.733
Persons	0.808
Cities	0.682
Companies	0.770
Total	0.757

Table 2. Changes in performance using Borda count data source filtering **(complete dataset, 100 entities)**

	MRR	Precision@1	MAP
Label Search	+ .0876	+ .1582	- .0505
All Linked Labels	+ .0133	+ .0129	- .0061
PageRank	+ .0471	+ .0200	+ .0272
Topic PageRank	+ .3126	+ .1800	+ .1691
Focussed PageRank	+ .1778	+ .1242	+ .0914
HITS	+ .0140	+/- .0000	- .0105
Topic HITS	+ .0039	+ .0200	- .0545
Focussed HITS	+ .0469	+ .0164	- .0405
Focussed Link Count	- .0236	+/- .0000	- .0926

Table 3. Overall performance for each ranking method (complete dataset, 100 entities)

	Precision@1	MRR	Precision@5	MAP-cut@5	NDCG-cut@5
Label Search	**0.66**	**0.76**	**0.31**	**0.62**	**0.70**
All Linked Labels	0.11	0.03	0.03	0.06	0.08
PageRank	0.07	0.12	0.04	0.04	0.08
Topic PageRank	0.05	0.12	0.03	0.04	0.14
Focussed PageRank	0.30	0.38	0.12	0.21	0.33
HITS	0.19	0.29	0.09	0.15	0.23
Topic HITS	0.54	0.60	0.21	0.42	0.56
Focussed HITS	**0.62**	**0.66**	**0.24**	**0.50**	**0.60**
Focussed Link Count	0.59	0.64	0.24	0.48	0.59

relevant document at a high rank. In conclusion, our choice of evaluation metrics is clearly targeted to identify such methods.

The best suited measures for this purpose are the measures *Precision@1* and *Mean Reciprocal Rank (MRR)*. Thus, in the following discussion we focus on *Precision@1* and *MRR*. *Precision@1* allows for identifying how often a method provides a relevant document at the very first position. *MRR* instead gives an idea of how far down in the ranking list the first relevant document appears. For both methods we considered a document to be relevant, iff the human experts judged it as highly relevant. Furthermore, we considered other well established metrics for evaluation of ranked retrieval, such as *Mean Average Precision (MAP)* and *Normalized Discounted Cumulative Gain (NDCG)*. However, these metrics are of less importance for our setting. All these metrics are supported by the TREC evaluation tool.

We first discuss the performance of the algorithms without the Borda count based filtering using the semantic typing of the entities. In Table 3 the results of the experiments are summarized. We observed for the overall experiment with 100 entities that *Label Search* is the best method followed by *Focussed HITS*. The increase in performance is statistically significant at a level of $p = 0.05$.

According to our setting with four domains (*movies, persons, companies* and *cities*) in three stratas (small, medium and large number of results available from the search engine) we additionally calculate all measures over these different subsets and compare

Table 4. Overview of MRR score with respect to all domains and strata.

	Movies	Persons	Companies	Cities	Small	Medium	Large
Label Search	0.53	0.86	0.78	**0.85**	**0.76**	**0.79**	**0.72**
All Linked Labels	0.12	0.00	0.00	0.00	0.00	0.03	0.06
PageRank	0.26	0.08	0.08	0.07	0.06	0.13	0.18
Topic PageRank	0.06	0.11	0.20	0.12	0.06	0.10	0.14
Focussed PageRank	0.45	0.42	0.44	0.21	0.25	0.41	0.50
HITS	0.29	0.33	0.36	0.17	0.24	0.18	0.45
Topic HITS	0.52	0.80	0.80	0.26	0.51	0.67	0.62
Focussed HITS	**0.59**	**0.88**	**0.87**	0.31	0.61	0.71	0.68
Focussed Link Count	0.56	0.87	0.87	0.26	0.57	0.69	0.67

the outcomes. When looking at each domain separately, the outcome is quite different (c.f. Table 4 for details on the MRR results). Here, the *Label Search* method tends to be lower than *Focussed HITS* except for the cities domain. But given the smaller test set within each domain, we could not identify a statistic significance in these cases. Regarding the three strata (which contain entities of all domains each) results are comparable to the global observations.

In conclusion, we can state that the simple baseline method (*Label Search*) of using entity labels as keywords for a web search works remarkably well. Both the extension to context and the analysis of link networks perform lower. However, there seems to be some evidence that for certain domains an improvement can be achieved.

Using the semantic typing of entities in order to implement a data source based result filter is beneficial for virtually all methods. The results in Table 2 show that both MRR and Precision@1 increase for all methods except *Focussed Link Count*. Even the already very good results of *Label Search* are significantly improved, leading to absolute values for MRR of 0.8443 and Precision@1 of 0.8181. This means that due to the post-process filtering we obtain methods which for 4 out of 5 entities provide good web document representations at rank 1 of the result list and on average show the first relevant document at rank 1.18.

5 Related Work

Our approach makes use of Linked Data [2, 4] as a source of structured data whereas the purpose is finding good (or appropriate) URLs on the document web aiming for a preferably comprehensive representation of the given entity. To the best of our knowledge there have not been any efforts to address this problem to date. Though, our work relates to several topics in varying degrees. The probably most related area is on generating structured queries and applying it to unstructured data like the document web in one way or another. Similarly to us, some works use a search engine and corresponding keywords to transform queries on structured data to comprehensible syntax for web search engines [12]. The results (documents) are often ranked, as well. In order to raise precision and as a follow-up, n-tuples [19] or simply facts [6] are extracted using information extraction methods [13]. However, some works rather focus on the generation or extraction of entities or objects from unstructured data starting with structured queries [20]. In fact, keyword search also plays a crucial role in semantic search [5] itself where it is also used for entity/object retrieval [9, 8]. More elaborated work comes up with an entity relevance model (ERM) based on keywords from entities which in their context is used to generalize SPARQL queries on different RDF datasets [10] or to improve RDF ranking [3]. The results both of some of the works mentioned as well as our approach can be used to enrich datasets of Linked Data which has already been described in e.g. [17] using information extraction. Other works also trying to achieve this or in parts are e.g [18, 22] in the so-called Small Web of organizations etc. They learn relations for taxonomies from websites by utilizing the hierarchical links between organizational webpages not only within a single page.

6 Conclusions

We defined a novel task of mapping entities to web URLs by on the one hand utilizing the entities' connections to other entities in a knowledge base and on the other hand web search engines providing webpages from the entities' labels. We compared different methods employing link analysis and web search at large using 100 entities from four different domains in our evaluation data set. The methods were evaluated using common IR measures like Precision and Mean Reciprocal Rank (MRR). The best overall method turned out to be *Label Search* followed by *Focussed HITS*. Looking into the individual domains, the latter showed better results for three out of the four domains though not being statistically significant. An investigation of the reasons for this behaviour are part of future work. Furthermore, we presented a result list filtering approach based on semantic typing of entities and result set fusion over data sources. This filter boosted the performance of all methods and, in particular, achieved for *Label Search* very high values for MRR and Precision@1.

7 Acknowledgements

Special thanks for participating in the evaluation experiment goes to our colleagues at WeST. The research leading to these results has received partial funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 257859, ROBUST.

References

[1] ASLAM, Javed A. ; MONTAGUE, Mark: Models for metasearch. In: *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA : ACM, 2001 (SIGIR '01), 276–284

[2] BERNERS-LEE, Tim: *Linked Data - Design Issues*. `http://www.w3.org/DesignIssues/LinkedData.html`. Version: 2006

[3] BICER, Veli ; TRAN, Thanh ; NEDKOV, Radoslav: Ranking support for keyword search on structured data using relevance models. In: *Proceedings of the 20th ACM international conference on Information and knowledge management*. New York, NY, USA : ACM, 2011 (CIKM '11), S. 1669–1678

[4] BIZER, Christian ; HEATH, Tom ; BERNERS-LEE, Tim: Linked Data - The Story So Far. In: *International Journal on Semantic Web and Information Systems* 5 (2009), Nr. 3, S. 1–22

[5] BLANCO, Roi ; HALPIN, Harry ; HERZIG, Daniel M. ; MIKA, Peter ; POUND, Jeffrey ; THOMPSON, Henry S. ; TRAN, Duc T.: Entity Search Evaluation over Structured Web Data. In: *Proceedings of the 1st International Workshop on Entity-Oriented Search at SIGIR 2011*. Beijing, PR China, 2011

[6] BODEN, Christoph ; LÖSER, Alexander ; NAGEL, Christoph ; PIEPER, Stephan: FactCrawl: A Fact Retrieval Framework for Full-Text Indices. In: *WebDB*, 2011

[7] BRIN, Sergey ; PAGE, Lawrence: The Anatomy of a Large-Scale Hypertextual Web Search Engine. In: *Computer Networks* 30 (1998), Nr. 1-7, S. 107–117

[8] HALPIN, Harry: A Query-driven Characterization of Linked Data. In: *Proceedings of the Linked Data Workshop at the World Wide Web Conference*, 2009

[9] HALPIN, Harry ; HERZIG, Daniel M. ; MIKA, Peter ; BLANCO, Roi ; POUND, Jeffrey ; THOMPSON, Henry S. ; TRAN, Duc T.: Evaluating Ad-Hoc Object Retrieval. In: *Proceedings of the International Workshop on Evaluation of Semantic Technologies (IWEST 2010)*. Shanghai, PR China : 9th International Semantic Web Conference (ISWC2010), 2010

[10] HERZIG, Daniel M. ; TRAN, Duc T.: One Query to Bind Them All. In: *Proceedings of the Second International Workshop on Consuming Linked Data (COLD2011),*, CEUR Workshop Proceedings (CEUR-WS.org), 2011

[11] JÄRVELIN, Kalervo ; KEKÄLÄINEN, Jaana: IR evaluation methods for retrieving highly relevant documents. In: *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, 2000 (SIGIR '00), S. 41–48

[12] JING LIU, Alon H. Xin Dong D. Xin Dong: Answering Structured Queries on Unstructured Data. In: *In WebDB*, 2006, S. 25–30

[13] KASTRATI, Fisnik ; LI, Xiang ; QUIX, Christoph ; KHELGHATI, Mohammadreza: Enabling Structured Queries over Unstructured Documents. In: *Proceedings of the 2011 IEEE 12th International Conference on Mobile Data Management - Volume 02*. Washington, DC, USA : IEEE Computer Society, 2011 (MDM '11), S. 80–85

[14] KLEINBERG, Jon M.: Authoritative sources in a hyperlinked environment. In: *J. ACM* 46 (1999), S. 604–632

[15] KRIPPENDORFF, Klaus: *Content Analysis: An Introduction to Its Methodology*. Sage, 2004

[16] LANGVILLE, Amy N. ; MEYER, Carl D.: *Google's PageRank and Beyond: The Science of Search Engine Rankings*. Princeton University Press, 2006

[17] LERMAN, Kristina ; GAZEN, Cenk ; MINTON, Steven ; KNOBLOCK, Craig: Populating the Semantic Web. In: *Information Sciences* (2003)

[18] LI, Jianqiang ; ZHAO, Yu: A Case Study on Linked Data Generation and Consumption. In: *Linked Data on the Web (LDOW2008)*, 2008

[19] LÖSER, Alexander ; NAGEL, Christoph ; PIEPER, Stephan ; BODEN, Christoph: Self-supervised web search for any-k complete tuples. In: *Proceedings of the 2nd International Workshop on Business intelligencE and the WEB*. New York, NY, USA : ACM, 2011 (BE-WEB '11), S. 4–11

[20] PHAM, Kim C. ; RIZZOLO, Nicholas ; SMALL, Kevin ; CHANG, Kevin Chen-Chuan ; ROTH, Dan: Object search: supporting structured queries in web search engines. In: *Proceedings of the NAACL HLT 2010 Workshop on Semantic Search*. Stroudsburg, PA, USA : Association for Computational Linguistics, 2010 (SS '10), S. 44–52

[21] POPOV, Borislav ; KIRYAKOV, Atanas ; MANOV, Dimitar ; KIRILOV, Angel ; GORANOV, Ognyanoff M.: Towards Semantic Web Information Extraction. In: *Proceedings of ISWC (Sundial Resort)*, 2003

[22] ZHAO, Yu ; LI, Jianqiang: Domain Ontology Learning from Websites. In: *Proceedings of the 2009 Ninth Annual International Symposium on Applications and the Internet*. Washington, DC, USA : IEEE Computer Society, 2009, S. 129–132

Underspecified Scientific Claims in Nanopublications

Tobias Kuhn and Michael Krauthammer

Department of Pathology, Yale University School of Medicine
kuhntobias@gmail.com, michael.krauthammer@yale.edu

Abstract. The application range of nanopublications — small entities of scientific results in RDF representation — could be greatly extended if complete formal representations are not mandatory. To that aim, we present an approach to represent and interlink scientific claims in an underspecified way, based on independent English sentences.

1 Introduction

This position paper introduces an approach to represent and interlink scientific statements with Semantic Web techniques, where these statements themselves do not necessarily have complete formal representations. To this aim, an extension of the concept of nanopublications is sketched. Nanopublications have been developed to make it easier to find, connect and curate core scientific statements and to determine their attribution, quality and provenance [2]. Small RDF-based data snippets — i.e. nanopublications — rather than classical narrative articles should be at the center of general scholarly communication [4]. Nanopublications are based on RDF extended with named graphs [1].

There seem to be two possible types of nanopublications: they can represent claims or data. Data is directly observed from experiments or studies, whereas claims are obtained from generalizing from such data. The approach presented here has a clear focus on claims and not so much on data statements. “Malaria is transmitted by mosquitoes” [2] is a simple example of such a claim.

2 Approach

The proposed approach is based on the idea that any scientific claim can be broken down into small pieces of “atomic” claims, each of which can be represented as a relatively short independent sentence in English (or another natural language, possibly using highly technical vocabulary). Even though most claims found in scientific publications are probably more complex than “malaria is transmitted by mosquitoes”, it seems reasonable to assume that they can be written down as independent sentences. By *independent* we mean that the sentence can stand on its own and does not contain references like “this behavior” that refer to some surrounding text. Nanopublications follow the same basic idea, but require the claims to be fully formalized in RDF. We propose to extend nanopublications with English sentences, which are the central part of our model of scientific claims. Figure 1 shows a schematic representation of several

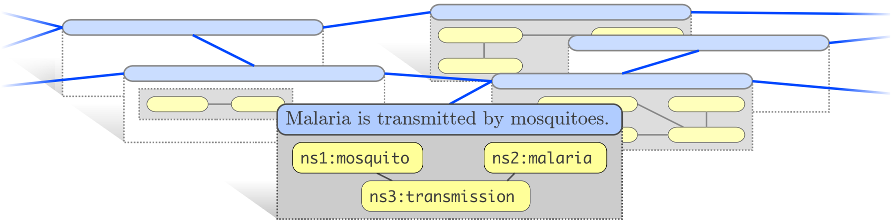


Fig. 1. Schematic representation of our model of scientific claims and their relations

claims according to our model. Each of the blue boxes contains an English sentence that represents the respective claim. Some claims have an additional formal representation in RDF (gray area), some do not (white area), and some are a mixture of the two (i.e. partial formalization). The important part is that all these claims, no matter whether formalized in RDF or not, can be interrelated and referenced, as indicated by the blue lines. These could be relations like "CLAIM1 contradicts CLAIM2" or "PERSON agrees with CLAIM". The white areas do not need to stay white forever: some of them might be filled with an RDF representation at a later point in time.

One could argue that any scientific claim can be represented in RDF in one way or another, given the appropriate vocabulary. In practice, however, the available vocabularies and ontologies are often not sufficient, especially for claims involving intended vagueness, modal concepts, temporal aspects, and novel ideas. RDF is extensible, but the development of accurate, useful and accepted models is a costly and slow process. By dropping the restriction that all claims need full RDF representations, the application range of nanopublications can be greatly extended.

As a more realistic example, let us consider the following sentence from the abstract of a biomedical article (PMID 19109537):

> [...] the risk of developing neurodegenerative disease in idiopathic REM sleep behavior disorder is substantial, with the majority of patients developing Parkinson disease and Lewy body dementia.

These are the two core claims that can be extracted as independent sentences:

- The risk of developing neurodegenerative disease in idiopathic REM sleep behavior disorder is substantial.
- The majority of patients with idiopathic REM sleep behavior disorder who develop a neurodegenerative disease develop Parkinson disease and Lewy body dementia.

To make these two sentences independent from each other, some parts have to be repeated. Still, the resulting sentences are reasonably short. The first one is a good example of vagueness in such claims ("substantial").

3 Integration

Here, we sketch how the ideas described above could be integrated into the existing standards. As a first step, to be able to refer to statements like scientific claims even if they are not fully represented in RDF, we need URIs for such entire statements. We put forward the point of view that such a statement is simply a string of characters to be interpreted according to a certain language, like English or German. We use URIs instead of RDF string literals, because the latter cannot be used in subject position of RDF triples. Such a statement URI could be `http://statements.org/en/Malaria+is+transmitted+by+mosquitoes`. Its semantics would be defined as all possible meanings that are given to it by the speakers of the respective language. This means that the authority behind such URIs (i.e. the fictitious `statements.org` in the given example) would not need to approve new statements, but everybody could make up such URIs and immediately use them. As a next step, we can integrate them in nanopublications.

The core part of a standard nanopublication is an *assertion* in the form of a named graph:

```
<> {
  :Pub1 np:hasAssertion :Pub1_Assertion .
  ...
}
:Pub1_Assertion { ... }
```

The curly brackets after `:Pub1_Assertion` would contain the actual assertion in the form of a set of RDF triples. To allow for underspecified assertions, we have to use a slightly more complex structure. With our approach, assertions consist of two subgraphs: a head and a body, where the body represents the actual (possibly unknown) formal representation:

```
<> {
  :Pub1 np:hasAssertion :Pub1_Assertion .
  :Pub1_Assertion np:containsGraph :Pub1_Assertion_Head .
  :Pub1_Assertion np:containsGraph :Pub1_Assertion_Body .
  ...
}
```

The head part is used to refer to different representations of the given assertion, such as the formal representation in the form of a named RDF graph or a natural representation in the form of an English sentence encoded in a URI:

```
:Pub1_Assertion_Head {
  :Pub1_Assertion
      st:asSentence st:en/Malaria+is+transmitted+by+mosquitoes ;
      st:asFormula :Pub1_Assertion_Body .
}
```

We can — but we are not obliged to — add a formalization of the given claim with `:Pub1_Assertion_Body { ... }`. Partial representations can be defined in a straightforward way with the help of subgraphs. Overall, this approach allows for defining nanopublications for virtually any possible scientific claim. Even claims that cannot be formalized in RDF can be included in the Semantic Web.

4 Discussion

There exist approaches like GeneRIF,[1] which is based on a similar idea but is restricted to a very specific domain (gene functions). Our approach is much more general and could subsume such specific solutions.

The approach sketched above in a certain sense uses Semantic Web techniques on a higher level than usual. Instead of representing relations between entities of the real world, we relate *statements* about the real world to other statements or entities. While such relations are no less fuzzy at this higher level than certain lower level relations, it is possible at the higher level to come up with a model that covers virtually all possible scientific claims. Many existing approaches based on RDF use this kind of higher level (e.g. provenance data for reified RDF triples), but they typically require the lower level to be spelled out too. We try to advocate the idea that we can describe things at the higher level without being specific about the lower one. Of course, it is always better to have RDF representations for both levels, but having just the higher one is better than nothing in cases where the lower level cannot be practically formalized (which might very well be the majority of cases).

Even though we only presented examples in English, our approach is inherently multilingual, as claims can be verbalized in different languages. Furthermore, instead of using unrestricted language, scientific claims could be expressed in a controlled natural language [5], in which case RDF representations could be automatically generated (depending on the used controlled natural language). Previous work indicates that this could be feasible for at least certain types of scientific claims [3].

We hope to be able to present a concrete proposal for underspecified nanopublications in the near future. We also plan to evaluate our approach by assessing scientific claims of existing publications.

References

1. Carroll, J., Bizer, C., Hayes, P., Stickler, P.: Named graphs, provenance and trust. In: WWW '05. pp. 613–622. ACM (2005)
2. Groth, P., Gibson, A., Velterop, J.: The anatomy of a nano-publication. Information Services and Use 30(1), 51–56 (2010)
3. Kuhn, T., Royer, L., Fuchs, N., Schroeder, M.: Improving text mining with controlled natural language: A case study for protein interations. In: DILS 2006. LNCS, vol. 4075, pp. 66–81. Springer (2006)
4. Mons, B., van Haagen, H., Chichester, C., den Dunnen, J., van Ommen, G., van Mulligen, E., Singh, B., Hooft, R., Roos, M., Hammond, J., Kiesel, B., Giardine, B., Velterop, J., Groth, P., Schultes, E.: The value of data. Nature genetics 43(4), 281–283 (2011)
5. Wyner, A., Angelov, K., Barzdins, G., Damljanovic, D., Davis, B., Fuchs, N., Hoefler, S., Jones, K., Kaljurand, K., Kuhn, T., Luts, M., Pool, J., Rosner, M., Schwitter, R., Sowa, J.: On controlled natural languages: Properties and prospects. In: CNL 2009, LNCS, vol. 5972, pp. 281–289. Springer (2010)

---

[1] `http://www.ncbi.nlm.nih.gov/gene/about-generif`

Discovering Names in Linked Data Datasets

Bianca Pereira[1], João C. P. da Silva[2], and Adriana S. Vivacqua[1,2]

[1]Programa de Pós-Graduação em Informática,
[2] Departamento de Ciência da Computação
Instituto de Matemática, Universidade Federal do Rio de Janeiro, Brazil
`bianca.pereira@ppgi.ufrj.br, {jcps,avivacqua}@dcc.ufrj.br`

Abstract. The Named Entity Recognition Task is one of the most common steps used in natural language applications. Linked Data datasets have been presented as promising background knowledge for Named Entity Recognition algorithms due to the amount of data available and the high variety of knowledge domains they cover. However, the discovery of names in Linked Data datasets is still a costly task if we consider the amount of available datasets and the heterogeneity of vocabulary used to describe them. In this work, we evaluate the usage of `rdfs:label` as a property referring to entities' name and we describe a set of heuristics created to discover properties identifying names for named entities in Linked Data datasets.

Keywords: Named Entity, Named Entity Recognition, Linked Data

1 Introduction

Named Entity Recognition (NER) in natural language texts is one of the most common tasks in Natural Language Processing. Since the sixth Message Understanding Conference (MUC) with the emergence of the term "named entity" and the formalization of the NER task, the techniques for recognizing names in texts have greatly evolved. Additionally, better knowledge bases not only for recognition of names but also for its disambiguation have been developed.

A named entity (NE) is an entity that can be identified by a proper name [2]. Originally NEs were instances of person, organization or location classes and also dates and numeric values. Nowadays there are many other classes that identify NEs [3] [4].

Techniques for NER range from dictionary-based approaches to rule or machine learning ones [10]. Over time, different knowledge bases have been used as background knowledge for the NER task: from manually created lists to datasets using knowledge available on the Web [4]. Recently, with the emergence of databases in Linked Data format, Linked Data datasets have been presenting as promising sources for NEs.

The Linked Open Data cloud (LOD cloud) provides knowledge in diverse human knowledge domains, including not only the most common types of entities mentioned previously as NE types, but also entities in the field of music, video, biology, among many others.

Several recent studies and tools have appeared linking NE mentions in free text to Linked Data resources [11]. The first step in this task is the discovery of which classes identify NEs and which properties refer to their names in the LD dataset. Only after this step, that is usually performed manually, the comparison between a name of a resource from the dataset with the name mentioned in the text can be made.

The heterogeneity in metadata used to describe LD datasets is one of the difficulties in using datasets from LOD cloud for NER. As a consequence of this heterogeneity, the identification of names in a LD dataset is a hard task, which starts with the identification of properties that refer to names, a costly task in and of itself. Due to this, works using LD datasets for NER and entity linking still use only a limited number of datasets available on the LOD cloud.

Our goal in this paper is to propose heuristics that help to determine which properties contain names of NEs as their values, henceforth called PIN (Property that Identifies Names), in generic LD datasets. Our results can be used to enable current tools to work with different datasets without requiring a manual analysis to understand all the metadata used to describe resources in a LD dataset.

The rest of this paper is organized as follows: in section 2 we present related work on using Linked Data for NER. In section 3 we explain the NER task and which features of Linked Data datasets can be used to perform this task. Following that, we evaluate the feasibility of using the `rdfs:label` property as a sole source of names in section 4 and present our algorithms for PIN identification in section 5. In section 6 we evaluate our heuristics and present our conclusions in section 7.

2 Related Works

There is a large number of tools (mostly commercial) using LD datasets for NER and linking. Despite the large number of LD datasets available on the LOD cloud they work only with a small set of them.

DBpedia Spotlight [9] is a tool which its main goal is to recognize names from a text and link them to resources from DBPedia[1]. DBpedia Spotlight uses a set of possible names also called surface forms created from the `rdfs:label` property as well as written variations of names taken from Wikipedia links. It is highly optimized for DBPedia and achieves high precision.

The work of Hoffart et al. [6] performs the same task as DBPedia Spotlight but it uses YAGO [12] as its source for NEs and the `yago:means` property as a source for names.

Large KB Gazetter[1] is a plug-in for GATE Platform[2] that enables using a generic LD dataset as a dictionary. It aims to allow any SPARQL query to be used as a source for NE names.

All previous work require knowledge about every vocabulary used to describe the LD datasets in order to use them as a source for NEs. We propose to

---

[1] http://nmwiki.ontotext.com/lkb_gazetteer/

[2] http://gate.ac.uk

use heuristics to allow them to identify NEs and their names from generic LD datasets without requiring manual analysis.

3 Using Linked Data for Named Entity Recognition

NER algorithms which are dictionary-based require some effort to create the dictionary used as background knowledge. Instead of manually creating these dictionaries, websites such as Wikipedia have been used as external knowledge for NER[7]. These new knowledge bases require different algorithms to structure their knowledge and extract entity names.

The main advantage of using LD datasets for NER tasks is that data is already structured. Algorithms that use Wikipedia as a gazetteer require preprocessing to extract all possible names of entities contained in its various pages. In another hand, LD datasets allow the creation of SPARQL queries for data retrieval, making the whole process much simpler. Another feature of LD datasets is the description of data using vocabularies or ontologies. This description enables the determination of an entity's type (person, place, etc.) through a simple query.

LD datasets are structured using RDF [5] resources to describe entities from the real world. Each resource is described through properties and relationships with other resources. Both property and relationship are specified by vocabularies or ontologies that indicate to a human what each one of them means. Furthermore, RDF Schema (RDFS) [8] presents a set of properties commonly used to describe resources in LD datasets. In our work, the `rdfs:label` is a relevant property because it describes a human-readable name for RDF resources often a NE name.

A starting point in searching for NE names in a LD dataset would be to use the contents of the `rdfs:label` property as DBPedia Spotlight does. In the following section, we present an analysis of the usage of `rdfs:label` as a unique source for names in LD datasets.

4 Using `rdfs:label` as a name

The most intuitive approach for the identification of names from NEs in a LD dataset is using the `rdfs:label` property. To verify the applicability of this approach, we conducted an analysis of a small set of datasets from the LOD cloud. Our goal was to see if this approach was sufficient for the task of acquiring names for NEs in generic LD datasets.

The first step was to select LD datasets that contain resources describing NEs that explicitly specify their name using properties. We selected a set of domain-specific datasets: Linked Movie Database [5], Geo Linked Data[8], Linked Brainz[3] and Jamendo (DBTune)[4]. The first dataset contains data from films

---

[3] http://linkedbrainz.c4dmpresents.org/

[4] http://dbtune.org/jamendo/

with information such as actors, characters and performances. Geo Linked Data describes spatial data, such as places and points of interest. The last two datasets are about music but Jamendo focuses on indepent musical groups and singers.

In the Linked Movie Database the `rdfs:label` property is present in almost all classes of entities described by the dataset. Among the classes there are those representing NEs and those representing other types of entity. We noticed that these other entities are, in fact, relationships between more than two entities.

For this first dataset, if we always use the `rdfs:label` property as a source of names we would extract some incorrect names. Further, given that the `rdfs:label` property is used to provide a human-readable label, and not necessarily the name of the entity, the NEs present in the dataset usually had the entity class as part of the value of the `rdfs:label` property. For instance, the entity identified by the URI http://data.linkedmdb.org/resource/film_character/253 is of *Film Character* class and contains the text "Kate (Film Character)" in its `rdfs:label` property. On the other hand, there is a set of properties that identify the names of various NEs in the dataset: `actor_name`, `director_name`, `cinematographer_name`, `editor_name`, among others. In the example mentioned above, the name of the entity is represented by `film_character_name` property whose value is "Kate".

The second dataset is Geo Linked Data. This dataset consists of ten named graphs, where seven of them are datasets and the others contain some metadata. Among these seven, we excluded two, which referred to statistical indexes and one that referred to years, which is not our focus at this point. Of the four remaining datasets we could verify that all names from NEs are exclusively described by the `rdfs:label` property. In addition, this property does not appear in entities that are not NEs. Even though it is possible to extract all the names using only the `rdfs:label` property, a large part of the entities have values in a format not commonly used. For example, an entity of *Aeropuerto* (*Airport*) class has the string "Sevilla, Aeropuerto de" as the `rdfs:label` property value, rather than "Aeropuerto de Sevilla".

The third dataset selected was the Linked Brainz, a dataset created from information available on the MusicBrainz website. Linked Brainz describes entities in the music domain such as singers, music groups and their work. It has ten classes that represent NEs but only seven use `rdfs:label` to describe the name of the entities. All NEs, even those using `rdfs:label`, use other properties not only to describe the most common name of the entity, but also to describe alternative names. The properties used are: `skos:altLabel` and `skos:notation`, described by the SKOS vocabulary, `foaf:name` defined by the FOAF vocabulary, `dc:title` described by the Dublin Core vocabulary, `vo:sortLabel` from the OpenVocab[5] vocabulary, and another `geo:name` property described by the Basic Geo (WGS84 lat/long)[6] vocabulary . Given that not all NE classes use the `rdfs:label` property, using only this property would exclude useful information.

---

[5] http://open.vocab.org/docs

[6] http://www.w3.org/2003/01/geo/

The last dataset in our analysis was Jamendo, from the DBTune.org website. This dataset was generated from the information of the Jamendo website and contains information about independent music groups and artists, and their work. This dataset does not use `rdfs:label` to describe their entities. All names are described by two properties: `dc:title` from the Dublin Core vocabulary and `foaf:name` from FOAF vocabulary.

We could verify that a range of properties may contain the names for NEs in a LD dataset. As this list is not fixed because it depends on the vocabularies used by each dataset, it is not possible to create an algorithm that considers the full list of every possible property that identify names of NEs. Thus we need to be able to identify automatically which properties contain names of NEs for each dataset.

5 Discovering properties that refer to names

In this section, we present a set of heuristics to identify PIN (Property that Identifies Names) in LD datasets. Each algorithm receives a LD dataset as input and returns a set of PIN for each class in the dataset. If a specific class does not represent NEs, the algorithm must not return a PIN for this class, otherwise, it should return one or more PIN. Each heuristic was created based on the assumption that names are represented by proper names. To identify if a given string is a proper name we are considering that every string with at least 50% of its words capitalized is a proper name.

The same basic algorithm is used differing only in the heuristic (score function and requisites) used.

Each algorithm recovers every class in a LD dataset and every property p that has a literal as its value for each class c found. After that, for each class c and each property p used to describe instances of this class the algorithm calculates a score based on the occurrence of proper names as values of p. Each heuristic identifies as a PIN the best scored property according to their respective requisite. As our goal is identify PIN that differ from one dataset to another we will give priority for other properties than `rdfs:label`.

Four heuristics were developed, and are described in the following subsections: Naive, Parametrized Naive, Multivalue and Multivalue with Threshold. The Naive and Parametrized Naive heuristics consider only the best scored property for each class (return a single result) and the Multivalue and Multivalue with Threshold heuristics return every property that score higher than a given value.

5.1 Naive Heuristic

The Naive heuristic is the simplest and returns the property that has the highest occurrence (higher score) of proper names as its value for each class.

The score(p,c) function is given by the sum of each occurrence of a proper name as a value of the property p in entities from class c (e_c) :

$$score_n(p,c) = \textstyle\sum_{e_c} \left\{ \begin{matrix} 1, \; if \; p_{\,value} = proper\ name \\ 0, \; otherwise \end{matrix} \right\} \tag{1}$$

If every score for properties is equal to zero, class c has no PIN.

5.2 Parametrized Naive Heuristic

The Naive heuristic does not impose any kind of restriction for a proper name to be recognized as a name. Therefore, this heuristic can return not only names but also acronyms and possibly descriptions that have a high frequency of capitalized words. Acronyms are usually short strings about 2 to 4 characters in upper case while descriptions tend to be paragraphs or a set of paragraphs formed by a large number of characters.

The Parametrized Naive heuristic aims to avoid occurrence of description texts and acronyms as value for PIN. The heuristic uses two constraints *min* and *max* to restrict the length of the string accepted as a name.

The score function (Formula 2) only counts occurrences in which p value is a proper name with length greater than or equal to *min* and lower than or equal to *max*.

$$score_{pn}(p,c) = \sum_{e_c} \left\{ \begin{array}{l} 1,\ if\ p_{\,value} = proper\ name, length(p_{\,value}) \in [min, max] \\ 0,\ otherwise \end{array} \right\} \tag{2}$$

If the highest score is equal to 0 for a given class then there are not any PIN associated with it.

5.3 Multivalue Heuristic

Given that many entities can be reffered to by a set of names instead of a unique name, we propose a heuristic to identify these alternate names as well. We can identify the most used name as a preferred name, and other names as alternate names or acronyms referring to the same entity. The previous heuristics return only one property as a PIN while this heuristic retrieves every possible PIN including properties referring to acronyms. This heuristic is the same as the Parametrized Naïve when considering the min value equal to zero and adding a parameter with the number of returned PIN per class with a value equal to one.

In the Multivalue heuristic we intend to accept acronyms as valid values but not descriptions. In this way, *max* is also used in the score function (Formula 3).

$$score_{Multi}(p,c) = \sum_{e_c} \left\{ \begin{array}{l} 1,\ if\ p_{\,value} = proper\ name, length(p_{\,value}) \leq max \\ 0,\ otherwise \end{array} \right\} \tag{3}$$

The requisite to decide if a property p can be chosen as a PIN for a given class is if its score is higher than zero.

5.4 Multivalue with Threshold Heuristic

This last heuristic is characterized by recognizing more than one property as a PIN for each class and identifying only the best scored properties rather than every property with a score greater than zero.

The score for Multivalue with Threshold is calculated based on the relative frequency of occurrence of property p referring to names for entities in class c. In other words, a property p will be considered a PIN only if it also appears describing a percentage of entities higher than a given threshold for a given class. The score function can be seen in Formula 4.

$$score_{Threshold}(p, c) = score_{Multi}(p, c)/|e_c| \tag{4}$$

A given entity in a real world can have many alternative names. Due to this, the Linked Data resource representing this entity may use the same property many times to describe these diverse names. In order to give the same weight for each property in the dataset this heuristic only counts one occurrence of each property for each instance of a class.

The threshold will be used to select PIN. Every property with a score higher than the threshold value will be considered as a PIN for class c.

6 Experiments

In our evaluation process we used the same datasets aforementioned excepting Geo Linked Data and Linked Brainz. The Geo Linked Data was not used because it only uses `rdfs:label` as a PIN. The Linked Brainz has about two billion triples what requires a machine with a high processing power and memory available to enable a good processing time.

The evaluated datasets were: Jamendo (DBTune.org) and Linked Movie Database[5].

6.1 Gold Standard

To enable the evaluation of our heuristics we have created a gold standard. It was manually developed and consists of a list of classes that have NEs as its instances and a list of PIN associated to those classes. We assume at this point that if a class represents NEs then each one of its instances is a NE.

The steps to create the gold standard are as follows:

- Identify all classes describing resources in the dataset.
- For each class identify all properties whose value is a literal.
- Analyze the meaning of each class and property
- Select classes that define NEs as its instances
- Select a set of PIN for each class that define NEs.

For the first two steps we used SPARQL queries to list all classes and their respective properties. Having all classes and their respective properties we analyzed their meaning. In other words, we have searched for the ontology description and if it does not exist or it is inconclusive we manually analyzed few instances of each class. These ontology descriptions are mainly searched based on their namespace and the respective LD dataset's project website.

Based on the meaning of each class and property we could identify those that describe NEs. Each NE is identified by one or more names. We say that a class represents NEs if it has one or more PIN associated with its instances. We assumed that names are proper names and are infrequently shared by many instances from the same class.

Each PIN was identified as referring to a preferential name, an alternative name or an acronym. Preferential name is the most frequent name for a given NE. Properties associated with preferential names appears only once in a Linked Data resource that describes a NE but those associated with alternative names and acronyms may appear zero or more times. Acronyms are identified by having many capitalized characters in a single word while alternative names do not have this feature.

There are some dataset features that should be pointed out. In every dataset there are classes that do not refer to NEs and therefore do not have PIN. Analyzing the meaning of classes we notice that a class does not always describe NEs or even entities. Some examples are: class *Playlist* in Jamendo and class *Performance* in Linked Movie Database. The instances of *Performance* class are not NEs but ternary relationships involving actors, films and characters. These instances have properties responsible to link them with other entities but their values are string representations of related entities names and not URIs, as is recommended for relationships in Linked Data.

Another feature founded is that we can not use only the ontology description to identify if a class describes NEs because sometimes the ontology description is not available as in the case of Linked Movie Database.

We also have to make some observations about properties classified as containing preferential or alternative names. In Linked Movie Database there are some properties that share the same values such as `dc:title` and `rdfs:label` for the class *Film* then in this case both were identified as referring to preferential names.

6.2 Evaluation

The goal for the heuristics developed is, primarily, the identification of PIN associated with preferential names. If a heuristic identifies alternative names or acronyms we understand this as a correct answer however it is not the best answer. In the case of Multivalue and Multivalue with Threshold we intend to retrieve every PIN from the LD dataset. In any case we understand the identification of PIN for classes that do not identify NEs as an error.

Our experiments were processed in two steps. Each one evaluates all heuristics using a different dataset. For each dataset we made a local installation using the RDF files provided by CKAN website in order to provide results that do not change during our experiments.

Jamendo Jamendo is a small dataset with 11 classes being 3 of them describing NEs, each one containing 1 PIN.

The Naive Heuristic found all three properties correctly plus another one: `mo:text` from *Lyrics* class. This represents song letters which are reproduced as a value for property `mo:text`. The incorrect classification of this property is due to the fact that we are considering every string as possible names regardless whether a string has many or few letters.

In the Parametrized Naive Heuristic, we considered that the value of the candidate properties should have minimum length of 4 and maximum length of 100 characters. Although the number of occurrence for `mo:text` has dropped, the same classes and properties were obtained.

There was an increase of false positive candidate PIN with the Multivalue Heuristic because it discovered a new property for each of the classes *Record* and *MusicArtist* since it selects not only the best scored properties but any property of a class with non-zero score. In the Multivalue with threshold heuristic, we established that the maximum length of a string is 100, and use values 0.4, 0.6, 0.8 and 0.95 as threshold. With 0.4, 0.6 and 0.8 as threshold, the classes *Record* and *MusicArtist* and their properties were correctly identified. Increasing the threshold to 0.95, no property was identified. The overall results of application of each heuristic can be seen in Table 1.

Table 1. Results from PIN identification heuristics for Jamendo Dataset

Heuristic	**PIN found**	**False positives**
Naive	3 (100%)	1
Parametrized Naive	3 (100%)	1
Multivalue	3 (100%)	3
Multivalue with Threshold (0.4 , 0.6, 0.8)	2 (66.67%)	0
Multivalue with Threshold (0.95)	0 (0%)	0

Linked Movie Database The Linked Movie Database[5] has a large number of classes (53 classes being 34 describing NEs). It has its own ontology that, unfortunately, does not have description for its classes and properties available on CKAN or dataset website.

This dataset has a particular feature. There are some entities that belongs to two different classes. For example, the resource identified by the URI `http://data.linkedmdb.org/resource/actor/1` has two `rdf:type` values: *Actor* and *Person*. Due to this the *Person* class does not have a unique PIN associated with preferential names and each one of these PIN does not appear together.

The Naive heuristic identified 31 out of 41 PINs. There was a draw between the hits number for `rdfs:label` and the correct PIN in many classes but as `rdfs:label` has lower priority according to our algorithm the correct properties were identified as PIN. Only two PIN had more hits than `rdfs:label`: `movie:film_character_name` from *Character* class and `movie:film_company_name` from *Film Company* class.

There were also a large number of false positives due to two features of the dataset. The first is that the Linked Movie Database has classes we did not recognize as identifying NEs such as *Film Focus* or *Film Distribution Medium.* Regardless whether these classes do not identify NEs they have properties describing their names such as "Theatre" or "CD" for instances of *Film Distribution Medium.* The second feature is literals as values for properties referring to relationships. There are some classes such as *Performance* that identify ternary relations between instances of *Actor*, *Character* and *Film* but properties relating an instance of *Performance* with instances of the other classes have strings as their values instead of URIs. In this case, these relationships were returned as PIN due to a wrong description used in the dataset.

The Parametrized Naive heuristic identified the correct PIN (`movie:country_name`) instead of properties referring to acronyms(`movie:country_iso_alpha3`) due to parameters min = 4 and max = 100 as expected. This heuristic had more false positives due to the recognition of `rdfs:label` rather than the correct PIN for some classes. It happens because the `rdfs:label` value is composed by the value of the correct name plus the class name so some values for the correct PIN were discarded by the min parameter but the `rdfs:label` for the respective entity was not discarded because its value has more characters. If `rdfs:label` has at least one hit more than the correct PIN, this heuristic will recognized it as a PIN.

The Multivalue heuristic identified all PIN from the dataset. It could also identified every PIN for the *Person* class because it does not restrict the number of hits to recognize a property as a PIN. Although the 100% of recall this heuristic retrieved a large number of false positive. This heuristic had the same problem identified in the Naive Heuristic but the previous get only one property as a PIN.

At last, the Multivalue with Threshold Heuristic did not identified any PIN for the *Person* class as we expected. It still have a high number of false positives due to features aforementioned. The overall results can be seen in Table 2.

Table 2. Results of PIN identification heuristics for Linked Movie Database

Heuristic	**PIN found**	**False positives**
Naive	31 (75.61%)	21
Parametrized Naive	19 (46.34%)	33
Multivalue	41 (100%)	85
Multivalue with Threshold (0.4)	35 (85.36%)	68
Multivalue with Threshold (0.6)	35 (85.36%)	65
Multivalue with Threshold (0.8)	34 (82.93%)	61
Multivalue with Threshold (0.95)	32 (78.05%)	54

6.3 Analysis

Each heuristic has different characteristics and the best fit will depend on the dataset features.

The Naive and Parametrized Naive Heuristics do not prioritize recall. They return only one PIN for each NE class and these PIN refer only to preferable names for NEs. Thus they presented a better precision because they usually identify the right PIN but they do not return every possible PIN from the dataset. In applications that need to recognize names in a LD dataset without many errors these two heuristics are preferable. Moreover, the Multivalue and Multivalue with Threshold have a better recall. The Multivalue Heuristic returns every possible PIN from the dataset recognizing every PIN in our experiments but also returning a high number of false positives. The Multivalue with Threshold Heuristic allows mantaining the recall but with more precision. As we increase the value of the Threshold we have less false positives with a good recall. These last two heuristics are preferable in applications that only need PIN to reduce the search space for names. The results for each heuristic can be seen in Table 3.

In addition our heuristics could also identify which classes have NE as their instances. Each class that have at least one PIN recognized can be seen as NE class.

Despite the high number of false positives, our heuristics have obtained a reasonable result in this preliminary study. The next step is evaluating our heuristics using a bigger set of LD datasets in order to acquire more insights about common Linked Data features and how the heuristics perform with new features. The heuristics may also be important to provide an overview of which properties are actually used to describe names for NEs in LD datasets and to reduce the search space for names of NEs described in the dataset. Therefore, they help using generic datasets in LD as knowledge bases for NER and linking tasks.

	Jamendo			Linked Movie Database		
Heuristics	Precision	Recall	F-Score	Precision	Recall	F-Score
Naive	0.75	1	0.8571	0.5962	0.7561	0.6667
Parametrized Naive	0.75	1	0.8571	0.3654	0.4634	0.4085
Multivalue	0.5	1	0.6667	0.3254	1	0.4910
Multivalue (Threshold = 0.4)	1	1	1	0.3398	0.8536	0.4861
Multivalue (Threshold = 0.6)	1	1	1	0.35	0.8536	0.4964
Multivalue (Threshold = 0.8)	1	1	1	0.3579	0.8293	0.5
Multivalue (Threshold = 0.95)	0	0	0	0.3721	0.7805	0.5039

Table 3. Overall Results for the application of every heuristic for PIN identification

7 Conclusion

In this paper we started to address the problem of finding names for NE in generic Linked Data datasets. Due to the heterogeneity in the description of these datasets, the identification of properties that have names as their values is not trivial. We analyzed the feasibility of using `rdfs:label` as a unique source for NE names and then presented a set of heuristics for identification of PIN, those properties whose values may be names for NEs.

We conducted a preliminary study using our heuristics with two datasets from the LOD cloud. Both datasets have a significant number of triples, classes and properties. We created a gold standard to evaluate our heuristics. Based on the results of the evaluation, we discovered that our heuristics can be used to identify PIN for these LD datasets, but given that the heuristics' accuracy were not 100%, we suggest that they undergo a process of manual review before they are used in applications that require 100% accuracy.

8 Acknowledgments

This work was supported by a grant from CAPES, Brazil.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: A nucleus for a web of open data. The Semantic Web pp. 722–735 (2007)
2. Chinchor, N.: Overview of muc-7/met-2 (1998)
3. Cohen, W., Sarawagi, S.: Exploiting dictionaries in named entity extraction: combining semi-markov extraction processes and data integration methods. In: Proceedings of the tenth ACM SIGKDD. pp. 89–98. ACM (2004)
4. Etzioni, O., Cafarella, M., Downey, D., Popescu, A., Shaked, T., Soderland, S., Weld, D., Yates, A.: Unsupervised named-entity extraction from the web: An experimental study. Artificial Intelligence 165(1), 91–134 (2005)
5. Hassanzadeh, O., Consens, M.: Linked movie data base. In: Proceedings of the 2nd Workshop on Linked Data on the Web (LDOW2009) (2009)
6. Hoffart, J., Yosef, M., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., Weikum, G.: Robust disambiguation of named entities in text pp. 782–792 (2011)
7. Kazama, J., Torisawa, K.: Exploiting wikipedia as external knowledge for named entity recognition. In: Proceedings of the EMNLP-CoNLL 2007. pp. 698–707 (2007)
8. Lopez-Pellicer, F., Silva, M., Chaves, M., Javier Zarazaga-Soria, F., Muro-Medrano, P.: Geo linked data. In: Database and Expert Systems Applications. pp. 495–502. Springer (2010)
9. Mendes, P., Jakob, M., García-Silva, A., Bizer, C.: Dbpedia spotlight: shedding light on the web of documents. In: Proceedings of the 7th International Conference on Semantic Systems. pp. 1–8. ACM (2011)
10. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticae Investigationes 30(1), 3–26 (2007)
11. Rizzo, G., Troncy, R.: Nerd: A framework for unifying named entity recognition and disambiguation extraction tools. EACL 2012 p. 73 (2012)
12. Suchanek, F., Kasneci, G., Weikum, G.: Yago: A large ontology from wikipedia and wordnet. Web Semantics: Science, Services and Agents on the World Wide Web 6(3), 203–217 (2008)

Can Entities be Friends?

Bernardo Pereira Nunes[1,2], Ricardo Kawase[1], Stefan Dietze[1], Davide Taibi[3], Marco Antonio Casanova[2], Wolfgang Nejdl[1]

[1] L3S Research Center - Leibniz University Hannover - Germany
{nunes, kawase, dietze, nejdl}@L3S.de

[2] Department of Informatics - PUC-Rio - Rio de Janeiro - Brazil
{bnunes, casanova}@inf.puc-rio.br

[3] Italian National Research Council - Institute for Educational Technology - Palermo - Italy
davide.taibi@itd.cnr.it

Abstract. The richness of the (Semantic) Web lies in its ability to link related resources as well as data across the Web. However, while relations within particular datasets are often well defined, links between disparate datasets and corpora of Web resources are rare. The increasingly widespread use of cross-domain reference datasets, such as Freebase and DBpedia for annotating and enriching datasets as well as document corpora, opens up opportunities to exploit their inherent semantics to uncover semantic relationships between disparate resources. In this paper, we present an approach to uncover relationships between disparate entities by analyzing the graphs of used reference datasets. We adapt a relationship assessment methodology from social network theory to measure the connectivity between entities in reference datasets and exploit these measures to identify correlated Web resources. Finally, we present an evaluation of our approach using the publicly available datasets Bibsonomy and USAToday.

Keywords: Linked data, data integration, link detection, semantic associations

1 Introduction

The emergence of the Linked Data principles [2] has led to the availability of a wide variety of structured datasets[1] on the Web. However, while the central goal of the Linked Data effort is to create a well-interlinked graph of Web data, links are still comparatively sparse, often focusing on a few highly referenced datasets such as DBpedia, YAGO [18] and Freebase, while the majority of data exists in a rather isolated fashion. This is of particular concern for datasets which describe the same or potentially related resources or real-world *entities*. For instance, within the academic field, a wealth of potentially related entities are described in bibliographic datasets and domain-specific vocabularies, while no explicit relationships are defined between equivalent, similar or related resources [5].

---

[1] `http://lod-cloud.net/state`

Furthermore, knowledge extraction, Named Entity Recognition (NER) tools and environments such as GATE [4], DBpedia Spotlight[2], Alchemy[3], AIDA[4] or Apache Stanbol[5] are increasingly applied to automatically generate structured data (entities) from unstructured resources such as Web sites, documents or social media. However, while such automatically generated data usually provides an initial classification and structure, for instance, the association of terms with entity types defined in a structured RDF schema (as in [14]), entities extracted via Natural Language Processing (NLP) techniques are usually noisy, ambiguous and lack sufficient semantics. Hence, identifying links between entities within such a particular dataset as well as with pre-existing knowledge serves three main purposes (a) enrichment, (b) disambiguation and (c) data consolidation. Often, dataset providers aim at *enriching* a particular dataset by adding links (*enrichments*) to such comprehensive reference datasets. Current interlinking techniques usually resort to map entities which refer to the same resource or real-world entity, e.g., by creating `owl:sameAs` references between an extracted entity representing the city "Berlin" with the corresponding Freebase and Geonames[6] entries.

However, additional value lies in the identification of related entities within and across datasets, e.g., by creating `skos:related` or `so:related` references between entities that are to some degree related [7]. In particular, the widespread adoption of reference datasets such as DBpedia or Freebase opens opportunities to discover related entities by analyzing the graph of used joint reference datasets to measure the relatedness, i.e., the semantic association [1, 17] between a given set of enrichments and, thus, entities. However, uncovering this relation would require the assessment of such reference graphs in order to (a) identify the paths between these given enrichments and (b) measure their meaning with respect to some definition of semantic relatedness.

In this paper, we describe an approach to identify relationships between disparate entities by analyzing the graphs of reference datasets using an algorithm adopted from social network theory and extended to the needs of our overall vision. The main goal is to detect and quantify the relatedness between given sets of disparate entities and thus, Web resources. We provide a general-purpose approach, which exploits the number of paths and the distance (length of a path) between given entities to compute a relatedness score between (a) extracted entities and (b) associated Web resources such as documents.

The remainder of this paper is structured as follows. Section 2 formally describes the problem addressed. Section 3 introduces our method. Section 4 and Section 5 show the evaluation strategies and their results, respectively. Section 6 reviews the literature. Finally, Section 7 summarizes our contributions and discusses future work.

---

[2] `http://dbpedia.org/spotlight`
[3] `http://www.alchemyapi.com`
[4] `http://adaptivedisclosure.org/aida/`
[5] `http://incubator.apache.org/stanbol`
[6] `http://www.geonames.org`

2 Problem Definition

In this work, we aim at finding and measuring the connectivity, i.e. semantic association, between disparate entities and use it as a measure to compute the relatedness of documents which refer to such entities. Exploiting implicit semantic relationships between entities, beyond mere linguistic similarity between different Web resources, allows to uncover different kind of semantic relationships between Web resources.

According to Sheth et al. [16], a semantic association between two resources exists if they have semantic connectivity or semantic similarity. In this work we focus on the semantic association given by semantic connectivity.

For instance, let $G = (E, P)$ be a graph (e.g. RDF dataset), where E and P denote a finite set of entities and properties, respectively. A property $p_i \in P$ is represented by a finite set of entities $\{e_i, e_j\}$, where $e_i, e_j \in E$. Thus, given two entities e_1 and e_n, they have *semantic connectivity* [16] iff exists at least one path $\rho^{<max(l)>}_{(e_1,e_n)} = \{\{e_1, e_2\}, \{e_2, e_3\}, ..., \{e_{n-1}, e_n\}\}$ that links each other with a maximum l properties between them. Contrasting with [16], we constrained the paths to a maximum length $max(l)$, since reference datasets (e.g. DBpedia and Freebase) are densely connected and, hence, the probability that any two entities be connected through longer paths tends to be high.

For performance reasons (see Section 3), we assume undirected graphs. Therefore, the paths $\rho^{<max(l)>}_{(e_1,e_n)} = \{\{e_1, e_2\}, \{e_2, e_3\}, ..., \{e_{n-1}, e_n\}\}$ and $\rho^{<max(l)>}_{(e_n,e_1)} = \{\{e_n, e_{n-1}\}, ..., \{e_3, e_2\}, \{e_2, e_1\}\}$ are considered to be equal, that is, $\rho^{<max(l)>}_{(e_1,e_n)} = \rho^{<max(l)>}_{(e_n,e_1)}$.

Thus, the semantic connectivity between two given entities e_i and e_j can be measured by a score $\lambda(\delta^{<max(l)>}_{(e_i,e_j)})$, where $\delta^{<max(l)>}_{(e_i,e_j)}$ is a set of paths $\rho^{<max(l)>}_{(e_i,e_j)}$. We say that there is a semantic association between e_i and e_j iff $\lambda(\delta^{<max(l)>}_{(e_i,e_j)}) > 0$, and that there is no semantic association between e_i and e_j iff $\lambda(\delta^{<max(l)>}_{(e_i,e_j)}) = 0$.

Section 3 provides the details about the measure chosen to compute the score between two entities. This measure is applied to detect connectivity between entities and connectivity between Web resources (e.g. documents).

3 Approach

In this section, we present a method for computing the semantic connectivity between entities as well as corresponding Web documents. The process is divided into the following steps: (a) entity recognition and enrichment; (b) discovery of semantic associations between entities; (c) computation of semantic connectivity scores that express the relatedness between the entities.

3.1 Entity Recognition and Enrichment

The entity recognition and enrichment process extract rich, structured data about entities, such as locations, organizations or persons from unstructured Web resources. One fundamental goal is to, not only recognize named entities but, to enrich these with references to established reference datasets such as DBpedia or Freebase as means to disambiguate and expand entity descriptions.

Our approach currently applies two different methodologies: (1) gradual named entity recognition (NER) followed by subsequent enrichment, (2) integrated NER and enrichment. The first approach is currently exploited by the previously introduced ARCOMEM project and deploys GATE[7] components as a NER tool together with self-developed enrichment techniques using typed queries on DBpedia and Freebase [14]. While GATE extracts isolated typed entities, for instance an entity of type location with the label "Athens", enrichment is used to expand each entity with additional knowledge and to provide means for disambiguation.

The second approach exploits combined NER and disambiguation techniques which directly extract DBpedia and Freebase entities out of unstructured resources. As part of the current experiments proposed in this paper, we use a local deployment of the DBpedia Spotlight Web Service. While both approaches show particular advantages and disadvantages, a thorough evaluation with respect to precision and recall of retrieved entities is currently ongoing. However, since the focus of this paper is on the next two steps, our experiments use an evaluated set of extracted and enriched DBpedia entities.

3.2 Discovery of Semantic Associations between Entities

The second step of our approach aims at retrieving all paths with up to a maximum length between two given entities in the DBpedia graph. As this is a computationally expensive task, we adopted a pre-processing strategy, also used in [10], which computes the maximal connected subgraphs through a breadth-first search algorithm.

Instead of starting to find all the paths between two nodes, the algorithm verifies if both nodes belong to the same subgraph in the triple set. If the two nodes do not belong to the same subgraph, then a priori we know that no path with up to a predetermined maximum length exists between them. Otherwise, the process of finding all paths between two given nodes is initiated.

The maximum length of a path will be discussed in the next section. However, it is obvious that calculating long paths is expensive.

3.3 Semantic Connectivity and Document Relatedness Score

In order to compute the connectivity between two given enriched entities, we applied the Katz index proposed in [9] to calculate the relatedness of actors in a social network. This index takes into account the set of all paths between two nodes. The index also uses a damping factor β^l that is responsible for exponentially penalizing longer paths. The equation to compute the Katz index is as follows:

$$Katz(a,b) = \sum_{l=1}^{\tau} \beta^l \cdot |paths_{(a,b)}^{<l>}| \tag{1}$$

where $|paths_{(a,b)}^{<l>}|$ is the number of paths between a and b of length l and $0 < \beta \leq 1$ is a positive damping factor. The smaller this factor is, the smaller is the contribution of longer paths to the Katz index. Obviously, if the damping factor is 1, all paths will have

---

[7] `http://gate.ac.uk`

the same weight independently of the path length. In this work, we used $\beta = 0.5$ as our damping factor, since this value presented better results.

After computing the semantic connectivity score for a set of entities in a enriched dataset, a ranking of the most related entities is generated for each entity.

A major problem with the Katz index is that it is computationally expensive, since finding all paths between two nodes in not practical for large graphs. Thus, to overcome this problem, we set a threshold ($\tau = 4$) to the maximum path length between nodes, a decision that is backed up by the small world [21] phenomenon, which indicates that a pair of nodes is separated by a small number of connections. Thus, to compute all paths above this threshold would mean to add a constant factor for all indices.

One of the main applications of measuring entity connectivity is to discover *document relatedness*. In order to achieve such goal, we combine the results of the Katz index formula with entity co-occurrence scores. Thus, documents that contain the same entities receive an extra similarity bonus that would not be granted by the Katz index. The semantic relatedness score between documents is computed by the Eq. 2.

$$DRS(A,B) = \sum_{i \in A, j \in B, i \neq j} Katz(i,j) + \frac{|entity(A) \cap entity(B)|}{2} \tag{2}$$

where *entity*(*A*) and *entity*(*B*) denote the set of entities occurring in documents A and B, respectively.

4 Evaluation

In this section, we present the evaluation process to validate our approach. Our evaluation aims to assess the following criteria:

Computed connectivity between entities. Given the lack of benchmarks for validating entity connectivity, we rely on the wisdom of crowds to verify the relation between entities found by our semantic approach. Our assumption is that from the associations between terms (entity labels) suggested by Web users over time, a valid measure for connectivity emerges. In summary, two terms (that name entities) that co-occur to a high degree on the Web are considered related. With this “crowd-sourced” strategy, we exploit the wisdom of crowds to detect the co-occurrence of entities on the Web (See Section 4.2 for details). To assess the agreement of both approaches, we use a variation of the Kendall’s Tau method [3].

Validity of computed document relationships. This evaluation is fundamental to prove the importance of considering the semantic associations between entities. Furthermore, although our motivation examples show a very strict scenario, where linguistic techniques would fail, our evaluation intends to show that this strategy also can be useful to improve linguistic approaches in common datasets.

4.1 Dataset

In this section, we describe the characteristics of the two distinct datasets used for the evaluation process. The first dataset consists of 200 randomly selected articles from the

USAToday[8] news Web site. Each article contains a title and a summary of the whole textual content. The second dataset consists of randomly selected documents from Bibsonomy[9], a repository of research publications, annotated based on a folksonomy. To sample the data, we randomly selected 5 tags and gathered the Bibsonomy entries for each of these tags. In total we ended up with 213 documents (titles and abstracts).

The entity recognition and enrichment process (Section 3.1) extracted 399 unique entities from the USAToday corpus while the Bibsonomy corpus was annotated with 1118 unique entities. That resulted in rather large number of entity pairs, each requiring to compute an individual relatedness score. For example, in the case of the USAToday dataset, we obtained approximately 80,000 pairs of entities and over 600,000 pairs of entities for the Bibsonomy dataset. As reported in Section 3.3, each comparison of entity pairs returns several distance values that are used to compute the semantic relatedness score between documents in our two corpora.

4.2 Crowd-sourced Connectivity Score

To compare and assess our retrieved connectivity scores, we introduced a crowd-sourced relationship detection approach. For this purpose, the Bing search engine[10] was used to identify entity correlations based on term co-occurrence on the Web.

In order to estimate a co-occurrence score, a query is submitted to the Bing search engine, retrieving the total number of search results that contain the labels of the queried pair of entities in their text body. Note that Bing and other search engines return an approximation of the number of existing Web pages that contain the queried terms. In addition, since search terms are *untyped* strings as opposed to entities, we are aware that this approach might carry ambiguous and misleading results. However, we assume that a large number of pages indicates high connectivity and a small number of pages indicates low connectivity between the queried terms. Thus, given two entities a and b, the final score is estimated by the Eq. 3.

$$CrowdScore(a,b) = \frac{Log(count(ab))}{Log(count(a))} \cdot \frac{Log(count(ab))}{Log(count(b))} \tag{3}$$

where *count(a)* is the number of Web pages that contain entity a, *count(b)* is the number of Web pages that contain entity b and *count(ab)* is the number of Web pages that contain both entities a and b. It is important to note that *count(ab)* is always less than or equal to *count(a)* and less than or equal to *count(b)*. Hence, the final score is already normalized to $0 \leq CrowdScore(a,b) \leq 1$. This score will be used as our benchmark. Thus, we rely on the wisdom of crowds to validate our approach.

4.3 Entity Connectivity Evaluation & Document Relatedness Evaluation

In the first step, we aim to evaluate the entity rankings given by both methods - semantic and crowd-sourced - we used a variation of Kendall's tau method, which is used for measuring the similarity of the top k items in two ranked lists.

---

[8] `http://www.usatoday.com`

[9] `http://www.bibsonomy.org`

[10] `http://www.bing.com/developers/`

Table 1. Kendall tau and Precision between the semantic and the crowd-sourced entity rankings.

	k@2		k@5		k@10		k@20	
Dataset	Kendall tau	Precision	Kendall tau	Precision	Kendall tau	Precision	Kendall tau	Precision
USAToday	0.01	0.09	0.13	0.19	0.20	0.21	0.22	0.23
Bibsonomy	0.0010	0.0016	0.0069	0.0081	0.0102	0.0109	0.0204	0.0210

The second experiment evaluates the ability to find evidences of document relatedness. In this step, we compute for each pair of documents a semantic-based relatedness score and a crowd-sourced score. To compute the semantic relatedness score between documents, we used Eq. 2 proposed in Section 3.3. Similarly to Eq. 2, we defined a crowd-sourced score that uses the CrowdScore defined in Eq. 3 as follows:

$$DRC(A,B) = \sum_{i \in A, j \in B, i \neq j} CrowdScore(i,j) + \frac{|entity(A) \cap entity(B)|}{2} \tag{4}$$

where A and B are documents; i and j are entities contained in each document respectively. As we explained in Section 3.3, a different score is given for pairs of entities where $i = j$.

Based on Eqs. 2 and 4, a list of the most related documents for each given document is generated. In order to assess the precision of the generated ranked lists in the USAToday dataset, we performed a manual evaluation to validate the top 1 related document for each of the 200 existing documents using both methods (semantically and crowd-sourced generated). The results show the precision of the top one related item.

For the Bibsonomy dataset we used the tags of each document as a ground truth for document relatedness [12]. In this evaluation, two documents are considered related if they share a set of tags. For this evaluation, we assessed precision at different levels. As mentioned earlier, comparison with linguistic clustering techniques is not suited for the purpose of our evaluation, since it would only detect correlation of terms, while our approach also considers semantic relationships between terms (as part of extracted entity labels).

5 Results

Regarding the agreement of the semantically generated entity ranking against the crowd-sourced ranking generated by the given co-occurrence of the terms in Web pages, as explained in Section 4.3, we performed a variation of the Kendall tau rank correlation coefficient, together with precision measures at different levels (see results in Table 1).

The main reason for these rather low values is that the information captured by both relatedness strategies expresses different relationships. While the crowd-sourced one gives us the overall human perception of the relatedness between different entities, the semantic strategy provide us with actual underlying connections between the entities.

Regarding to the document relatedness evaluation, Table 2 shows the results regarding the manually assessed recommendations for both strategies, verifying the validity of the semantic entity-document score. On the USAToday dataset the success of both strategies perform quite good (over 76% for the semantic-based relatedness and over

Table 2. Precision @k for the documents relatedness recommendations in Bibsonomy dataset (left table). Precision @1 manually evaluated for the documents relatedness recommendations in USAToday dataset (right table).

Bibsonomy				
Precision	@1	@2	@5	@10
Semantic-based	0.78	0.86	0.82	0.76
Crowd-sourced	0.70	0.70	0.74	0.70

USAToday	
Precision	@1
Semantic-based	0.76
Crowd-sourced	0.65

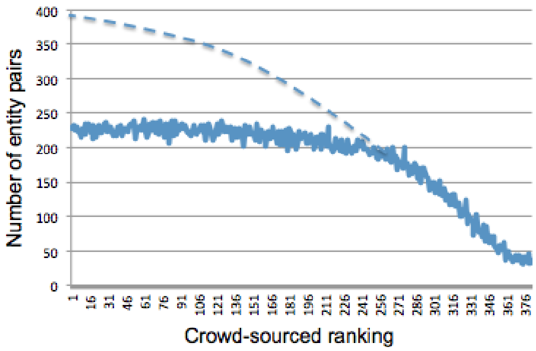


Fig. 1. The X-axis represents the ranking position x of entity pairs according to our crowd-sourced connectivity rankings. The Y-axis represent the number of entity pairs ranked at xth position that have a semantic relation according to our connectivity threshold.

65% for the crowd-sourced), giving us the proof of concept that, both strategies can be used for suggesting related items, even though the top related items are not the same.

For the Bibsonomy dataset we performed an automatic evaluation to access the precision of the document placement regarding the tag assignments. As explained in Section 4.3, we assumed the tag assignments as the ground truth for document relatedness [12]. Table 2 exposes the results for precision of the recommended documents considering the top k results. Both methods reached over 70% of performance that demonstrates their significant potential.

After computed all semantic and crowd-sourced scores between the pairs of entities, we obtained for each entity two ordered lists ranking all entities (m) according to each score. In Fig. 1, we represent data generated based on the USAToday dataset. The X-axis represents particular sets of entity-pairs ordered according to their connectivity ranking achieved based on the crowd-sourced activity ranking. The (x) value denotes all entity pairs (mx) which are ranked at the xth position in each particular entity ranking list. The Y-axis represent the number of entity pairs (nx) that have a semantic relation according to our semantic connectivity scores (solid line, $(\lambda(\delta^{<max(l)>}_{(e_i,e_j)}) > 0)$) within the particular set mx at xth ranking position.

Ideally, we expect that for every entity pair ranked at the top position (left on X-axis), would exist some semantic relation. The plot shows that for the top 1 crowd-sourced pairs, we found around 225 pairs that have such relation.

In this sense, the dotted line represents the ideal result. From these results, we can deduct that the pairs that are in between the area below the dotted line and above the solid line are most probably missing some semantic relation. Identifying the correct items that have some missing relations is the first step for the task of actually discovering which ones are the exactly missing relations. Complementary, by observing the missing semantic ranked pairs on the *X*-axis, we can identify which entities miss some relation given by the crowd-sourced. It is worth noting that since the 260th rank position in the *X*-axis, the behavior of the curve are in line with our expectations, i.e., the lower the correlation between crowd-sourced, the lower is the semantic connectivity.

As for a qualitative analysis of the document relatedness evaluation, we picked up a document (i) from the USAToday corpus and its most related document (ii) according to our semantic-based approach. The underlined terms refer to the recognized entities in each document derived from the entity recognition and enrichment process (see Section 3.1).

(i) The Charlotte Bobcats could go from the NBA's worst team to its best bargain.

(ii) The New York Knicks got the big-game performances they desperately needed from Carmelo Anthony and Amar'e Stoudemire to beat the Miami Heat.

Although both documents are related to basketball, a linguistic approach would fail to point out both documents as related. First, both documents have too short descriptions, which make it harder for a linguistic approach to detect their similarity. Second, in this particular case, there are no significant common words between the documents. However, by applying our semantic-based approach, it is possible to measure a score of connectivity between both documents. For example, once the term *Charlotte Bobcats* was enriched by the entity `http://dbpedia.org/resource/Charlotte_Bobcats` in the document (i) and the term *New York Knicks* was enriched by the entity `http://dbpedia.org/resource/New_York_Knicks` in the document (ii), a semantic score is assigned to each pair of entities found to generate an overall score of connectivity between both documents.

6 Related Work

The approach of applying actor/network theory to data graphs has been discussed by Kaldoudi et al. [8]. Graph summarization is a very interesting approach to exploit semantic knowledge in annotated graphs. Thor et al. [19] exploited this technique for link prediction between genes in the area of Life Sciences. Their approach relies on the fact that the graph summarization techniques create compact representations of the original graph adopting some criteria for the creation, correction and deletion of edges and for grouping nodes. Thus, a prediction function ranks the most potential edges and then suggests possible links between two given genes.

Another approach to identify potential links between nodes is presented by Potamias et al. [13], where they describe an algorithm based on Dijkstra's shortest path along with random walks in probabilistic graphs to define distance functions that identifies the k closest nodes from a given source node. Lehmann et al. [10] introduces the RelFinder that is able to show relationships between two different objects in DBpedia. Their approach is based on the breadth-first search algorithm, which is responsible for finding all related objects in the tripleset. Then, the information gathered is stored in a relational database for further querying and visualization. In this work, we use the RelFinder approach to exploit the relationship between objects (see Section 3.2). Contrasting with RelFinder, Seo et al. [15] proposed the OntoRelFinder that uses a RDF Schema for finding the relationships between two objects through its class relationships.

An interesting work in social networks is also presented by Leskovec et al. [11]. Their technique suggests positive and negative relationships between people in a social network. The notion of negative and positive relationships is also addressed in our method, but taking into account the length of the paths, as aforementioned. Similarly, Xiang et al. [22] present a work based on the homophily principle (i.e., people tend to associate and interact with people with similar characteristics) to estimate relationship strength between people. For this, they present an unsupervised model that takes into account the shared attributes and interactions between individuals in a social network. This approach meets our assumptions that the closer two objects are, the higher is the proximity between them.

Finding semantic associations between two given objects is also discussed in the context of ontology matching [6, 20, 23]. In our case, hub ontologies could also be used to infer missing relationships into another ontology.

Contrasting with the approaches just outlined, we combine different techniques to uncover relationships between disparate entities, which allows us to exploit the relationships between entities to identify correlated Web resources.

7 Discussion and Outlook

We have presented a general-purpose approach to discover relationships between Web resources based on the relationships between extracted entities together with an evaluation and discussion of experimental results. We found that, uncovering relationships between data entities helps to detect correlations of documents that, a priori, linguistic approaches would not reveal. Linguistic methods are based on the co-occurrence of words in a set of documents, while our semantic-based approach relies on semantic relations between entities as represented in reference datasets. A hybrid approach would overcome this deficiency. However, in cases where extracted entities have to be matched, term frequency or linguistic similarity-based approaches cannot be applied. An interesting application of our work lies in document and data clustering which can be exploited, for instance, for entity based document recommenders.

During our evaluation experiments, we achieved an average of 80% of precision for the Bibsonomy dataset when suggesting the most related documents given one document (top 1, top 2, top 5, top 10), while for the USAToday dataset we achieved 0.76% of precision. We also presented a crowd-sourced strategy that takes into account the co-occurrence of entities in Web searches, thus relying on the wisdom of crowds. This approach achieved an average of 71% of precision for the Bibsonomy dataset, while the USAToday presented 65% of precision. This leads to the conclusion that both produce fairly good indicators for document relatedness.

Although both approaches have achieved good results, an evaluation based on the Kendall's tau rank correlation has shown that both differ in the relationships they uncovered. Naturally, the numbers presented by the Kendall's tau evaluation are subjected to noise caused by *misannotated* entities during the NER/enrichment process and the approximated values given by the search engine. Nevertheless, we believe that these proposed evaluations are the first step to identify missing connections in a semantically enriched dataset. Finally, we deduct that each strategy is complementary to each other. Semantically deducted relations are able to find connections between entities that do not necessarily often co-occur in contrast to the crowd-sourced analysis based on co-occurrence.

The main issues faced during the experimental work were the low performance and accuracy of the NER tools at hand, and high computational demands when applying our relatedness computation to larger amounts of data. That restricted our experiments to a limited dataset. Moreover, one of the key weaknesses of the Katz index in the context of our work is the fact that it treats all edges equally. Thus, when applying it to Linked Data graphs, valuable semantics about the meaning of each edge (i.e., property) is not considered during the relatedness computation. We are currently investigating ways to extend the Katz index by distinguishing between different property types. Hence, future work will plan to (a) apply weights to different path types between the entities according to the semantics of the properties they represent in order to provide a more refined score; and (b) investigate means to combine our two complementary relationship discovery approaches.

8 Acknowledgement

This work has been partially supported by CAPES (Process n^o 9404-11-2) and the European Union Seventh Framework Programme (FP7/2007-2013), grant agreement No 270239 (ARCOMEM), CNPq grants 301497/2006-0 and 475717/2011-2 and FAPERJ grants E-26/103.070/2011.

References

1. Anyanwu, K., Sheth, A.: p-queries: enabling querying for semantic associations on the semantic web. In: Proceedings of the 12th international conference on World Wide Web. pp. 690 – 699. ACM Press New York, NY, USA, Budapest, Hungary (2003)
2. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. International Journal on Semantic Web and Information Systems (IJSWIS) 5(3), 1–22 (2009)
3. Carmel, D., Cohen, D., Fagin, R., Farchi, E., Herscovici, M., Maarek, Y.S., Soffer, A.: Static index pruning for information retrieval systems. In: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 43–50. SIGIR '01, ACM, New York, NY, USA (2001), `http://doi.acm.org/10.1145/383952.383958`
4. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A framework and graphical development environment for robust NLP tools and applications. In: Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (2002)

5. Dietze, S., Yu, H., Giordano, D., Kaldoudi, E., Dovrolis, N., Taibi, D.: Linked education: interlinking educational resources and the web of data. In: Proceedings of the 27th ACM Symposium On Applied Computing, Special Track on Semantic Web and Applications. SAC '12, ACM, New York, NY, USA (2012)
6. Groß, A., Hartung, M., Kirsten, T., Rahm, E.: Mapping Composition for Matching Large Life Science Ontologies. In: Proceedings of the 2nd International Conference on Biomedical Ontology. ICBO 2011 (2011)
7. Halpin, H., Hayes, P.J., McCusker, J.P., McGuinness, D.L., Thompson, H.S.: When owl: sameas isn't the same: an analysis of identity in linked data. In: Proc. of the 9th International Semantic Web Conference,Vol. Part I. pp. 305–320. Berlin, Heidelberg (2010), `http://dl.acm.org/citation.cfm?id=1940281.1940302`
8. Kaldoudi, E., Dovrolis, N., Dietze, S.: Information organization on the internet based on heterogeneous social networks. In: Proceedings of the 29th ACM international conference on Design of communication. pp. 107–114. SIGDOC '11, ACM, New York, NY, USA (2011), `http://doi.acm.org/10.1145/2038476.2038496`
9. Katz, L.: A new status index derived from sociometric analysis. Psychometrika 18(1), 39–43 (March 1953), `http://ideas.repec.org/a/spr/psycho/v18y1953i1p39-43.html`
10. Lehmann, J., Schppel, J., Auer, S.: Discovering unknown connections - the DBpedia relationship finder. In: Proceedings of 1st Conference on Social Semantic Web. Leipzig (CSSW07), 24.-28. September. Lecture Notes in Informatics (LNI), vol. P-113 of GI-Edition. Bonner Kllen Verlag (September 2007), `http://www.informatik.uni-leipzig.de/~auer/publication/relfinder.pdf`
11. Leskovec, J., Huttenlocher, D.P., Kleinberg, J.M.: Predicting positive and negative links in online social networks. CoRR abs/1003.2429 (2010), `http://dblp.uni-trier.de/db/journals/corr/corr1003.html#abs-1003-2429`
12. Peters, I., Haustein, S., Terliesner, J.: Crowdsourcing in article evaluation. In: Proceedings of the 3rd ACM International Conf. on Web Science. pp. 1–4. Koblenz, Germany (2011), `http://journal.webscience.org/487/`
13. Potamias, M., Bonchi, F., Gionis, A., Kollios, G.: k-nearest neighbors in uncertain graphs. PVLDB 3(1), 997–1008 (2010), `http://dblp.uni-trier.de/db/journals/pvldb/pvldb3.html#PotamiasBGK10`
14. Risse, T., Dietze, S., Peters, W., Doka, K., Stavrakas, Y., Senellart, P.: Exploiting the social and semantic web for guided web archiving. In: Proceedings of the International Conference on Theory and Practice of Digital Libraries 2012. TPDL '12, Springer LNCS (2012)
15. Seo, D., Koo, H., Lee, S., Kim, P., Jung, H., Sung, W.K.: Efficient finding relationship between individuals in a mass ontology database. In: Kim, T.H., Adeli, H., Ma, J., Fang, W.C., Kang, B.H., Park, B., Sandnes, F.E., Lee, K.C. (eds.) FGIT-UNESST, vol. 264. pp. 281–286. Communications in Computer and Information Science, Springer (2011), `http://dblp.uni-trier.de/db/conf/fgit/unesst2011.html#SeoKLKJS11`
16. Sheth, A., Aleman-Meza, B., Arpinar, I.B., Halaschek-Wiener, C., Ramakrishnan, C., Bertram, Y.W.C., Avant, D., Arpinar, F.S., Anyanwu, K., Kochut, K.: Semantic association identication and knowledge discovery for national security applications. Journal of Database Management 16(1), 3353 (2005)
17. Sheth, A.P., Ramakrishnan, C.: Relationship web: Blazing semantic trails between web resources. IEEE Internet Computing 11(4), 77–81 (2007), `http://dblp.uni-trier.de/db/journals/internet/internet11.html#ShethR07`
18. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: A core of semantic knowledge. In: 16th international World Wide Web conference. ACM Press, New York, NY, USA (2007)
19. Thor, A., Anderson, P., Raschid, L., Navlakha, S., Saha, B., Khuller, S., Zhang, X.N.: Link prediction for annotation graphs using graph summarization. In: 10th International Confer-

ence on The Semantic Web, Vol. Part I. pp. 714–729. ISWC'11, Berlin, Heidelberg (2011), `http://dl.acm.org/citation.cfm?id=2063016.2063062`

20. Vidal, V.M.P., de Macedo, J.A.F., Pinheiro, J.C., Casanova, M.A., Porto, F.: Query processing in a mediator based framework for linked data integration. IJBDCN 7(2), 29–47 (2011), `http://dblp.uni-trier.de/db/journals/ijbdcn/ijbdcn7.html#VidalMPCP11`
21. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature 393(6684), 440–442 (Jun 1998), `http://dx.doi.org/10.1038/30918`
22. Xiang, R., Neville, J., Rogati, M.: Modeling relationship strength in online social networks. In: Rappa, M., Jones, P., Freire, J., Chakrabarti, S. (eds.) WWW. pp. 981–990. ACM (2010), `http://dblp.uni-trier.de/db/conf/www/www2010.html#XiangNR10`
23. Xu, L., Embley, D.W.: Discovering direct and indirect matches for schema elements. In: DASFAA. pp. 39–46. IEEE Computer Society (2003), `http://dblp.uni-trier.de/db/conf/dasfaa/dasfaa2003.html#XuE03`

Entity Extraction:
From Unstructured Text to DBpedia RDF Triples

Peter Exner and Pierre Nugues

Department of Computer science
Lund University
`peter.exner@cs.lth.se`
`pierre.nugues@cs.lth.se`

Abstract. In this paper, we describe an end-to-end system that automatically extracts RDF triples describing entity relations and properties from unstructured text. This system is based on a pipeline of text processing modules that includes a semantic parser and a coreference solver. By using coreference chains, we group entity actions and properties described in different sentences and convert them into entity triples. We applied our system to over 114,000 Wikipedia articles and we could extract more than 1,000,000 triples. Using an ontology-mapping system that we bootstrapped using existing DBpedia triples, we mapped 189,000 extracted triples onto the DBpedia namespace. These extracted entities are available online in the N-Triple format[1].

1 Introduction

By using the structured and semi-structured information from Wikipedia, DBpedia [1] has created very large amounts of linked data and is one the most significant achievements of the Semantic Web initiative. Datasets from DBpedia are used in a wide range of applications such as faceted search, model training for information extraction, etc.

DBpedia focuses on extracting structured information from Wikipedia articles, such as infobox templates and categorization information. However, the unstructured text of the articles is left unprocessed. Some recent projects have attempted to use this text content to extend the DBpedia triple base. Examples include iPopulator [2] that populates incomplete infoboxes with attribute values it identifies from the article text, while two recent systems, LODifier [3] and KnowledgeStore [4], extract semantic information from the text. LODifier creates RDF triples based on WordNet URIs while KnowledgeStore uses its own ontology. Nonetheless, these systems show limitations in the form of preexisting infobox templates or data structures that are not fully compliant with the DBpedia namespace.

In this paper, we introduce a framework to carry out an end-to-end extraction of DBpedia triples from unstructured text. Similarly to LODifier and KnowledgeStore, our framework is based on entities and identifies predicate–argument structures using a generic semantic processing pipeline. However, instead of recreating new semantic structures, we integrate the DBpedia property ontology and therefore make the reuse

---

[1] `http://semantica.cs.lth.se/`

and extension of the DBpedia dataset much easier. Starting from the DBpedia dataset, we link the triples we extract from the text to the existing DBpedia ontology, while going beyond the existing infobox templates. Applications already using DBpedia would then benefit from a richer triple store.

We applied our system to over 114,000 Wikipedia randomly selected articles and we could extract more than 1,000,000 triples. Using the ontology-mapping system that we bootstrapped using existing DBpedia triples, we mapped 189,000 extracted triples onto the DBpedia namespace. Interestingly, we could rediscover from the text 15,067 triples already existing in the DBpedia dataset. We evaluated our framework on a sample of 200 sentences and we report a F1 score of 66.3% on the mapped triples.

2 Related Work

The extraction of relational facts from plain text has long been of interest in information extraction research. The key issue in relation extraction is to balance the trade-off between high precision, recall, and scalability. With the emergence of the Semantic Web and numerous ontologies, data integration has become an additional challenge.

There has been a considerable amount of research on semi-supervised [5–7] methods using bootstrapping techniques together with initial seed relations to create extraction patterns. Unsupervised approaches [8, 9] have contributed further improvements by not requiring hand-labeled data. These approaches have successfully answered scalability and precision factors, when applied on web-scale corpora. The challenge of ontology and data integration has been addressed by [10].

Due to concerns on scaling, the use of syntactic or semantic relation extraction techniques in relation extraction has been relatively sparse. Few systems carry out a complete analysis of the source documents using coreference resolution or discourse analysis to extract all statements. Exceptions include LODifier [3] and KnowledgeStore [4], that have extracted semantic information and applied coreference resolution. However, the entities extracted by these systems have not been integrated to a single homogenous ontology.

In contrast to these approaches, we suggest an end-to-end system, that extracts all the entity relations from plain text and attempts to map the entities onto the DBpedia namespace. We balance precision and recall by employing a combination of NLP tools, including semantic parsing, coreference resolution, and named entity linking. Scalability issues are handled by parallelizing the tasks on a cluster of computers. Furthermore, we propose an ontology mapping method that bootstraps learning from existing triples from the DBpedia dataset.

3 System Architecture

The architecture of our system is a pipeline that takes the Wikipedia articles as input and produces entities in the form of DBpedia RDF triples. As main features, the system includes a generic semantic processing component based on a semantic role labeler (SRL) to discover relations in text, an automatic learning of ontology mappings to link the extracted triples to the DBpedia namespace, and an algorithm to rank named entity

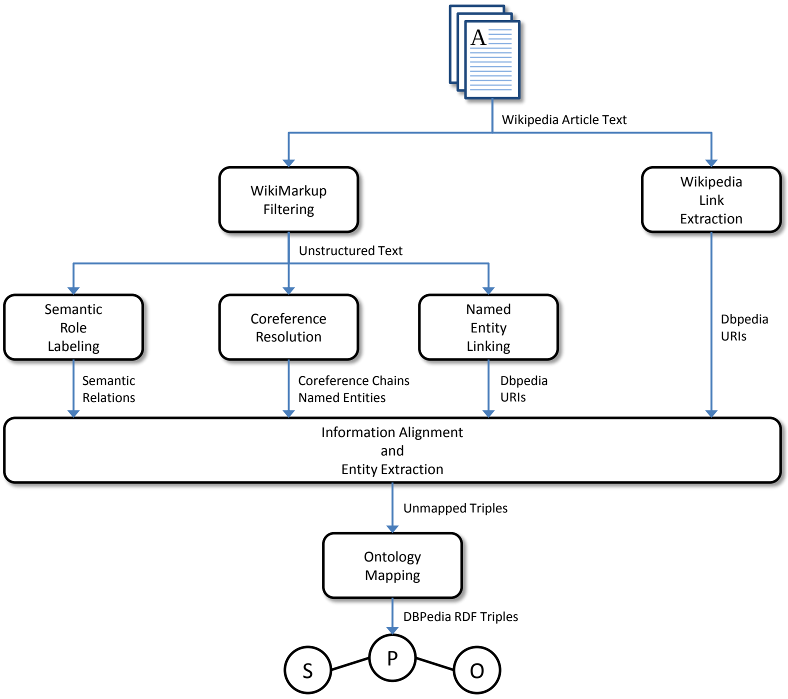


Fig. 1. Overview of the entity extraction pipeline.

links (NEL) found in coreference chains in order to discover representative mentions. In total, the end-to-end processing of Wikipedia article text consists of seven modules (Figure 1):

1. A *WikiMarkup filtering* module that removes the Wikimedia markup, providing the plain text of the articles to the subsequent modules;
2. A *Wikipedia link extractor* that extracts Wikipedia links from the articles;
3. A *semantic parsing* module, Athena [11], a framework for large-scale semantic parsing of text written in natural language;
4. A *coreference resolution* module that detects and links coreferring mentions in text;
5. A *mention-to-entity linking* module that links mentions to a corresponding DBpedia URI;
6. An *information aligning and entity extracting* module that aligns the output from top-level modules and extracted entities in the form of triples.
7. An *ontology mapping* module that carries out the final mapping of predicates from the Propbank nomenclature onto the DBpedia namespace.

4 Processing of Wikipedia Article Text

WikiMarkup Filtering. Prior to any analysis, the text must be filtered. This is an essential step that seeks to remove annotations and markups without affecting the running text. Without this step, subsequent modules would fail in their analysis and lead to erroneous extractions.

Wikipedia articles are composed of text written in natural language annotated with a special markup called wikitext or wiki markup. It is a simple markup language that allows among other things the annotation of categories, templates, and hyperlinking to other Wikipedia articles. Wikipedia also allows the use of common HTML tags.

By filtering Wikipedia text, we aim at removing all annotations, sections that contain only links and references, and keeping only the running text. This process is difficult since the HTML syntax is often invalid. The most common errors are tags that are left unclosed or are incorrectly nested.

Wikipedia Link Extraction. During the Wikipedia link extraction, we extract and preserve the original links along with their corresponding mentions in the article. In addition to extracting the links annotated by the article authors, we make the assumption that the first noun phrase in the first sentence corresponds to the article link. The rationale behind it is that the longest coreference chain in the article often starts with this first mention.

The direct correspondence between Wikipedia articles and DBpedia resources allows us to map Wikipedia links onto their corresponding DBpedia URI by simply adding the DBpedia namespace.

Semantic Parsing. Frame semantics [12] is a linguistic theory that assumes that the meaning of a sentence is represented by a set of predicates and arguments. The Proposition Bank [13] is a project that applied this theory to annotate corpora with predicate-argument structures. For each predicate, Propbank identifies up to six possible core arguments denoted *A0*, *A1*, ..., and *A5* that go beyond the traditional annotation of subjects and objects. Propbank also includes modifiers of predicates, such as temporal and location adjuncts. These roles are instrumental in performing the extraction of entities as they allow the identification of properties containing temporal and locational data with high precision.

We use the Athena framework created for parallel semantic parsing of unstructured text. At its core, the system uses a high-performance multilingual semantic role labeler that obtained top scores in the CONLL-2009 shared task [14, 15].

Coreference Resolution. A coreference resolver creates chains of coreferring mentions by discovering and linking anaphoric phrases to their antecedents. We used a coreference solver, included in the Stanford CoreNLP package [16, 17], to link mentions of entities in the different parts of text. This allows us to group entity actions and properties described in different sentences. CoreNLP uses a pipeline of tokenizers, part-of-speech tagger, named entity recognizer, syntactic parser, and coreference solver to annotate unstructured text. In addition to coreference annotation, we store the named entity classification created by the pipeline. The named entity classes are used to filter named entity links having a conflicting ontology classification.

Named Entity Linking. An important step in entity extraction is the grounding of named entities to unique identifiers. In most articles, only the first mention of a named entity is annotated with a corresponding Wikipedia link; subsequent mentions are often left unannotated. Wikifier [18] is a named entity linking system that annotates unstructured text with Wikipedia links. By applying Wikifier, we can link unannotated named entities in the Wikipedia articles to a corresponding DBpedia URI.

Ontology Mapping. During semantic parsing, the sentences are annotated with predicate–argument structures called rolesets. As dictionary, the parser uses PropBank that defines more than 7,000 rolesets. Propbank associates each predicate with a set of senses, for instance *bear* has six senses denoted *bear.01*, *bear.02*, ..., *bear.06*. Finally, each predicate-sense has a set of core arguments that differ with each roleset. For example, *bear.02* has two core arguments: *A0*, the mother, and *A1*, the child. Considering only the core roles, this amounts to more than 20,000 roles.

The objective of ontology mapping is to map the predicate and argument roles from PropBank onto DBpedia properties. We perform this final step to create the DBpedia RDF triples. Figure 2 shows an example of end-to-end processing to DBpedia RDF triples of the sentences: *Luc Besson (born 18 March 1959) is a French film director, writer and producer. Besson was born in Paris to parents who were both Club Med scuba diving instructors.*

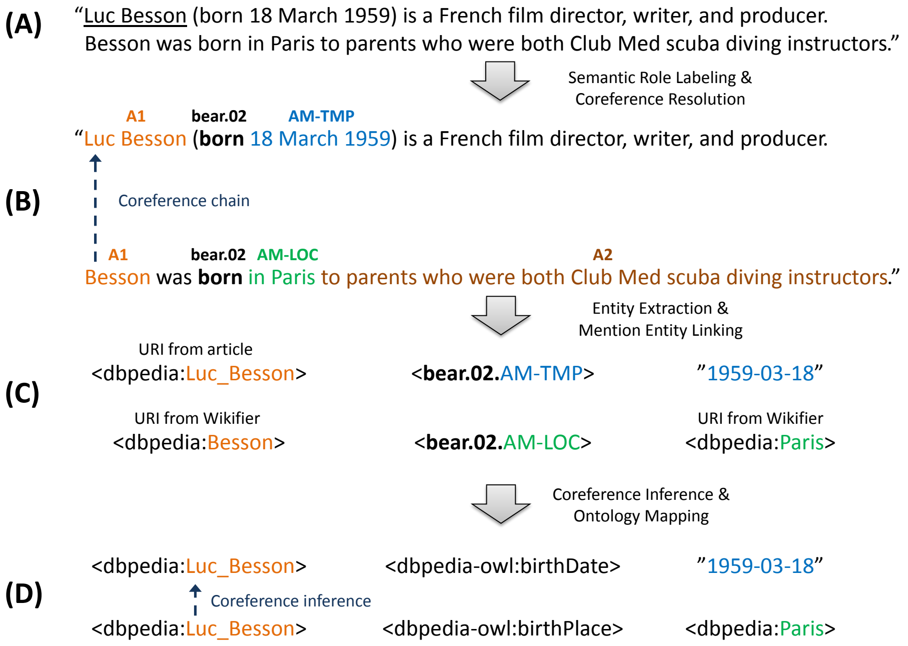


Fig. 2. An ideal conversion from text to the DBpedia RDF triples: (A) The input sentences. (B) The sentences after semantic parsing and coreference resolution. (C) Entity extraction. (D) Ontology mapping.

5 Entity Extraction

The arguments created during semantic parsing are searched in order to find named entity links corresponding to RDF subjects and objects. This process uses the mentions discovered by the coreference solver, Wikipedia links predicted by Wikifier, and

Wikipedia links extracted from the article. In order to keep the task tractable, we have limited the entities to those found in DBpedia and we do not introduce new named entities to the DBpedia ontology.

RDF Subjects. PropBank uses the *A0* label as the argument describing agents, causers, or experiencers, while arguments labeled as *A1* describe entities undergoing a state of change or being affected by an action. In both cases, arguments labeled *A0* or *A1* can be considered containing RDF subjects and are consequently searched for named entity links. Arguments labeled *A0* are searched first, arguments labeled *A1* are only searched if a named entity link wasn't discovered in the preceding arguments.

RDF Objects. Following the subject extraction, the remaining arguments are examined to discover potential objects. The core arguments and two auxiliary arguments, temporal *AM-TMP* and location *AM-LOC*, are searched. The extracted data types can be categorized as following: Named entity links expressed as DBpedia URIs, dates and years, integers, and strings. We search date expressions in the temporal arguments *AM-TMP* using regular expressions. By using seven common date patterns, we are able to extract a large amount of date and year expressions. We associate the location arguments *AM-LOC* to named entity links representing places. These links are extracted only if they are classified as *dbpedia-owl:Place* by the DBpedia ontology.

Named Entity Link Ranking and Selection. During the search of RDF subject and objects, we search and select candidate named entity links in the following order:

1. Wikipedia links, converted to DBpedia URIs. We consider named entity links extracted from the article as being most trustworthy.
2. Wikifier-predicted Wikipedia links, converted to DBpedia URIs, and having a DBpedia ontology class matching the predicted named entity class. A predicted named entity link is chosen only in the case when an extracted Wikipedia link isn't given. Furthermore, predicted links are pruned if their DBpedia ontology class doesn't match the named entity class predicted by the Stanford coreference solver.
3. Coreference mentions; the most representative named entity link (according to the score described in section Using Coreference Chains) in the coreference chain is selected. We consider named entities inferred through coreference chains as the least trustworthy and select them only if an extracted or predicted named entity link is not given. A mention placed in the wrong coreference chain will be considered as an incorrect named entity link; a situation which Wikifier can rectify with higher precision.

Using Coreference Chains. Coreference chains are used to propagate named entity links to arguments having neither an extracted nor a predicted named entity link. This situation arises most commonly for arguments consisting of a single pronoun. Before propagation takes place, we determine the most representative named entity link in the coreference chain using a ranking and scoring system:

- Extracted named entity links are always selected over predicted links.

- A score of +2 is given to a named entity link if it has a DBpedia ontology class matching the predicted named entity class.
- The score is increased by the number of tokens of the named entity minus 1.
- If a tie is given between equally scoring named entity links, the link closest to the top of the chain is selected.

We derived the set of scoring rules by performing an empirical examination of coreference chains. We observed that coreference chains representing people, often started with a mention containing the full name, followed by single-token mentions having only the first or last name. The named entity links of single-token mentions, as predicted by Wikifier, often incorrectly pointed to either a place or a family. By rewarding named entity links having multiple tokens and matching ontology classes, we filtered these incorrect links. Table 1 shows an example, where the mention *Robert Alton*, a person name, is given the highest score due to matching entity classes and token length. Although the mention *Alton* refers to the same entity and belongs to the coreference chain, an incorrect named entity link to a city (Alton, Illinois) has been predicted. Given our previous rules, the predicted named entity link is discarded due to a mismatch with the predicted named entity class. The correct named entity link is thus resolved by propagating the link through the coreference chain.

Unmapped Triple Generation. Given a set of extracted RDF subjects and objects, we create binary relations from n-ary predicate–argument relations by a combinatorial generation. We discover negative relations by searching the argument roles for *AM-NEG*; these are then discarded.

Mention	NE class	NE link	DBpedia ontology class	Score
Alton	Person	dbpedia:Alton,_Illinois	dbpedia-owl:Place	0
Robert Alton	Person	dbpedia:Robert_Alton	dbpedia-owl:Person	3
He				0

Table 1. Example showing how scoring resolves the most representative predicted named entity link in a coreference chain. NE stands for named entity. The NE class is obtained from the NE recognition module in the Stanford CoreNLP package. The NE link is predicted by Wikifier.

6 Ontology Mapping

The final step in extracting DBpedia RDF triples from text concerns the mapping of predicates onto the DBpedia namespace. The unmapped extracted triples have predicates described using the Propbank dictionary. The predicates together with their sense and unique set of argument roles comprise more than 20,000 different roles. With ontology mapping, our goal is to map the resulting triples onto a more general roleset described by 1,650 DBpedia properties.

Since the manual mapping of such a large amount of roles is a requiring task, we wished to perform the it automatically. Our approach was to bootstrap the learning of ontology mappings by matching the subject and object of the extracted triples onto existing triples in the DBpedia dataset. Generic mappings are created and reused by generalizing DBpedia URIs, found in subjects and objects, to 43 top-level DBpedia ontology classes. The learning process consists of the following steps (Figure 3):

1. The subject and object of the extracted triples are matched exactly to existing triples in the DBpedia dataset.
2. From the matching set of triples, links between Propbank roles and DBpedia properties are created. The mappings with the highest count are stored.
3. The subject and object of the extracted triples that contain DBpedia URIs are generalized to 43 top-level DBpedia ontology classes. Objects containing strings, dates and numbers are generalized to the categories: String, Date, and Number respectively.

As an example, consider the following sentence:

Besson married Milla Jovovich on 14 December 1997.

We extract the triple:

<dbpedia:Luc_Besson> <marry.01.A1> <dbpedia:Milla_Jovovich>

and match to the existing triple in the DBpedia dataset:

<dbpedia:Luc_Besson> <dbpedia-owl:spouse> <dbpedia:Milla_Jovovich>

and finally generalize the subjects and objects to create the mapping:

<dbpedia-owl:Person> <marry.01.A1> <dbpedia-owl:Person>
maps to:
<dbpedia-owl:spouse>

Table 2 shows five of the most frequent mappings learned during the bootstrapping process. Most systems express mappings as alignments between single entities belonging to different ontologies. In addition, we also retain a generalized form of the related subject and object entities in such alignments. Together with the predicate sense and object argument role, we use them to express a more detailed mapping. By including the generalized object and its argument role in our mappings, we can differentiate between different domains for a certain predicate, such as between a birth place and a birth date.

When a new sentence, describing the same relation, is encountered for which there is no existing DBpedia triple we can perform our ontology mapping. Mappings learned from one entity can thus be used on other entities as more descriptive entities create mappings that are reused on entities with fewer properties. As an example, consider the following sentence:

On April 30, 2008, Carey married Cannon at her private estate...

with the extracted triple:

<dbpedia:Mariah_Carey> <marry.01.A1> <dbpedia:Nick_Cannon>

generalized to:

<dbpedia-owl:Person> <marry.01.A1> <dbpedia-owl:Person>

we can apply our previously learned mapping and infer that:

<dbpedia:Mariah_Carey> <dbpedia-owl:spouse> <dbpedia:Nick_Cannon>

that is not present in DBpedia.

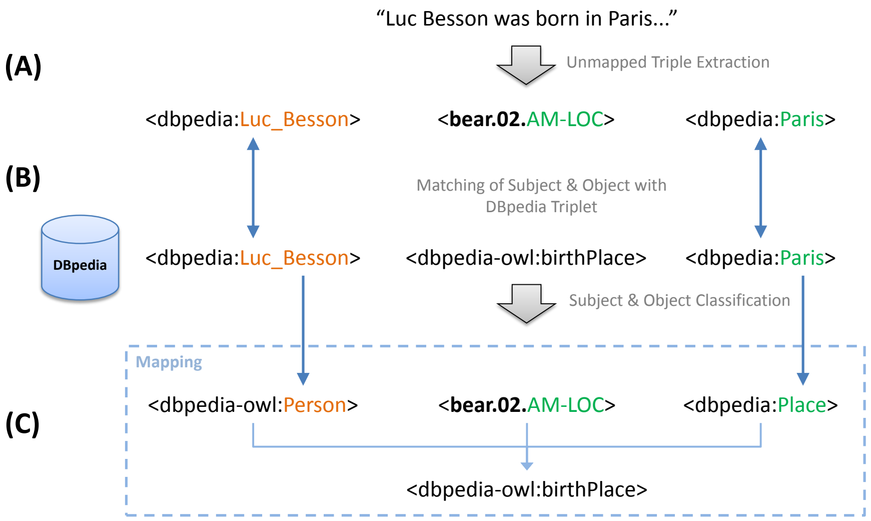


Fig. 3. (A) An unmapped triple is extracted from the sentence. (B) The extracted triple is matched to an existing DBpedia triple. (C) A mapping is created by linking the predicates between the two different namespaces and generalizing the subject and object.

Subject	Predicate	Object	Mapping
dbpedia-owl:Person	bear.02.AM-LOC	dbpedia-owl:Place	dbpedia-owl:birthPlace
dbpedia-owl:Person	bear.02.AM-TMP	xsd:date	dbpedia-owl:birthDate
dbpedia-owl:Person	marry.01.A1	dbpedia-owl:Person	dbpedia-owl:spouse
dbpedia-owl:Organisation	locate.01.AM-LOC	dbpedia-owl:Place	dbpedia-owl:city
dbpedia-owl:Organisation	establish.01.AM-TMP	xsd:integer	dbpedia-owl:foundingYear

Table 2. Five of the most frequent ontology mappings learned through bootstrapping.

7 Experimental Results

The aim of the evaluation is to answer the question of how much information in the form of entity relation triples can be extracted from sentences. We also wish to evaluate the quality of the extracted triples. Since there is no gold standard annotation of entities found in the main text of Wikipedia articles, we performed the evaluation by manually analyzing 200 randomly sampled sentences from different articles. Sampled sentences are examined for relevant subject-predicate-object triples and compared to the corresponding retrieved triples. We computed the precision, recall, and F1 scores, and in the occurrence of an extraction error, we made a note of the originating source.

We evaluated the attributes of each triple in a strict sense: Each extracted attribute must exactly match the corresponding attribute in the sentence. For instance, in evaluating the birthplace of a person, if a sentence states a city as the location, we only consider an extracted DBpedia link to the city as correct. In contrast, if the extracted link refers to a more generalized toponym, such as region or country, we mark the extracted object as erroneous.

In total, we processed 114,895 randomly selected articles amounting to 2,156,574 sentences. The articles were processed in approximately 5 days on a cluster of 10 machines. Table 3, left, shows the number of processed articles categorized by DBpedia ontology classes. From the processed articles, we extracted a total of 1,023,316 triples, of which 189,610 triples were mapped to the DBpedia ontology. The unmapped triples differ in having the predicate localized to the Propbank namespace. In Table 3, right, we can see that from the 189,610 extracted triples, 15,067 triples already exist in the DBpedia dataset. This means that our framework introduced 174,543 new triples to the DBpedia namespace. Almost 3% of the extracted triples are duplicates, the majority of these are triples repeated only once. Since a statement with the same meaning can occur in more than one article, we consider these occurrences natural. In comparing the number of extracted triples to the number of processed sentences, we find that roughly every second sentence yields one extracted triple. In comparison to the number of processed articles, we extracted nearly 9 triples per article.

The extracted mapped triples reached a F1 score of 66.3%, a precision of 74.3%, and a recall of 59.9%. The largest source of errors came from predicates, 46.4%, followed by subjects, 27.2%, and objects, 26.4%.

Based on post-mortem analysis of the evaluated triples, we find that reasons for the extraction errors can be attributed to the following causes:

- An incorrect mapping from the Propbank predicate-argument roles to the DBpedia ontology properties.
- A new entity is detected, that has previously not been introduced to the DBpedia datasets and therefore lacks a corresponding DBpedia URI.
- The wrong URI is predicted for an entity and cannot be resolved or corrected by the scoring algorithm.
- A mention is placed in the incorrect coreference chain by the coreference solver.

The majority of errors stem from erroneous ontology mappings. We believe that ontology mapping can be improved by using a more fine grained approach to the subject-object generalization. Currently, we categorize subjects and objects to 43 top-level DBpedia ontology classes out of 320 possible classes. In addition, increasing the amount of bootstrapping data used during learning can be done by utilizing links to other datasets, such as LinkedMDB[2]. We also believe that the current rule-based mapping system could be replaced by a more capable system based on machine learning.

The linking of mentions to DBpedia URIs was also found to be a major source for errors. We believe that retraining Wikifier using a more current Wikipedia dump may improve named entity linking.

8 Conclusions and Future Work

In this paper, we described an end-to-end framework for extracting DBpedia RDF triples from unstructured text. Using this framework, we processed more than 114,000

---

[2] `http://www.linkedmdb.org/`

English Wikipedia	
Persons	32,423
Places	63,503
Organisations	18,969
Total Articles	114,895
Sentences	2,156,574

Type	Count
DBpedia Mapped Triples (of which 15,067 already exist in DBpedia)	189,610
Unmapped Triples	833,706
Total	1,023,316

Table 3. Left table: An overview of entity extraction statistics. **Right table**: The number of extracted triples grouped by triple type.

articles from the English edition of Wikipedia. We extracted over 1,000,000 triples that we stored as N-Triples. We explored a method for creating ontology mappings of predicates between the Proposition Bank and the DBpedia namespace. By bootstrapping the learning of mappings through the alignment of extracted triples with triples in the DBpedia dataset, we mapped 189,000 triples to the DBpedia namespace. We evaluated the mapped triples on a randomly selected sample of 200 sentences and we report a F1 score of 66.3%.

The largest source of errors stemmed from incorrect mappings. For instance, a mapping describing a person receiving a thing corresponding to an award property, requires a more detailed analysis since the thing received may represent items other than awards. We believe this can be significantly improved by applying a more fine-grained approach during the generation of mappings. We also believe that retraining the named entity linker and improving filtering of erroneous coreference mentions may increase the results.

The resulting database may be used to populate Wikipedia articles with lacking or sparse infoboxes and to aid article authors. We also believe that a future application of the framework might be to fully create Wikipedias from unstructured text.

By using the framework, we wish to bridge the gap between unstructured and annotated text, in essence, creating training material for machine learning. One possible application that we wish to investigate is the creation of parallel corpora, by means of entity linking.

An archive of extracted triples is available for download in N-Triple format[3].

Acknowledgments. This research was supported by Vetenskapsrådet, the Swedish research council, under grant 621-2010-4800 and has received funding from the European Union's seventh framework program (FP7/2007-2013) under grant agreement 230902.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A nucleus for a web of open data. In: The Semantic Web. Volume 4825 of LNCS. Springer Berlin (2007) 722–735

---

[3] `http://semantica.cs.lth.se/`

2. Lange, D., Böhm, C., Naumann, F.: Extracting structured information from Wikipedia articles to populate infoboxes. In: Proceedings of the 19th CIKM, Toronto (2010) 1661–1664
3. Augenstein, I., Padó, S., Rudolph, S.: LODifier: Generating linked data from unstructured text. In: The Semantic Web: Research and Applications. Volume 7295 of LNCS. Springer Berlin (2012) 210–224
4. Cattoni, R., Corcoglioniti, F., Girardi, C., Magnini, B., Serafini, L., Zanoli, R.: The KnowledgeStore: an entity-based storage system. In: Proceedings of LREC 2012, Istanbul (2012)
5. Agichtein, E., Gravano, L.: Snowball: extracting relations from large plain-text collections. In: Proceedings of DL '00, New York, ACM (2000) 85–94
6. Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A.M., Shaked, T., Soderland, S., Weld, D.S., Yates, A.: Web-scale information extraction in knowitall. In: Proceedings of WWW '04, New York, ACM (2004) 100–110
7. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, E.R., Mitchell, T.M.: Toward an architecture for never-ending language learning. In: Proceedings of AAAI-10. (2010) 1306–1313
8. Banko, M., Etzioni, O.: Strategies for lifelong knowledge extraction from the web. In: Proceedings of K-CAP '07, New York, ACM (2007) 95–102
9. Fader, A., Soderland, S., Etzioni, O.: Identifying relations for open information extraction. In: Proc. of EMNLP '11. (2011) 1535–1545
10. Suchanek, F.M., Sozio, M., Weikum, G.: Sofie: A self-organizing framework for information extraction. In: Proceedings of WWW '2009, New York (2009) 631–640
11. Exner, P., Nugues, P.: Constructing large proposition databases. In: Proc. of LREC'12, Istanbul (2012)
12. Fillmore, C.J.: Frame semantics and the nature of language. Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech **280** (1976) 20–32
13. Palmer, M., Gildea, D., Kingsbury, P.: The Proposition Bank: an annotated corpus of semantic roles. Computational Linguistics **31** (2005) 71–105
14. Björkelund, A., Hafdell, L., Nugues, P.: Multilingual semantic role labeling. In: Proceedings of CoNLL-2009, Boulder (2009) 43–48
15. Bohnet, B.: Top accuracy and fast dependency parsing is not a contradiction. In: Proceedings of COLING-2010, Beijing (2010) 89–97
16. Raghunathan, K., Lee, H., Rangarajan, S., Chambers, N., Surdeanu, M., Jurafsky, D., Manning, C.: A multi-pass sieve for coreference resolution. In: Proc. of EMNLP-2010, Boston (2010) 492–501
17. Lee, H., Peirsman, Y., Chang, A., Chambers, N., Surdeanu, M., Jurafsky, D.: Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task. In: Proceedings of the CoNLL-2011 Shared Task, Boulder (2011) 28–34
18. Ratinov, L., Roth, D., Downey, D., Anderson, M.: Local and global algorithms for disambiguation to Wikipedia. In: Proceedings of the 49th Annual Meeting of the ACL, Portland (2011) 1375–1384

A Semantic Best-Effort Approach for Extracting Structured Discourse Graphs from Wikipedia

André Freitas[1], Danilo S. Carvalho[2], João C. P. da Silva[3], Seán O'Riain[1], and Edward Curry[1]

[1]Digital Enterprise Research Institute (DERI)
National University of Ireland, Galway
[2]Department of Systems Engineering and Computer Science (COPPE) &
[3]Computer Science Department
Federal University of Rio de Janeiro (UFRJ)

Abstract. Most information extraction approaches available today have either focused on the extraction of simple relations or in scenarios where data extracted from texts should be normalized into a database schema or ontology. Some relevant information present in natural language texts, however, can be irregular, highly contextualized, with complex semantic dependency relations, poorly structured, and intrinsically ambiguous. These characteristics should also be supported by an information extraction approach. To cope with this scenario, this work introduces a *semantic best-effort information extraction approach*, which targets an information extraction scenario where text information is extracted under a pay-as-you-go data quality perspective, trading high-accuracy, schema consistency and terminological normalization for domain-independency, context capture, wider extraction scope and maximization of the text semantics extraction and representation. A semantic information extraction framework (*Graphia*) is implemented and evaluated over the Wikipedia corpus.

Keywords: Semantic Best-effort extraction, Information Extraction, Semantic Networks, RDF, Linked Data, Semantic Web

1 Introduction

The Linked Data Web brings the vision of a semantic data graph layer on the Web which can improve the ability of users and systems to access and semantically interpret information. Currently most datasets on the Linked Data Web, such as DBpedia, are built from data already structured in different formats, which are mapped to an ontology/vocabulary and are transformed into RDF. Despite its fundamental importance as a grassroots movement to make available a first layer of data on the Web, sharing structured databases on the Web will not be sufficient to make the Semantic Web vision [1] concrete. Most of the information available on the Web today is in a unstructured text format. The integration of this information into the Linked Data Web is a fundamental step towards enabling the Semantic Web vision.

The semantics of unstructured text, however, does not easily fit into structured datasets. While the representation of structured data assumes a high level of regularity, relatively simple conceptual models and a consensual semantics between the users of a structured dataset, the representation of information extracted from texts need to take into account large terminological variation, complex context patterns, fuzzy and conflicting semantics and intrinsically ambiguous sentences. Most information extraction (IE) approaches targeting the extraction of facts from unstructured text have focused on extraction scenarios where accuracy, consistency and a high level of lexical and structural normalization are primary concerns, as in the automatic construction of ontologies and databases. These IE approaches can be complemented by alternative information extraction scenarios where accuracy, consistency and regularity are traded by domain-independency, context capture, wider extraction scope and maximization of the text semantics representation, under a *pay-as-you-go* data quality perspective [8], where data semantics and data quality are built and improved over time. We call an information extraction strategy focused on these aspects a *semantic best-effort information extraction* approach. This type of approach provides a complementary semantic layer, enriching existing datasets and bridging the gap between the Linked Data Web and the Web of Documents.

This work focuses on the construction and analysis of a *semantic best-effort information extraction approach.* The approach extracts *structured discourse graphs* (SDGs) from texts, a representation introduced in [5] which focuses on a RDF compatible graph representation which maximizes the representation of text elements and context under a pay-as-you-go data extraction scenario. Potential applications of this work are: (i) structured and unstructured data integration (ii) open information extraction for IR support, (iii) enrichment of existing Linked Datasets such as DBpedia and YAGO [6].

The contributions of this paper are: (i) deepening the discussion on the pay-as-you-go semantic best-effort information extraction, (ii) a semantic best-effort graph extraction pipeline based on the SDG representation (iii) the implementation of the pipeline in the *Graphia* extraction framework and (iv) the evaluation of the extraction pipeline using Wikipedia as a corpus.

This paper is organized as follows: section 2 provides a motivational scenario based on DBpedia and Wikipedia; section 3 provides a overview of the SDG representation model [5]; section 4 describes the architecture and the components of the semantic best-effort extractor; section 5 provides an experimental analysis of the extraction approach using Wikipedia as a corpus; section 6 analyses the related work in the area; finally, section 7 provides a conclusion and describes future work.

2 Motivational Scenario

The core motivation for a semantic best-effort (SBE) extraction is to provide a structured discourse representation which can enrich datasets with information present in unstructured texts. Currently datasets such as DBpedia are created by extracting (semi-)structured information from Wikipedia. With an appropriate graph representation, it is possible to provide an additional layer for knowledge discovery (KD), search, query and navigation (Figure 1). As a motivational scenario suppose a user wants to know possible connections between Barack Obama and Indonesia. Today this information cannot be directly found in DBpedia, and the user would need to browse and read through Wikipedia articles to find this information. A semantic best-effort structured discourse graph (SDG) can provide an additional link structure extracted from text which, starting from the DBpedia entity Barack Obama, can be used by an application to find the semantic connection with the other DBpedia entity Indonesia. This intermediate layer between text and datasets (Figure 1) has a different level of representation from traditional, ontology-based RDF datasets. In the example, the sentence and its corresponding extracted graph (Figure 1), the temporal references ('from age six to ten') are not resolved to a normalized temporal representation, and only the information present in the verb tense is used to define a temporal context, showing the semantic best-effort/pay-as-you-go nature of the approach. Additionally, the context where the original sentence is embedded in the text is mapped to the graph through a *context_link*. A semantic best-effort extraction/representation provides the core structure of the sentence and its discourse context, maximizing the representation of the text information, allowing the future extension/refinement of the extracted information. The representation of complex and composite relations is a fundamental element in information extraction. In the example scenario, a simple relation extraction would focus on the extraction of triples such as (*Barack Obama, attended, local school*) which does not provide a connection between Barack Obama and Indonesia.

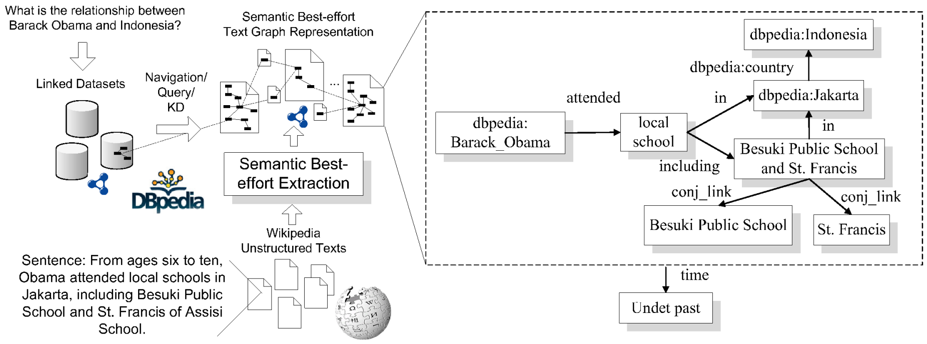

What is the relationship between Barack Obama and Indonesia?
Linked Datasets
Navigation/ Query/ KD
Semantic Best-effort Text Graph Representation
Semantic Best-effort Extraction
Wikipedia Unstructured Texts
Sentence: From ages six to ten, Obama attended local schools in Jakarta, including Besuki Public School and St. Francis of Assisi School.
DBpedia
dbpedia:Indonesia
dbpedia:country
dbpedia:Jakarta
attended
in
in
dbpedia: Barack_Obama
local school
including
Besuki Public School and St. Francis
conj_link
conj_link
Besuki Public School
St. Francis
time
Undet past

Fig. 1. Motivational scenario and example of a SBE graph representation.

3 Representing Text as Discourse Graphs

The objective of structured discourse graphs (SDGs) introduced in [5] is to provide a principled representation for text elements which supports a semantic best-effort extraction. A semantic best-effort (SBE) extraction aims at maximizing the amount of extracted information present in the text, capturing the semantic context and the semantic dependencies where a given fact is embedded. A SBE extraction also minimizes the semantic impact of potential extraction errors by maximizing the semantic isolation between structures associated with different types of extraction operations (e.g. relation extraction, temporal resolution and co-reference resolution) and by facilitating the process of navigating back to the original text source. This isolation facilitates the data consumption/interpretation process under the pay-as-you-go scenario, where the impact of possible incomplete or erroneous extractions is minimized. SDGs provide a representation complementary to Discourse Representation Structures (DRSs). In fact, DRSs can be represented as SDGs [5]. SDGs approach the representation problem from both a data generation (under a SBE scenario) and also from a data consumption perspective. SDGs are designed to be an RDF-based graph representation from the start, also providing a principled semantic interpretation of the graph data through a graph navigation algorithm, facilitating its use under the Linked Data context.

The following items describe the main elements of the structured discourse graph model introduced in [5]. Real sentence graphs extracted from the Wikipedia article *Barack Obama* by the *Graphia*[1] framework are used as examples to introduce the elements of the extraction model. The elements described below are combined into a graph structure which allows a principled algorithmic interpretation model. A more detailed discussion on the SDG representation can be found in [5]. The SDG representation consists of the following core elements:

Named, non-named entities and properties: *Named entities* include categories such as proper nouns, temporal expressions, biological species, substances, among other categories. A named entity is defined by one or more proper nouns (**NNP**) in a noun phrase (**NP**). In RDF, named entities map to instances. *Non-named entities* are more subject to vocabulary variation ('President of the United States', 'American President'), i.e. polysemy and homonymy. Additionally, non-named entities have more complex compositional patterns: commonly non-named entities are composed with less specific named or non-named entities, which can be referenced in different contexts. A non-named entity is defined by one or more nouns (**NN**), adjectives (**JJ**) in a noun phrase (**NP**). In RDF a non-named entity maps to a class which can be referred both as a class and as an instance (*punning*)[2]. *Properties* are built from verbs (**VB**) or from passive verb constructions. Named, non-named entities and properties form the basic triple (relation) pattern which is complemented by the SDG elements below.

Quantifiers & Generic Operators: Represent a special category of nodes which provide an additional qualification over named or non-named entities. Both quantifiers and generic operators are specified by an enumerated set of elements which map to *adverbs*, *numbers*, *comparative and superlative* (suffixes and modifiers). Examples of quantifiers and operators are: *Quantifier:* e.g. one,

---

[1] http://graphia.dcc.ufrj.br

[2] http://www.w3.org/2007/OWL/wiki/Punning

two, (cardinal numbers), many (much), some, all, thousands of, one of, several, only, most of; *Negation:* e.g. not *Modal:* e.g. could, may, shall, need to, have to, must, maybe, always, possibly; *Comparative:* e.g. largest, smallest, most, largest, smallest. Ex.: Figure 2(E).

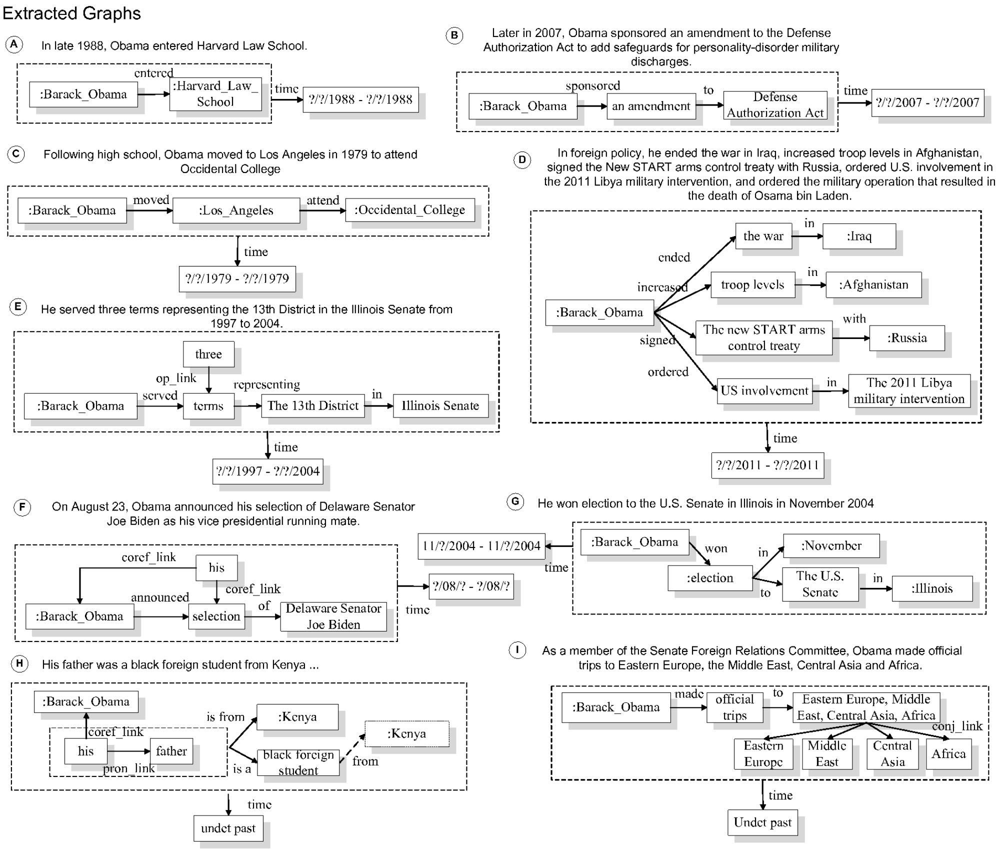

Extracted Graphs
A In late 1988, Obama entered Harvard Law School.
:Barack_Obama
entered
:Harvard_Law_School
time
?/?/1988 - ?/?/1988
B Later in 2007, Obama sponsored an amendment to the Defense Authorization Act to add safeguards for personality-disorder military discharges.
:Barack_Obama
sponsored
an amendment
to
Defense Authorization Act
time
?/?/2007 - ?/?/2007
C Following high school, Obama moved to Los Angeles in 1979 to attend Occidental College
:Barack_Obama
moved
:Los_Angeles
attend
:Occidental_College
time
?/?/1979 - ?/?/1979
D In foreign policy, he ended the war in Iraq, increased troop levels in Afghanistan, signed the New START arms control treaty with Russia, ordered U.S. involvement in the 2011 Libya military intervention, and ordered the military operation that resulted in the death of Osama bin Laden.
:Barack_Obama
ended
the war
in
:Iraq
increased
troop levels
in
:Afghanistan
signed
The new START arms control treaty
with
:Russia
ordered
US involvement
in
The 2011 Libya military intervention
time
?/?/2011 - ?/?/2011
E He served three terms representing the 13th District in the Illinois Senate from 1997 to 2004.
three
op_link
:Barack_Obama
served
terms
representing
The 13th District
in
Illinois Senate
time
?/?/1997 - ?/?/2004
F On August 23, Obama announced his selection of Delaware Senator Joe Biden as his vice presidential running mate.
coref_link
his
coref_link
announced
:Barack_Obama
selection
of
Delaware Senator Joe Biden
?/08/? - ?/08/?
time
G He won election to the U.S. Senate in Illinois in November 2004
11/?/2004 - 11/?/2004
time
:Barack_Obama
won
:election
in
:November
to
The U.S. Senate
in
:Illinois
H His father was a black foreign student from Kenya ...
:Barack_Obama
coref_link
his
father
pron_link
is from
:Kenya
is a
black foreign student
from
:Kenya
time
undet past
I As a member of the Senate Foreign Relations Committee, Obama made official trips to Eastern Europe, the Middle East, Central Asia and Africa.
:Barack_Obama
made
official trips
to
Eastern Europe, Middle East, Central Asia, Africa
conj_link
Eastern Europe
Middle East
Central Asia
Africa
time
Undet past

Fig. 2. Examples of extracted sentence graphs from the Wikipedia article *Barack Obama*. Nodes with a ':' depict entities resolved to DBpedia URIs.

Triple Trees: Not all sentences can be represented in one triple. On a normalized dataset scenario, one semantic statement which demands more than one triple is mapped to a conceptual model structure (as in the case of events for example) which is not explicitly present in the discourse. In the unstructured text graph scenario, sentences which demand more than one triple can be organized into a triple tree. A triple tree is built by a mapping from the syntactic tree of a sentence to a set of triples, where the sentence subject defines the root node of the triple tree. The interpretation of a triple tree is defined by a complete

DFS traversal of the tree, where each connected path from the root node to a non-root node defines an *interpretation path.* Ex.: Figure 2(C).

Context elements: A fact extracted from a natural language text demands a semantic interpretation which may depend on different contexts where the fact is embedded (such as a temporal context). Intra-sentence dependencies are given by dependencies involving a different clause in the same sentence. Intra-sentence context for a triple can be represented by the use of reification (Figure 2). Contexts can also be important to define the semantics of an entity present in two or more triple trees. For example the interpretation of an entity which is neither a root and a leaf node (Figure 2(D)) demands the capture of the pairwise combination of its backwards and forward properties in multiple contexts. This is lost in a typical dereferenciation process where all properties and objects associated with an entity are returned. A third level of context can be defined by mapping the dependencies between extracted triple trees, taking into account the sentences ordering and the relation to text elements in the original discourse. Ex.: Temporal nodes in Figure 2.

Co-Referential elements: Some discourse elements contain indirect references to named entities (*pronominal* & *non-pronominal* co-references). Co-references can refer to either intra or inter sentences named entities. While in some cases co-references can be handled by substituting the co-referent term by the named entity (as in personal pronouns), in other cases this direct substitution can corrupt the semantics of the representation (as in the case of reflexive and personal pronouns) or can mask errors in a semantic best-effort extraction scenario. Co-reference terms include: you, I, someone, there, this, himself, her, this, that, etc. Ex.: Figure 2(F)(H)(I).

Resolved & normalized entities: Resolved entities are entities where a node-substitution in the graph was made from a co-reference to a named entity (e.g. a *personal pronoun* to a named entity). Normalized entities are entities which were transformed to a normalized form. A temporal normalization where date & time references are mapped to a standardized format (September 1st of 2010 mapped to 01/09/2010). Ex.: Figure 2(A)-(G).

4 Structured Discourse Graphs Extraction

4.1 Mapping Natural Language to SDGs

This section describes the basic components of a semantic best-effort extraction pipeline targeting the proposed representation. The extraction pipeline was designed targeting Wikipedia as a corpus. Wikipedia has a *factual discourse*, *a topic-oriented text organization* and *named entities KB given by DBpedia.* The extraction pipeline takes as input Wikipedia texts and returns an extracted RDF graph and a sentence-based graph visualization. The extraction pipeline consists of the following components (Figure 3):

1. Syntactic analysis: The first step in the extraction process is the syntactic parsing of the natural language text into syntactic trees (C-Structures). This module uses the Probabilistic Context-Free Grammar (PCFG) implemented in the Stanford parser. The C-Structures for the sentences are passed to the next modules.

2. Named entity resolution: This component resolves named entities text references to existing DBpedia URIs. The first step consists in the use of the DBpedia Spotlight service[3] where the full article is sent and is returned with annotated URIs. The second step consists in the use of Part-of-Speech tags together with C-Structures to aggregate words into entity candidates which were not resolved by the DBpedia Spotlight service. The entity candidates' strings are sent as search terms to a local entity index which indexes all DBpedia URIs using TF/IDF over labels extracted from the URIs. Returned URIs mapping to the search string terms are used to enrich the original annotated text file with additional URI annotations. The output of this component is the original text with a set of named entity terms annotated with URIs.

3. Personal co-reference resolution and normalization: This component resolves pronominal co-references including personal, possessive and reflexive pronouns. Personal pronouns instances are substituted by the corresponding entities. Possessive and reflexive pronouns are annotated with the corresponding entities that will later define the co-reference links. The co-reference resolution process is done by the pronoun-named entity gender and number agreement (by taking into account gender information present in a name list) and by applying a heuristic strategy based on text distance between the pronoun and named entity candidates. The output of this component are C-Structures with annotated named entities, co-reference substitutions for personal pronouns and possessive and reflexive pronouns annotated with named entities.

4. Graph extraction: The graph extraction module takes as input the annotated C-Structures and generates the triple trees for each sentence by the application of a set of transformation rules based on syntactic conditions through a DFS traversal of the C-Structure. Instead of focusing on terminology-dependent patterns, these rules are based on syntactic patterns. The core set of syntactic rules are split into 6 major categories: *subject*, *predicate*, *object*, *prepositional phrase & noun complement*, *reification*, *time*. Additional details about the graph extraction algorithm can be found online [4] .

1. *Subject: Subjects* are activated by noun phrases (**NP**) when NPs are higher into the syntactic hierarchy and without any NPs as child nodes. This rule applies the following actions: (i) concatenates the nouns in case of compound subjects; (ii) Adds the subject as a node into the triple tree; (iii) adds a URI in case the subject is a named entity.
2. *Predicate: Predicates* are defined by verbal phrases (**VB***). This rule applies the following actions: (i) verifies the verb tense and activates the rule which transforms the verb tense into a temporal representation; (ii) concatenates the neighboring verbs in case there is more than one verb; (iii) verify if the

---

[3] http://dbpedia.org/spotlight

[4] http://treo.deri.ie/sdg

verb has a property pattern and concatenates the pattern nodes defining them as a labelled edge on the triple tree; (iv) adds the predicate words to the verb/property-pattern and removes these words from the object node in the triple tree; (v) verifies the presence of an explicit temporal reference in the predicate; (vi) Adds the explicit or implicit temporal references as a reification.

3. *Object:* This rule is activated when the search reaches a NP node that does not have a child NP and is after a verb phrase. The rule applies the following actions: (i) identifies the object head; (ii) concatenates the nouns in case of a compound object; (iii) creates an object node with the object in the triple tree; (iv) in case the words in the node correspond to a recognized entity, adds the associated URI.
4. *Prepositional phrase & Noun complement:* This rule is activated when the search finds a NP node that does not have a NP as a child and that has a prepositional phrase (**PP**) as a sibling node. The goal of this rule is to find ownership relations in subjects and objects. The rule applies the following actions: (i) concatenates the words in the noun phrase; (ii) creates a graph node connected by an edge with a preposition.
5. *Reification:* This rule is activated when the search finds a preposition node. It ignores the prepositional phrases which modifies NPs, which are handled by the previous rule. The rule applies the following actions: (i) concatenates the words in the PP, excluding the preposition; (ii) creates a reification for the prepositional phrase; (iii) verifies the existence of explicit temporal references and creates temporal reification nodes.
6. *Time:* This rule is not applied over the nodes of the syntactic structure and it is indirectly invoked by the other rules. This rule identifies explicit and implicit date references. Dates are detected by a set of regular expressions, which detects and normalizes explicit date references to a predefined format. Implicit date references (verb tenses) are detected by the analysis of POS tags.

5. Graph construction: This component receives the triple trees from the previous component and outputs the final graph serialization. Context URIs are created among different sentences and among each sentence and the article context URI. Additionally, local URIs are created for each resource which was not resolved to a DBpedia URI.

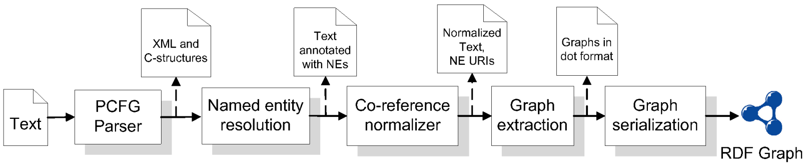


Fig. 3. High-level architecture of the SBE graph extraction pipeline.

5 Extraction & Evaluation

This section focuses on the analysis of the feasibility of a semantic best-effort extraction by evaluating the proposed extraction pipeline. The *key questions* that are targeted by the evaluation are: (i) the verification of the feasibility of extracting structured discourse graphs following the SDG representation; (ii) the quantification of the errors associated with each extraction step and (iii) the determination of which extraction error mostly impacts the semantics of the extracted graph.

The evaluation methodology is based on the work of Harrington & Clark [2], which selected a list of sample factual articles associated with named entities, and evaluated the extracted semantic networks according to a set of errors. The evaluation differs in relation to the corpus (here the corpus is the English Wikipedia) and on the final set of error categories (the error categories in this work target the generation of the core elements of the representation). Articles were selected randomly satisfying the following criteria: 2 articles about people, 2 articles about organizations and 1 article about a place. Each article has a number of characters greater than 40K. The article size served as an indicator of a more diverse discourse sample base and of the quality of the discourse. The selected articles were: Apple Inc., Google, Napoleon, Paris, John Paul II.

Error Categories	Apple	Google	Napoleon	John Paul	Paris	Avg.
Reification construction	0.20	0.13	0.10	0.06	0.17	**0.132**
Pronominal co-reference	0.13	0.10	0.03	0.00	0.03	**0.058**
Conjunction	0.00	0.06	0.03	0.03	0.06	**0.036**
Named entity	0.06	0.06	0.00	0.06	0.10	**0.056**
Subject construction	0.10	0.06	0.10	0.10	0.20	**0.112**
Object construction	0.16	0.23	0.26	0.23	0.34	**0.244**
Triple tree construction	0.33	0.26	0.20	0.30	0.31	**0.280**
Predicate construction	0.23	0.23	0.06	0.13	0.03	**0.136**
Explicit temporal reference	0.16	0.03	0.10	0.20	0.06	**0.110**
Accuracy						
Correct graphs	0.39	0.46	0.40	0.56	0.43	**0.448**
Complete graphs	0.16	0.23	0.20	0.16	0.06	**0.162**
Interpretable graphs	0.99	0.96	0.93	0.96	0.94	**0.956**

Table 1. Accuracy and frequency of extraction error categories.

The quality of the extraction was manually evaluated for each graph generated from a sentence. Sentences which were not well-formed or which were classified as outside the scope of the extraction pipeline (sentences with complex *subordination* structures [4]) were removed from the evaluation set. The final dataset consists of 1033 relations (triples) from 150 sentences which were manually classified [4] . Comparatively, for a related work using human-based evaluation, Harrington & Clark [2] evaluates approximately 160 relations and 5 topics. The extraction pipeline was implemented in Python following the architecture outlined in the previous section. A web evaluation platform was built to allow an efficient manual evaluation process. In the evaluation platform the original natural language sentence and a visualization of the extracted graph are displayed to a human evaluator, who classifies the sentences in relation to: (i) 10 sentence features (to guarantee an heterogeneous and complete sample set, which evaluates all aspects of the extraction pipeline), (ii) 9 error categories (indicate the quality impact of each pipeline component) and (iii) the accuracy of the extraction (to evaluate how each error category impacts the final extraction). The list of sentence features can be found online [4] . Table 1 shows the categorized frequency of errors for each article together with the associated extraction accuracy. To evaluate the accuracy in a semantic best-effort scenario three measures were defined: the *correctness*, the *completeness* and the *interpretability* of the graph extractions. These three measures represent different levels of accuracy: A *correct graph* is an extracted graph which is fully consistent with the semantic model; a *complete graph* is a correct graph which maps all the information of a sentence, and an *interpretable graph* is a graph fragment which has the correct semantics of its *basic triple paths* (core s, p, o pattern from the main clause), despite the possible presence of extraction errors in other extracted structures (such as co-reference links and reified statements). The correctness of the basic triple paths is the most important element in the extraction, highly impacting the interpretability and usability of the extracted SDG.

The high percentage of interpretable graphs, shows that there is a basic triple path which is correct in **95.6%** of the extracted graphs. The extractor is able to extract an informational and correct fragment in practically all the sentences. **55.2%** of graphs contained some extraction error. Only **16.2%** of the extracted graphs mapped all the information contained in the sentence, which shows the major direction for improvement (completeness), but which is aligned with a pay-as-you-go scenario. The major justification for the lack of completeness is the fact that the SBE extractor, in many occasions, ignores sentence structures which are not central (e.g. appositive) and do a partial extraction. The most impacting error categories were triple trees, object and reification construction, categories which are strongly interrelated. The error frequencies indicate that the existing extractor still needs to be improved in relation to object construction criteria, in particular in relation to the extraction of non-named entities. The low frequency of errors related to named and temporal entities shows the robustness on the determination of these semantic pivots. The relatively high reification construction error frequency shows that the breadth of the rules for extracting prepositional phrases is still limited. The proposed representation supported the best-effort extraction by isolating errors from different parts of the extraction pipeline, keeping a high number of graph fragments interpretable even when a component of the pipeline fails. The final extracted graphs were easily represented as RDF. However, the centrality on the modelling of context brings mechanisms such as reifications, named graphs (quads) and quints to the center of the discussion for text representation.

6 Related Work

Existing related work can be classified in three main categories: *semantic networks extraction from texts* [2, 3], *open relation extraction* [7, 4] and *ontology extraction from Wikipedia* [6].

Harrington & Clark [2] describe AskNet, an information extraction system which builds large scale semantic networks from unstructured texts. The extraction pipeline of AskNet starts with the parsing of text sentences using the C&C parser [2], a parser based on the linguistic formalism of Combinatory Categorial Grammar (CCG). A Named Entity Recognition (NER) stage is performed using the C&C NER tagger. After the sentences are parsed, AskNet uses the Boxer semantic analysis tool [2], which produces a first-order logic representation based on the semantic model of the Discourse Representation Theory (DRT). A low coverage pronoun resolution approach is used for pronominal co-references. Wojtinnek et al. [3] provides an introductory discussion on the RDF translation of the AskNet output. No principled discussion on the discourse and graph representation is provided in [2, 3]. Despite having similar objectives, the approach used in the SBE extraction pipeline is significantly different (parser, NER and pronominal co-reference resolution strategy). On the representation side, this work targets a graph representation and algorithmic interpretation which focuses on RDF and is not directly mediated by DRT.

TextRunner [7] is an open information extraction (domain independent) framework. TextRunner uses a single-pass extractor consisting of a POS-tagger and a lightweight noun phrase chunker to determine the core entities in a sentence, normalizing relations by removing less semantically significative terms (e.g. modifiers), defining a probabilistic redundancy model based on the frequency of normalized facts as a correctness estimator. Comparatively, TextRunner focuses on the extraction of simple relations and does not cover the representation of more complex discourse structures. Co-reference resolution is not covered in its extraction process. Nguyen et al. [4] propose an approach for relation extraction over Wikipedia by mining frequent subsequences from the syntactic and semantic path between entity pairs in the corpus. The approach uses dependency structures and semantic role labelling and does not focus on the extraction and representation of complex relations.

YAGO2 is an extension of YAGO which targets the extraction and representation of temporal and spatial statements. To assign a spatio-temporal dimension to the facts, a new representation (SPOTL(X)) is proposed. The focus on Wikipedia, the centrality of the representation of reifications, and the definition of a temporal model are common aspects between YAGO2 and this work.

7 Conclusion & Future Work

This work focuses on the analysis of a semantic best-effort extraction approach using structured discourse graphs (SDGs), a RDF-based discourse representation format. A semantic best-effort extraction pipeline is proposed and is implemented on the *Graphia* framework. The quality of the proposed extraction approach is evaluated over Wikipedia. The final extraction achieved **44.8%** of correctness, **16.2%** completeness and an interpretability of **95.6%**. The representation played a key role in isolating errors from different components of the extraction pipeline, impacting on the interpretability performance. The final approach showed a high coverage of the elements of the SDG representation model, with the evaluation pointing into a main direction for improvement: increasing the extraction completeness. The evaluation of error categories shows that this can be achieved by improving non-named entity recognition criteria and the treatment of prepositional phrases.

Acknowledgments. The work presented in this paper has been funded by Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Lion-2).

References

1. T. Berners-Lee, J. Hendler and O. Lassila, The Semantic Web, Scientific American, May, p. 29-37, 2001.
2. B. Harrington and S. Clark. ASKNet: Creating and Evaluating Large Scale Integrated Semantic Networks. Intl. Journal of Semantic Computing, 2(3), 2009.
3. P-R. Wojtinnek, B. Harrington, S. Rudolph and S. Pulman. Conceptual Knowledge Acquisition Using Automatically Generated Large-Scale Semantic Networks. In Proc. of the 18th Intl. Conference on Conceptual Structures, 2010.
4. D. P.T Nguyen, Y. Matsuo and M. Ishizuka, Exploiting Syntactic and Semantic Information for Relation Extraction from Wikipedia. IJCAI Workshop on Text-Mining & Link-Analysis, 2007.
5. A. Freitas, D.S. Carvallho, J.C. P. da Silva, S. O'Riain, E. Curry. A Structured Discourse Graph Representation for a Semantic Best-Effort Text Extraction (to appear), available at: http://treo.deri.ie/sdg, 2012.
6. J. Hoffart, F. Suchanek, K. Berberich and G. Weikum. YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia. Special Issue of the Artificial Intelligence Journal, 2012.
7. M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead and O. Etzion. Open Information Extraction from the Web. In Proc. of the Intl. Joint Conference in Artificial Intelligence, 2007.
8. M. J. Franklin, A. Y. Halevy, D. Maier: From databases to dataspaces: a new abstraction for information management. SIGMOD Record 34(4): 27-33, 2005.

Query Segmentation and Resource Disambiguation Leveraging Background Knowledge

Saeedeh Shekarpour[1], Axel-Cyrille Ngonga Ngomo[1], and Sören Auer[1]

Department of Computer Science, University of Leipzig Johannisgasse 26,
04103 Leipzig {`lastname`}`@informatik.uni-leipzig.de`

Abstract. Accessing the wealth of structured data available on the Data Web is still a key challenge for lay users. Keyword search is the most convenient way for users to access information (e.g., from data repositories). In this paper we introduce a novel approach for determining the correct resources for user-supplied keyword queries based on a hidden Markov model. In our approach the user-supplied query is modeled as the observed data and the background knowledge is used for parameter estimation. Instead of learning parameter estimation from training data, we leverage the semantic relationships between data items for computing the parameter estimations. In order to maximize accuracy and usability, query segmentation and resource disambiguation are mutually tightly interwoven. First, an initial set of potential segmentations is obtained leveraging the underlying knowledge base; then the final correct set of segments is determined after the most likely resource mapping was computed using a scoring function. While linguistic methods like named entity, multi-word unit recognition and POS-tagging fail in the case of an incomplete sentences (e.g. for keyword-based queries), we will show that our statistical approach is robust with regard to query expression variance. Our experimental results when employing the hidden Markov model for resource identification in keyword queries reveal very promising results.

1 Introduction

The Data Web currently amounts to more than 31 billion triples[1] and contains a wealth of information on a large number of different domains. Yet, accessing this wealth of structured data *remains* a key challenge for lay users. The same problem emerged in the last decade when users faced the huge amount of information available of the Web. Keyword search has been employed by popular Web search engines to provide access to this information in a user-friendly, low-barrier manner. However, keyword search in structured data raises two main difficulties: First, the *right segments of data items* that occur in the keyword queries have to be identified. For example, the query *'Who produced films starring Natalie*

---

[1] See http://www4.wiwiss.fu-berlin.de/lodcloud/state/ (May 23th, 2012)

Portman' can be segmented to (*'produce'*, *'film'*, *'star'*, *'Natalie Portman'*) or (*'produce'*, *'film star'*, *'Natalie'*, *'Portman'*). Note that the first segmentation is more likely to lead to a query that contain the results intended for by the user. Second, these segments have to be disambiguated and mapped to the right resources. Note that the resource ambiguity problem is of increasing importance as the size of knowledge bases on the Linked Data Web grows steadily. Considering the previous example[2], the segment *'film'* is ambiguous because it may refer to the class `dbo:Film` (the class of all movies in DBpedia) or to the properties `dbo:film` or `dbp:film` (which relates festivals and the films shown during these festivals). In this paper, we present an automatic query segmentation and resource disambiguation approach leveraging background knowledge. Note that we do not rely on training data for the parameter estimation. Instead, we leverage the semantic relationships between data items for this purpose. While linguistic methods like named entity, multi-word unit recognition and POS-tagging fail in the case of an incomplete sentences (e.g. for keyword-based queries), we will show that our statistical approach is robust with regard to query expression variance. This article is organized as follows: We review related work in Section 2. In Section 3 we present formal definitions laying the foundation for our work. In the section 4 our approach is discussed in detail. For a comparison with natural language processing (NLP) approaches section 5 introduces an NLP approach for segmenting query. Section 6 presents experimental results. In the last section, we close with a discussion and an outlook on potential future work.

2 Related Work

Keyword queries are usually short and lead to significant keyword ambiguity [13]. Segmentation has been studied extensively in the natural language processing (NLP) literature e.g., [8]). NLP techniques for chunking such as part-of-speech tagging or name entity recognition cannot achieve high performance when applied to query segmentation. The work [7] addresses the segmentation problem as well as spelling correction. It employs a dynamic programming algorithm based on a scoring function for segmentation and cleaning. The work presented in [11] proposes an unsupervised approach to query segmentation in Web search. The work [15] is a supervised method based on Conditional Random Fields (CRF) whose parameters are learned from query logs. For detecting named entities, [3] uses query log data and Latent Dirichlet Allocation. In addition to query logs, various external resources such as Webpages, search result snippets and Wikipedia titles and using a history of the user activities have been used [9, 12, 1, 10]. Still, the most common approach is using context for disambiguation [6, 2, 5]. In this work, resource disambiguation is based on the structure of the knowledge at hand as well as semantic relations between the candidate resources mapped to the valid segments of the input query.

---

[2] The underlying knowledge base and schema used throughout the paper for examples and evaluation is DBpedia 3.7 dataset and ontology.

```
Data: $q$: n-tuple of keywords, knowledge base
Result: SegmentSet: Set of segments
SegmentSet=new list of segments;
start=1;
while $start <= n$ do
    $i = start$;
    while $S_{(start,i)}$ is valid do
        $SegmentSet.add(S_{(start,i)})$;
        i++;
    end
    start++;
end
```

Algorithm 1: Naive algorithm for determining all valid segments taking the order of keywords into account.

3 Formal Specification

RDF data is modeled as a directed, labeled graph $G = (V, E)$ where V is a set of nodes i.e. the union of entities and property values, and E is a set of directed edges i.e. the union of object properties and data value properties. The user-supplied query can be either a complete or incomplete sentence. However, after removing the stop words, typically set of keywords remains. The order in which keywords appear in the original query is partially significant. Our approach can map adjacent keywords to a joint resource. However, once a mapping from keywords to resources is established the order of the resources does not affect the SPARQL query construction anymore. This is a reasonable assumption, since users will write strongly related keywords together, while the order of only loosely related keywords or keyword segments may vary. The input query is formally defined as an n-tuple of keyword, i.e. $Q = (k_1, k_2, ..., k_n)$. We aim to transform the input keywords into a suitable set of entity identifiers, i.e. resources $R = \{r_1, r_2 ... r_m\}$. In order to accomplish this task the input keywords have to be grouped together as segments and for each segment a suitable resource should be determined.

Definition 1 (Segment and Segmentation). *For a given query Q, a segment $S_{(i,j)}$ is a sequence of keywords from start position i to end position j which is denoted as $S_{(i,j)} = (k_i, k_{i+1}, ..., k_j)$. A query segmentation is an m-tuple of segments $SG_q = (S_{(0,i)}, S_{(i+1,j)}, ..., S_{(m,n)})$ where the segments do not overlap with each other and arranged in a continuous order, i.e. for two continuous segments $S_x, S_{x+1} : Start(S_{x+1}) = End(S_x) + 1$. The concatenation of segments belonging to a segmentation forms the corresponding input query Q.*

Definition 2 (Resource Disambiguation). *Lets the segmentation $SG' = (S^1_{(0,i)}, S^2_{(i+1,j)}, ..., S^x_{(m,n)})$ be a suitable segmentation for the given query Q. Each segment is mapped to multiple candidate resources from the underlying knowledge base, i.e. $S^i \rightarrow R^i = \{r_1, r_2 ... r_h\}$. The aim of disambiguation is to choose an x-tuple of resources from the Cartesian product of sets of candidate resources $(r_1, r_2, ..., r_x) \in \{R^1 \times R^2 \times ... R^x\}$ for which each r_i has two important properties. First, it is among the highest ranked candidates for the corresponding*

Segments	Samples of Candidate Resources
video	`1. dbp:video`
video game	`1. dbo:VideoGame`

game	`1. dbo:Game 2. dbo:games 3. dbp:game`
	`4. dbr:Game 5. dbr:Game_On`

publish	`1. dbo:publisher 2. dbp:publish 3. dbr:Publishing`
mean	`1. dbo:meaning 2. dbp:meaning 3. dbr:Mean 4. dbo:dean`
mean hamster	`1. dbr:Mean_Hamster_Software`
mean hamster software	`1. dbr:Mean_Hamster_Software`
hamster	`1. dbr:Hamster`
software	`1. dbo:Software 2. dbp:software`

Table 1. Generated segments and samples of candidate resources for a given query.

segment with respect to the similarity as well as popularity and second it shares a semantic relationship with other resources in the x-tuple.

When considering the order of keywords, the number of segmentations for a query Q consisting of n keywords is $2^{(n-1)}$. However, not all these segmentations contain valid segments. A valid segment is a segment for which at least one matching resource can be found in the underlying knowledge base. Thus, the number of segmentations is reduced by excluding those containing invalid segments. Algorithm 1 is an extension of the greedy approach presented in [15]. This naive approach finds all valid segments when considering the order of keywords. It starts with the first keyword in the given query as first segment, then it includes the next keyword into the current segment as a new segment and checks whether adding the new keyword would make the new segment no longer valid. We repeat this process until we reach the end of the query. As a running example, lets assume the input query is *'Give me all video games published by Mean Hamster Software'*. Table 1 shows the set of valid segments based on naive algorithm along with some samples of the candidate resources.

Resource Disambiguation using a ranked list of Cartesian product tuples: A naive approach for finding the correct $x-tuple$ of resources is using a ranked list of tuples from the Cartesian product of sets of candidate resources $\{R^1 \times R^2 \times ...R^n\}$. The n-tuples from the Cartesian product are simply sorted based on the aggregated relevance score (e.g. similarity and popularity) of all contained resources.

4 Query Segmentation and Resource Disambiguation using Hidden Markov Models

In this section we describe how hidden Markov models are used for query segmentation and resource disambiguation. First we introduce the concept of hidden Markov models and then we detail how we define the parameters of a hidden Markov model for solving the query segmentation and entity disambiguation problem.

4.1 Hidden Markov Models

The Markov model is a stochastic model containing a set of states. The process of moving from one state to another state generates a sequence of states. The probability of entering each state only depends on the previous state. This memoryless property of the model is called *Markov property.* Many real-world processes can be modeled by Markov models. A hidden Markov model is an extension of the Markov model, which allows the observation symbols to be emitted from each state with a finite probability. The main difference is that by looking at the observation sequence we cannot say exactly what state sequence has produced these observations; thus, the state sequence is *hidden.* However, the probability of producing the sequence by the model can be calculated as well as which state sequence was most likely to have produced the observations.

A hidden Markov model (HMM) is a quintuple $\lambda = (X, Y, A, B, \pi)$ where:

- X is a finite set of states, Y denotes the set of observed symbols;
- $A : X \times X \to \mathbb{R}$ is the transition matrix that each entry $a_{ij} = Pr(S_j|S_i)$ shows the transition probability from state i to state j;
- $B : X \times Y \to \mathbb{R}$ represents the emission matrix, in which each entry $b_{ih} = Pr(h|S_i)$ is associated with the probability of emitting the symbol h from state i;
- π denoting the initial probability of states $\pi_i = Pr(S_i)$.

4.2 State Space and Observation Space

State Space. A state represents a knowledge base entity. Each entity has an associated *rdfs:label* which we use to label the states. The actual number of states X is potentially high because it contains theoretically all RDF resources, i.e. $X = V \cup E$. However, in practice we limit the state space by excluding irrelevant states. A relevant state is defined as a state for which a valid segment can be observed. In other words, a valid segment is observed in an state if the probability of emitting that segment is higher than a certain threshold θ. The probability of emitting a segment from a state is computed based on a similarity scoring which we describe in the section 4.3. Therefore, the state space of the model is pruned and contains just a subset of resources of the knowledge base, i.e. $X \subset V \cup E$. In addition to these candidate states, we add an **unknown entity state** to the set of states. The *unknown entity* (UE) state comprises all entities, which are not available (anymore) in the pruned state space. The *observation space* is the set of all valid segments found in the input user query (using e.g. the Algorithm 1). It is formally is defined as $O = \{o|\text{o is a valid segment}\}$.

4.3 Emission Probability

Both the labels of states and the segments contain sets of words. For computing the emission probability of the state i and the emitted segment h, we compare the similarity of the label of state i with the segment h in two levels, namely string-similarity level and set-similarity level: (1) The *set-similarity level* measures the difference between the label and the segment in terms of the number of words using the *Jaccard similarity.* (2) The *string-similarity level* measures the string similarity of each word in the segment with the most similar word in the label using the *Levenshtein distance.* Our similarity scoring method is now a combination of these two metrics. Consider the segment $h = (k_i, k_{i+1}, ..., k_j)$ and the words from the label l divided into a set of keywords M and stopwords N, i.e. $l = M \cup N$. The total similarity score between keywords of a segment and a label is then computed as follows:

$$b_{ih} = Pr(h|S_i) = \frac{\sum_{k=i}^{j} argmax_{\forall m_i \in M}(\sigma(m_i, k_t))}{|M \cup h| + 0.1 * |N|}$$

This formula is essentially an extension of the *Jaccard similarity coefficient.* The difference is that in the numerator, instead of using the cardinality of intersections the sum of the string-similarity score of the intersections is computed. As in the Jaccard similarity, the denominator comprises the cardinality of the union of two sets (keywords and stopwords). The difference is that the number of stopwords have been down-weighted by the factor 0.1 to reduce their influence (since they do not convey much meaningful information).

4.4 Hub and Authority of States

Hyperlink-Induced Topic Search (HITS) is a link analysis algorithm for ranking Web pages [4]. Authority and hub values are defined in terms of one another and computed in a series of iterations. In each iteration, hub and authority values are normalized. This normalization process causes these values to converge eventually. Since RDF data is graph-structured data and entities are linked together, we employed a weighted version of the HITS algorithm in order to assign different popularity values to the states in the state space. For each state we assign a hub value and an authority value. A good hub state is one that points to many good authority states and a good authority state is one that is pointed to from many good hub states. Before discussing the HITS computations, we define the edges between the states in the HMM. For each two states i and j in the state space, we add an edge if there is a path in the knowledge base between the two corresponding resources of maximum length k. Note, that we also take property resources into account when computing the path length.The path length between resources in the knowledge base is assigned as weight to the edge between corresponding states. We use a weighted version of the HITS algorithm to take the distance between states into account. The authority of a state is computed as:

For all $S_i \in S$ which point to S_j : $auth_{S_j} = \sum_{\forall i} w_{i,j} * hub_{S_i}$ And the hub value of a state is computed as:

For all $S_i \in S$ which are pointed to by S_j : $hub_{S_j} = \sum_{\forall i} w_{i,j} * auth_{S_i}$ The weight $w_{i,j}$ is defined as $w_{i,j} = k - pathLength(i,j)$, where $pathLength(i,j)$ is the length of the path between i and j. These definitions of hub and authority for states are the foundation for computing the transition probability in the underlying hidden Markov model.

4.5 Transition Probability

As mentioned in the previous section, each edge between two states shows the shortest path between them with the length less or equal to k-hop. The edges are weighted by the length of the path. Transition probability shows the probability of going from state i to state j. For computing the transition probability, we take into account the connectivity of the whole of space state as well as the weight of the edge between two states. The transition probability values decrease with the distance of the states, e.g. transitions between entities in the same triple have higher probability than transitions between entities in triples connected through extra intermediate entities. In addition to the edges recognized as the shortest path between entities, there is an edge between each state and the *Unknown Entities* state. The transition probability of state j following state i denoted as $a_{ij} = Pr(S_j|S_i)$. For each state i the condition $\sum\limits_{\forall S_j} Pr(S_j|S_i) = 1$ should be held. The transition probability from the state i to *Unknown Entity* (UE) state is defined as:

$a_{iUE} = Pr(UE|S_i) = 1 - hub_{S_i}$ And means a good hub has less probability to go to *UE* state. Thereafter, the transition probability from the state i to state j is computed as:

$a_{ij} = Pr(S_j|S_i) = \frac{auth_{S_j}}{\sum\limits_{\forall a_{ik}>0} auth_{S_k}} * hub_{S_i}$. Here, the edges with the low distance value and higher authority values are more probable to be met.

4.6 Initial Probability

The initial probability π_{S_i} is the probability that the model assigns to the initial state i in the beginning. The initial probabilities fulfill the condition $\sum\limits_{\forall S_i} \pi_{S_i} = 1$. We denote states for which the first keyword is observable by $InitialStates$. The initial states are defined as follows:

$$\pi_{S_i} = \frac{auth_{S_i} + hub_{S_i}}{\sum\limits_{\forall S_j \in InitialStates} (auth_{S_j} + hub_{S_j})}$$

In fact, π_{S_i} of an initial state depends on both hub and authority values.

4.7 Viterbi Algorithm for the K-best Set of Hidden States

The optimal path through the Markov model for a given sequence (i.e. input query keywords) reveals disambiguated resources forming a correct segmentation. The *Viterbi algorithm* or *Viterbi path* is a dynamic programming approach for finding the optimal path through the markov model for a given sequence. It discovers the most likely sequence of underlying hidden states that might have generated a given sequence of observations. This discovered path has the maximum joint emission and transition probability of involved states. The sub paths of this most likely path also have the maximum probability for the respective sub sequence of observations. The naive version of this algorithm just keeps track of the most likely path. We extended this algorithm using a tree data structure to store all possible paths generating the observed query keywords. Therefore, in our implementation we provide a ranked list of all paths generating the observation sequence with the corresponding probability. After running the Viterbi algorithm for our running example, the disambiguated resources are: {*dbo:VideoGame, dbo:publisher, dbr:Mean-Hamster-Software*} and consequently the reduced set of valid segments is: {*VideoGam, publisher, Mean-Hamster-Software*} .

5 Query Segmentation using Natural Language Processing

Natural language processing (NLP) techniques are commonly used for text segmentation. Here, we use a combination of named entity and multi-word unit recognition services as well as POS-tagging for segmenting the input-query. In the following, we discuss this approach in more detail.

Detection of Segments: Formally, the detection of segments aims to transform the set of keywords $K = \{k_1, .., k_n\}$ into a set of segments $\mathcal{T} = \{t_1, ..., t_m\}$ where each k_i is a substring of exactly one $t_j \in \mathcal{T}$. Several approaches have already been developed for this purpose, each with its own drawbacks: Semantic lookup services (e.g., *OpenCalais*[3] and *Yahoo! SeoBook*[4] as used in the current implementation) allow to extract *named entities* (NEs) and *multi-word units* (MWUs) from query strings. While these approaches work well for long queries such as *"Films directed by Garry Marshall starring Julia Roberts"*, they fail to discover noun phrases such as "highest place" in the query *"Highest place of Karakoram"*. We remedy this drawback by combining lookup services and a simple noun phrase detector based on POS tags. This detector first applies a POS tagger to the query. Then, it returns all sequences of keywords whose POS tags abide by the following right-linear grammar:

1. $S \rightarrow adj\ A$
2. $S \rightarrow nn\ B$
3. $A \rightarrow B$
4. $B \rightarrow nn$
5. $B \rightarrow nn\ B$

where S is the start symbol, A and B are non-terminal symbols and nn (noun) as well as adj (adj) are terminal symbols. The compilation of segments is carried as follows: We send the input K to the NE and MWU detection services as well as to the noun phrase detector. Let $\mathcal{N}$ be the set of NEs, $\mathcal{M}$ the set of MWUs and $\mathcal{P}$ the set of noun phrases returned by the system. These three sets are merged to a set of labels $\mathcal{L} = (\mathcal{N} \oplus \mathcal{M}) \oplus \mathcal{P}$, where $\oplus$ is defined as follows:

$$A \oplus B = A \cup B \backslash \{b \in B | \exists a \in A\ overlap(a, b)\}, \tag{1}$$

where $overlap(a, b)$ is true if the strings a and b overlap. The operation $\oplus$ adds the longest elements of B to A that do not overlap with A. Note that this operation is not symmetrical and prefers elements of the set A over those of the set B.

---

[3] http://viewer.opencalais.com/

[4] http://tools.seobook.com/yahoo-keywords/

6 Evaluation

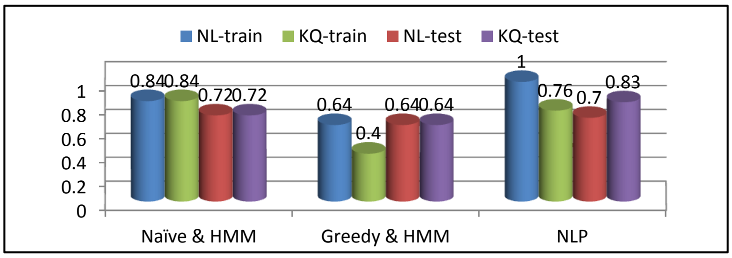
The goal of our experiments was to measure the accuracy of resource disambiguation approaches for generating adequate SPARQL queries. Thus, the main question behind our evaluation was as follows: Given a keyword-based query(KQ) or a natural-language query (NL) and the equivalent SPARQL query, how well do the resources computed by our approaches resemble the gold standard. It is important to point out that a single erroneous segment or resource can lead to the generation of a wrong SPARQL query. Thus, our criterion for measuring the correctness of segmentations and disambiguations was that *all of the recognized segments* as well as *all of the detected resources* had to match the gold standard.

Experimental Setup So far, no benchmark for query segmentation and resource disambiguation has been proposed in literature. Thus, we created such a benchmark from the DBpedia fragment of the question answering benchmark *QALD-2*[5]. The QALD-2 benchmark data consists of 100 training and 100 test questions in natural-language that are transformed into SPARQL queries. In addition, it contains a manually created keyword-based representation of each of the natural-language questions. The benchmark assumed the generic query generation steps for question answering: First, the correct segments have to be computed and mapped to the correct resources. Then a correct SPARQL query has to be inferred by joining the different resources with supplementary resources or literals. As we are solely concerned with the first step in this paper, we selected 50 queries from the QALD-2 benchmark (25 from the test and 25 from the training data sets) that were such that each of the known segments in the benchmark could be mapped to exactly one resource in the SPARQL query and vice-versa. Therewith, we could derive the correct segment to resource mapping directly from the benchmark[6]. Queries that we discarded include *"Give me all soccer clubs in Spain"*, which corresponds to a SPARQL query containing the resources {`dbo:ground, dbo:SoccerClub, dbr:Spain` }. The reason for discarding this particular query was that the resource `dbo:ground` did not have any match in the list of keywords. Note that we also discarded queries requiring schema information beyond DBpedia schema. Furthermore, 6 queries out of the 25 queries from the training data set and 10 queries out of 25 queries from the test data set required a query expansion to map the keywords to resources. For instance, the keyword *"wife"* should be matched with *"spouse"* or *"daughter"* to *"child"*. Given that the approaches at hand generate and score several possible segmentations (resp. resource disambiguation), we opted for measuring the *mean reciprocal rank MRR* [14] for both the query segmentation and the resource disambiguation tasks. For each query $q_i \in Q$ in the benchmark, we compare the rank r_i assigned by different algorithms to the correct segmentation and to the resource disambiguation: $MRR(\mathcal{A}) = \frac{1}{|Q|} \sum\limits_{q_i} \frac{1}{r_i}$. Note that if the correct segmentation

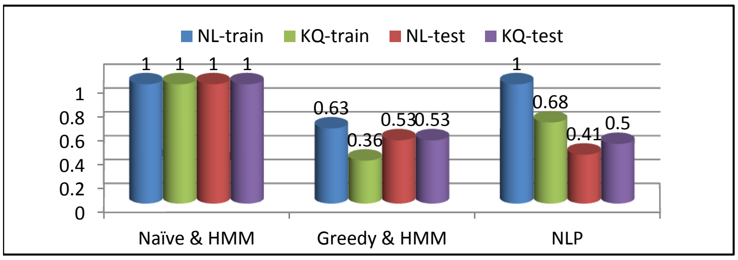
---

[5] http://www.sc.cit-ec.uni-bielefeld.de/qald-2

[6] The queries and result of the evaluation and source code is available for download at http://aksw.org/Projects/lodquery



(a) Queries that require query expansion are included.

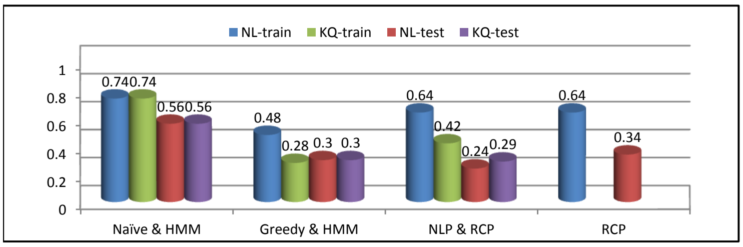


(b) Queries that require query expansion are not included.

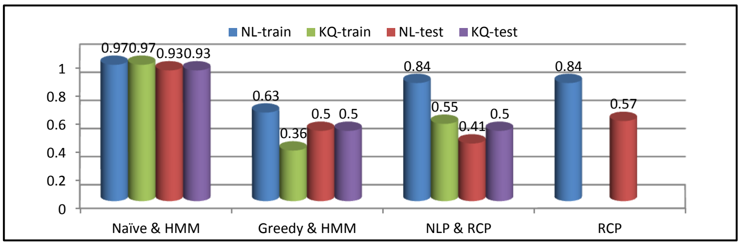
Fig. 1. Mean reciprocal rank of query segmentation (first stage).

(resp. resource disambiguation) was not found, the reciprocal rank is assigned the value 0. The parameter analysis revealed that the optimal value of θ for punning the state space is the range $[0.6, 0.7]$ which we set it to 0.7.

Results We evaluated our hidden Markov model for resource disambiguation by combining it with the naive (Naive & HMM) and the greedy segmentation (Greedy & HMM) approaches for segmentation. We use the natural language processing (NLP) approach as a baseline in the segmentation stage. For the resource disambiguation stage, we combine ranked Cartesian product (RCP) with the natural language processing (NLP & RCP) and manually injected the correct segmentation (RCP) as the baseline. Note that we refrained from using any query expansion method. The segmentation results are shown in Figure 1. The MRR are computed once with the queries that required expansion and once without. Figure 1(a), including queries requiring expansion, are slightly in favor of NLP, which achieves on overage a 4.25% higher MRR than Naive+HMM and a 24.25% higher MRR than Greedy+HMM. In particular, NLP achieves optimal scores when presented with the natural-language representation of the queries from the "train" data set. Naive+HMM clearly outperforms Greedy+HMM in all settings. The main reason for NLP outperforming Naive+HMM with respect to the segmentation lies in the fact that Naive+HMM and Greedy+HMM are dependent on matching segments from the query to resources in the knowledge base (i.e. segmentation and resource disambiguation are interwoven). Thus, when no resource is found for a segment (esp. for queries requiring expansion) the HMM prefers an erroneous segmentation, while NLP works independent from the disambiguation



(a) Queries that require query expansion are included.



(b) Queries that require query expansion are not included.

Fig. 2. Mean reciprocal rank of resource disambiguation (second stage).

phase. However, as it can be observed NLP depends on the query expression. Figure 1(b) more clearly highlights the accuracy of different approaches. Here, the MRR without queries requiring expansion is shown. Naive+HMM perfectly segments both natural language and keyword-based queries. The superiority of intertwining segmentation and disambiguation in Naive+HMM is clearly shown by our disambiguation results in the second stage in Figure 2. In this stage, Naive+HMM outperforms Greedy+HMM, NLP+RCP and RCP in all four experimental settings. Figure 2(a) shows on average 24% higher MRR, although queries requiring expansion are included. In the absence of the queries that required an expansion (Figure 2(b)), Naive+HMM on average by 38% superior to all other approaches and 25% superior to RCP. Note that RCP relies on correct segmentation which in reality is not always a valid assumption. Generally, Naive+HMM being superior to Greedy+HMM can be expected, since the naive approach for segmentation generates more segments from which the HMM can choose. Naive+HMM outperforming RCP (resp. NLP+RCP) is mostly related to RCP (resp. NLP+RCP) often failing to assign the highest rank to the correct disambiguation. One important feature of our approach is, as the evaluation confirms, the robustness with regard to the query expression variance. As shown in Figure 2, Naive+HMM achieves the same MRR on natural-language and the keyword-based representation of queries on both – the train and the test – datasets. Overall, Naive+HMM significantly outperforms our baseline Greedy+HNM as well as state-of-the-art techniques based on NLP.

7 Discussion and Future Work

We explored different methods for bootstrapping the parameters (i.e. different distributions tested e.g., normal, Zipf) of the HMM. The results achieved with these methods only led to a very low accuracy. The success of our model relies on transition probabilities which are based on the connectivity of both the source and target node (hub score of source and sink authority) as well as taking into account the connectivity (authority) of all sink states. Employing the HITS algorithm leads to distributing a normalized connectivity degree across the state space. More importantly, note that considering a transition probability to the unknown entity state is crucial, since it arranges states with the same emitted segments in a descending order based on their hub scores. Most previous work has been based on finding a path between two candidate entities. For future, we aim to realize a search engine for the Data Web, which is as easy to use as search engines for the Document Web, but allows to create complex queries and returns comprehensive structured query results[7]. A first area of improvements is related to using dictionary knowledge such as hypernyms, hyponyms or co-hyponyms.

References

1. D. J. Brenes, D. Gayo-Avello, and R. Garcia. On the fly query entity decomposition using snippets. *CoRR*, abs/1005.5516, 2010.
2. L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. Placing Search in Context: the Concept Revisited. In *WWW*, 2001.
3. J. Guo, G. Xu, X. Cheng, and H. Li. Named entity recognition in query. ACM, 2009.
4. J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5), 1999.
5. R. Kraft, C. C. Chang, F. Maghoul, and R. Kumar. Searching with context. In *WWW '06: 15th Int. Conf. on World Wide Web*. ACM, 2006.
6. S. Lawrence. Context in web search. *IEEE Data Eng. Bull.*, 23(3):25–32, 2000.
7. K. Q. Pu and X. Yu. Keyword query cleaning. *PVLDB*, 1(1):909–920, 2008.
8. Lance A. Ramshaw and Mitchell P. Marcus. Text chunking using transformation-based learning. *CoRR*, 1995.
9. K. M. Risvik, T. Mikolajewski, and P. Boros. Query segmentation for web search. 2003.
10. A. Shepitsen, J. Gemmell, B. Mobasher, and R. Burke. Personalized recommendation in social tagging systems using hierarchical clustering. ACM, 2008.
11. B. Tan and F. Peng. Unsupervised query segmentation using generative language models and wikipedia. In *WWW*. ACM, 2008.
12. B. Tan and F. Peng. Unsupervised query segmentation using generative language models and wikipedia. ACM, 2008.
13. A. Uzuner, B. Katz, and D. Yuret. Word sense disambiguation for information retrieval. AAAI Press / The MIT Press, 1999.
14. E. Vorhees. The trec-8 question answering track report. In *Proceedings of TREC-8*, 1999.
15. X. Yu and H. Shi. Query segmentation using conditional random fields. ACM, 2009.

---

[7] A prototype of our progress in this regard is available at http://sina.aksw.org.

Facetted Browsing of Extracted Fusion Tables Data for Digital Cities

Gianluca Quercini[1], Jochen Setz[2], Daniel Sonntag[2], and Chantal Reynaud[1]

[1] Laboratoire de Recherche en Informatique
Université Paris Sud XI, Orsay, France
[2] German Research Center for Artificial Intelligence (DFKI)

Abstract. Digital cities of the future should provide digital information about points-of-interest (POIs) for virtually any user context. Starting from several *Google Fusion* tables about city POIs, we extracted and transferred useful POI data to RDF to be accessible by SPARQL requests. In this initial application context, we concentrated on museum and restaurant resources as the result of a precision-oriented information extraction part. With the current application system we are able to retrieve, filter, and order digital cities POI data in multiple ways. With the help of facets, users can do more than just browsing the museums and restaurants. They can filter the relevant objects according to available metadata criteria such as city, country, and POI categories. Different views allow us to visualize the objects of interest as tables, thumbnails, or POIs on an interactive map. In addition, any complementary information on the cities where museums and restaurants are located are retrieved from DBpedia and displayed at query time.

1 Introduction

Digital Cities of the Future should feature a democratic city space through a citizen-centric model, which is the vision of the EIT action line Digital Cities[3]. Citizen participation could take different forms, e.g., the execution of necessary actions to improve the city's performance and sustainability, or, as in the direction we pursue, the collection and usage of data to be broadcast, or used to analyse and sense the status and the dynamics of the city as a place where people live and spend their spare-time. As part of the EIT ICT Labs KIC activity DataBridges, Data Integration for Digital Cities, we are developing a framework that enables the enrichment of data related to points-of-interests (POIs) in cities (e.g., restaurants, museums, or theatres) and supports the applications which aim at using the data to provide specific and dynamic city services (e.g., city tour recommender systems). We are confronted with two major challenges on which we will focus in this demo paper:

- Collecting as much data as possible about digital city POIs.
- Organizing the data into facets so that it can be easily browsed.

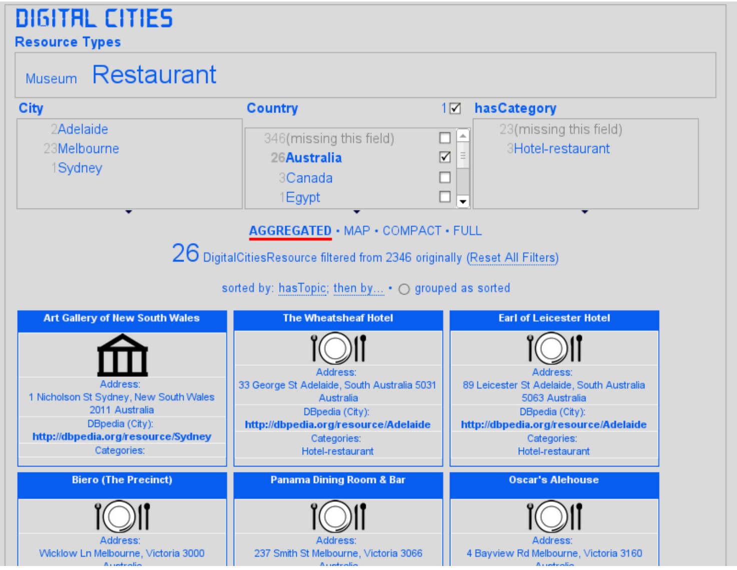
---

[3] `http://eit.ictlabs.eu/action-lines/`

2 Which Facetted Browsing Functionality do We Provide?

Starting from several Google Fusion tables, the data is being transferred to RDF to be accessible by SPARQL requests. We first concentrated on museum and restaurant resources. With the current demonstrator, we are able to filter and order digital cities in multiple ways, rather than in a single, pre-determined, taxonomic order. With the help of automatically generated facets, users can browse museums and restaurants in an elegant way, but also filter the relevant objects according to the available metadata criteria: city, country, and/or POI categories. Different views allow us to visualize the objects of interest as tables, thumbnails, and points-of-interest on an interactive map. In addition, any complementary information on the cities where museums and restaurants are located are retrieved from DBpedia and displayed at query time.

Figure 1 shows digital cities results of restaurants and museums in Australia. The facets in the upper part allow us to filter the city, country, and category resources (here restaurant types) as indicated. Different types of resources are aggregated into the result lenses shown in the lower part of Figure 1 which displays one museum and five restaurant results.


DIGITAL CITIES
Resource Types
Museum
Restaurant
City
Country
hasCategory
AGGREGATED • MAP • COMPACT • FULL
(Reset All Filters)
sorted by: hasTopic; then by... • ○ grouped as sorted

City:
2 Adelaide
23 Melbourne
1 Sydney

Country: 1 ☑
346 (missing this field) ☐
26 **Australia** ☑
3 Canada ☐
1 Egypt ☐

hasCategory:
23 (missing this field)
3 Hotel-restaurant

26 DigitalCitiesResource filtered from 2346 originally

Art Gallery of New South Wales
Address:
1 Nicholson St Sydney, New South Wales 2011 Australia
DBpedia (City):
http://dbpedia.org/resource/Sydney
Categories:

The Wheatsheaf Hotel
Address:
33 George St Adelaide, South Australia 5031 Australia
DBpedia (City):
http://dbpedia.org/resource/Adelaide
Categories:
Hotel-restaurant

Earl of Leicester Hotel
Address:
89 Leicester St Adelaide, South Australia 5063 Australia
DBpedia (City):
http://dbpedia.org/resource/Adelaide
Categories:
Hotel-restaurant

Biero (The Precinct)
Address:
Wicklow Ln Melbourne, Victoria 3000

Panama Dining Room & Bar
Address:
237 Smith St Melbourne, Victoria 3066

Oscar's Alehouse
Address:
4 Bayview Rd Melbourne, Victoria 3160

Fig. 1: Facetted browsing of digital city POIs

Our web application allows us to switch between different views of a single filter being applied, namely aggregated result view, map view, compact view, and full view. Figure 2 shows a result map of 26 filtered restaurant resources. The Google map indicates Australian restaurants and museums as landmarks.

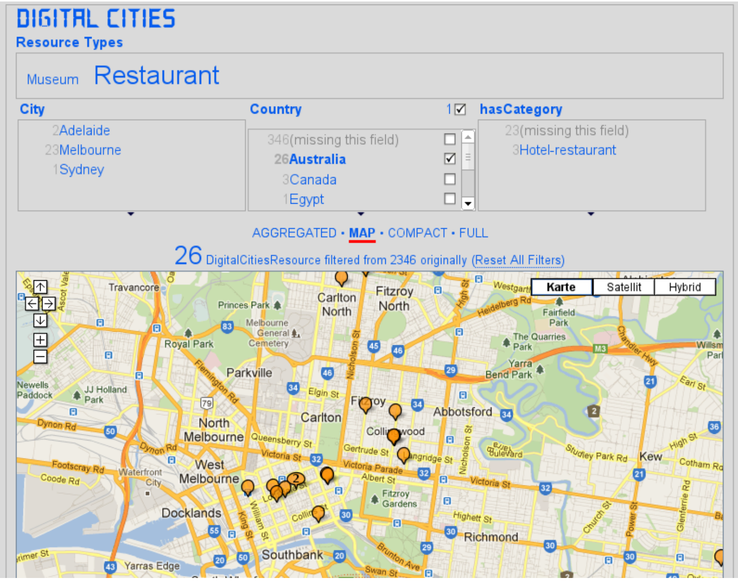


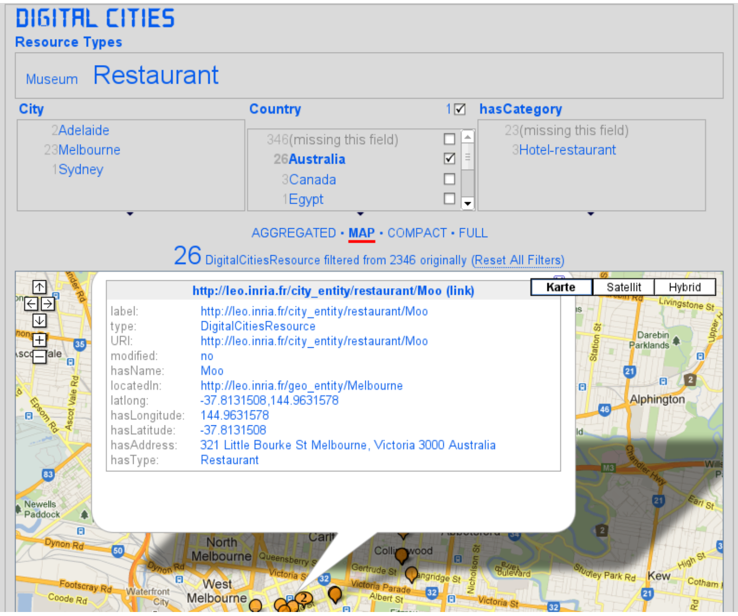
Fig. 2: Digital cities restaurant results indicated on a map

3 Which Features Have We Implemented?

We make use of multiple Linked Data sources; DBpedia contents are extracted according to semantic city links of the filtered resources. Photos related to cities described in DBPedia are provided by the Linked Data resource flickr wrappr [4]. The system's browser page is dynamically generated and updated according to the retrieval sets of the combined Linked Data queries. In addition to interactive boxes, links, and tables, the GUI uses Javascript widgets to visualize individual museum and restaurant retrieval results. External data is provided by a slideshow which enables us to jump directly to the data of interest, here DBpedia comments or Flickr photos. The slideshows can be enabled by selecting the links provided in the aggregated view. Figure 3 shows the selection of a point-of-interest (restaurant) on the map; location-based details of the resource, which are obtained from DBpedia, can be highlighted.

---

[4] `http://www4.wiwiss.fu-berlin.de/flickrwrappr/`



http://leo.inria.fr/city_entity/restaurant/Moo (link)

label:	http://leo.inria.fr/city_entity/restaurant/Moo
type:	DigitalCitiesResource
URI:	http://leo.inria.fr/city_entity/restaurant/Moo
modified:	no
hasName:	Moo
locatedIn:	http://leo.inria.fr/geo_entity/Melbourne
latlong:	-37.8131508,144.9631578
hasLongitude:	144.9631578
hasLatitude:	-37.8131508
hasAddress:	321 Little Bourke St Melbourne, Victoria 3000 Australia
hasType:	Restaurant

Fig. 3: Map details of one of the 26 filtered digital cities results (restaurants)

4 How Does the System Work Technically?

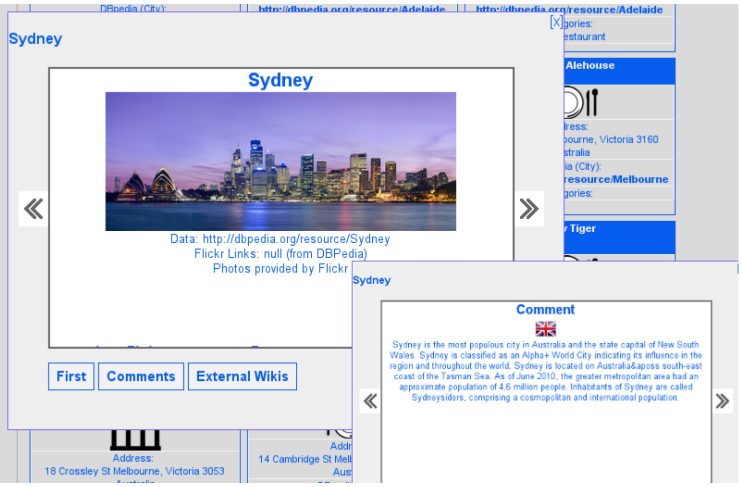
Google Fusion Tables (GFT) are data tables which are consolidated into a Web application that hosts a vast collection of tables contributed by people over the Internet [4]. We developed a tool that automatically converts these tables to RDF data. This problem of understanding the semantics of tables has already been addressed by numerous research groups [6,8,14]. In particular the approaches described in [6,14], which rely on probabilistic models, show promising results. Based on our extraction process from GFT tables, we implemented a graphical user interface from scratch by using the open-source knowledge management tool Exhibit[5]. After the facets and lenses have been specified, several GFT tables are converted to RDF and loaded onto our DFKI Virtuoso server. Note that the set of GFT from which data is obtained is pre-determined off-line; as a result, to add further data to the application we would need to manually obtain another set of tables. An interesting improvement of the application would consist in having tables loaded dynamically as they are added to GFT.

The interactive GUI then triggers several SPARQL queries (at query-time) and provides additional DBpedia information about cities of filtered restaurants and museums in multiple languages (according to established DBpedia links). Additionally, we use the web service of *flickr wrappr* at query-time in order to retrieve RDF links to relevant photos of DBpedia resources. Figure 4 shows

---

[5] `http://www.simile-widgets.org/exhibit/`

the slideshow which we built from the tables. A click on a city location where restaurants or museums are located triggers an ad-hoc query to DBpedia to fetch more information about the city. The slideshow contains comments, external links, and photos from flickr.



Data: http://dbpedia.org/resource/Sydney
Flickr Links: null (from DBPedia)
Photos provided by Flickr

Sydney is the most populous city in Australia and the state capital of New South Wales. Sydney is classified as an Alpha+ World City indicating its influence in the region and throughout the world. Sydney is located on Australia&aposs south-east coast of the Tasman Sea. As of June 2010, the greater metropolitan area had an approximate population of 4.6 million people. Inhabitants of Sydney are called Sydneysiders, comprising a cosmopolitan and international population.

Address:
18 Crossley St Melbourne, Victoria 3053

Fig. 4: Slideshow built from Fusion tables, DBpedia data and flickr photos

4.1 Data Extraction from Google Fusion Tables

All tables in GFT have a relational database structure. Each table is identified by a unique name, which is an alphanumeric string in GFT being automatically assigned to the table upon its creation. Moreover, each column must have a *name* (or *header*) and a *type*, of which GFT defines four: TEXT, NUMBER, LOCATION and DATE. In essence, tables in GFT have three major advantages over other tabular data (spreadsheets, HTML tables, HTML lists) that can be found on the Web:

- GFT tables have a very simple and neat structure. Columns in a GFT table do not branch into several sub-columns, like in spreadsheets;
- The columns of GFT tables are usually typed, which makes easier the semantic annotation of data;
- GFT provides a simple, efficient, and well-documented API that allows applications to query, create, delete, and update tables by using the *Standard Query Language*.

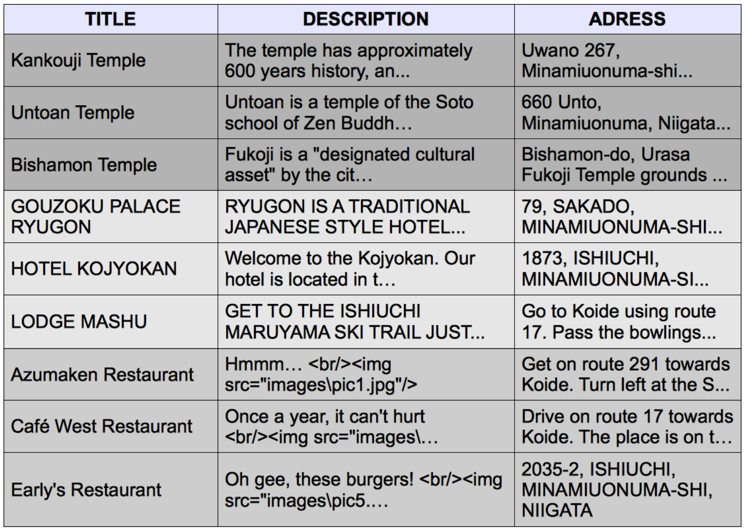
In order to extract data for the Digital Cities facetted browsing context from GFT tables, we first created an ontology which describes major Digital

Cities POIs, such as restaurants and museums in our specific case. The ontology is needed to drive the extraction of important data, such as the name of the POI, its location, contact information as well as its category (e.g. *archaeological museum*, *Italian restaurant*).

The second step consists of selecting GFT tables that contain data about museums and restaurants. The Google web search engine indexes the tables in GFT, which means that they can be searched in the same way as regular Web pages. Therefore, a search for “restaurant” returns all tables in GFT that contain the keyword ”restaurant”. However, the mere fact that a table T mentions the keyword “restaurant” does not necessarily imply that the table has actually data on restaurants.

One of the following may occur:

- T mentions the word “restaurant” only incidentally and therefore has no data on restaurants at all.
- T has data on restaurants along with data on other entities.
- T is entirely dedicated to restaurants.



TITLE	DESCRIPTION	ADRESS
Kankouji Temple	The temple has approximately 600 years history, an...	Uwano 267, Minamiuonuma-shi...
Untoan Temple	Untoan is a temple of the Soto school of Zen Buddh…	660 Unto, Minamiuonuma, Niigata...
Bishamon Temple	Fukoji is a "designated cultural asset" by the cit…	Bishamon-do, Urasa Fukoji Temple grounds ...
GOUZOKU PALACE RYUGON	RYUGON IS A TRADITIONAL JAPANESE STYLE HOTEL...	79, SAKADO, MINAMIUONUMA-SHI...
HOTEL KOJYOKAN	Welcome to the Kojyokan. Our hotel is located in t…	1873, ISHIUCHI, MINAMIUONUMA-SI...
LODGE MASHU	GET TO THE ISHIUCHI MARUYAMA SKI TRAIL JUST...	Go to Koide using route 17. Pass the bowlings...
Azumaken Restaurant	Hmmm… <br/><img src="images\pic1.jpg"/>	Get on route 291 towards Koide. Turn left at the S...
Café West Restaurant	Once a year, it can't hurt <br/><img src="images\…	Drive on route 17 towards Koide. The place is on t…
Early's Restaurant	Oh gee, these burgers! <br/><img src="images\pic5.…	2035-2, ISHIUCHI, MINAMIUONUMA-SHI, NIIGATA

Fig. 5: Excerpt of a GFT table with data on POIs

Figure 5 shows an excerpt of a GFT table which contains information on heterogeneous POIs: temples (first three rows), hotels (the three rows in the middle) and restaurants (the last three rows). Therefore, our extraction algorithm needs to process the table row by row to select those that have the desired information. As shown in Figure 6, the extraction algorithm takes as an input a table T and a list of types of POIs from our ontology and:

- Identifies the *rows* of T that contain information on POIs of any of the input types.
- Determines the *cells* that contain the *names* of those POIs.

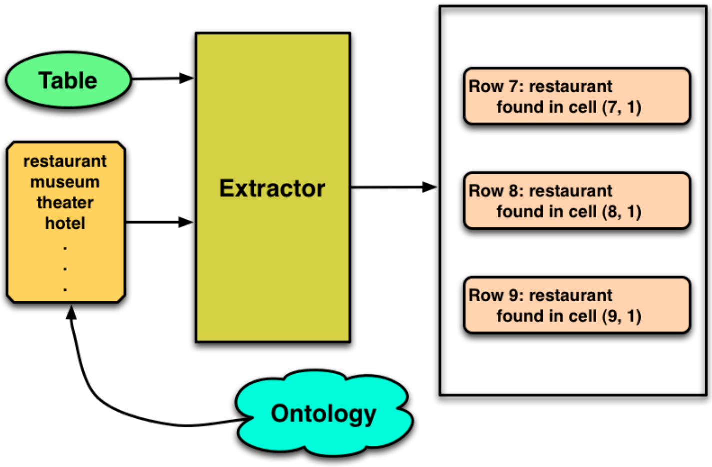


Table
restaurant
museum
theater
hotel
.
.
.
Extractor
Ontology
Row 7: restaurant found in cell (7, 1)
Row 8: restaurant found in cell (8, 1)
Row 9: restaurant found in cell (9, 1)

Fig. 6: Generalized extraction algorithm

As an example, our algorithm correctly identifies that the last three rows of the table shown in Figure 5 have information on restaurants and that the name of those restaurants are in the first column.

Our extraction algorithm goes through three steps:

1. **Pre-processing.** The cells that are not likely to contain names of POIs of the given types are ruled out. This is done by looking at the syntactic properties of the content of each cell, as well as the GFT types of the columns in which they occur.
2. **Annotation.** The content of the remaining cells is submitted to a Web search engine in order to obtain short textual descriptions that are used to determine whether the cells contain names of the POIs of the given types.
3. **Extraction.** The columns that provide the values for the attributes of the POIs are selected and information is extracted.

At the *pre-processing* step, our algorithm rules out the following cells:

- Cells that contain values that follow a certain pattern, that is usually captured by regular expressions: phone numbers, URLs, email addresses, numeric values and geographic coordinates.
- Cells containing long values, such as verbose descriptions (e.g., those in the second column in Figure 5).

- Cells that belong to columns with a specific GFT type, such as LOCATION, DATE and NUMBER.

At the *annotation* step, our algorithm resorts to a Web search engine to understand whether the cells that have not been ruled out at pre-processing contain names of POIs. More specifically, our algorithm submits the content of a cell $T(i, j)$ to the *Bing* [6] search engine and uses the top-10 results as an additional external context to understand the content of the cell itself. Each result returned by the search engine consists of a link to a Web page and a short description (referred to as *snippet*) of the page itself. Our algorithm uses a multi-class text classifier to determine whether a snippet is the description of a POI of a certain type t (for instance, "restaurant"); if the majority of the snippets returned by querying the search engine with the content of $T(i, j)$ are classified as descriptions of restaurants, then $T(i, j)$ is considered as containing the name of a restaurant.

Finally, we need to select the columns that provide values for the attributes or properties of restaurants. The identification of the columns is driven by our ontology, which includes the prominent properties of POIs; in the case of restaurants, these are *name*, *address*, *average price*, *category*, *telephone number* and *website*. In order to match each attribute with the corresponding column in T, if any, we use simple heuristics. The column with the *name* is retrieved while filtering the rows of T; as for *telephone number* and *website*, we use regular expressions; the *address* is usually in a column with type LOCATION and is parsed with a online geotagger such as *Yahoo! PlaceFinder*; finally, the column with the *average price* (respectively, *category*) is considered to be the one with title *price* (respectively, *category* or *type*). Once the columns containing the attributes are determined, the data in the selected rows are extracted from the table and inserted in the ontology.

With this procedure we extracted data on 1500 restaurants, 500 museums, 160 theatres, 67 hotels and 109 schools; only the data on restaurants and museums are already available in our application system.

An evaluation shows the performance of our algorithm to determine whether a cell contains the name of a POI of a given type based on the snippets returned by the search engine. The evaluation has been performed on 40 GFT tables, containing information on POIs with 5 different types: restaurants, museums, theatres, hotels, and schools. In total, we have 287 references to restaurants, 240 to museums, 160 to theatres, 67 to hotels, and 109 to schools. Each table has been annotated manually to obtain a ground truth for our evaluation. We collected names of 300 POIs for each type under examination from DBPedia and used them to retrieve snippets from Bing that we used to train two multi-class text classifiers: a SVM and a Naive Bayes classifier.

The results of our evaluation are shown in Table 1: precision is computed as the number of cells correctly identified by the algorithm over the number of cells identified by the algorithm (i.e., how many of the cells identified are identified

---

[6] We chose Bing because it provides an API with less limitations than other Web search engines in terms of query allowance.

Type	Method	Precision	Recall	F-measure
Restaurants	SVM	0.89	0.69	**0.78**
	Bayes	0.59	0.80	0.68
Museums	SVM	0.83	0.82	**0.83**
	Bayes	0.45	0.93	0.61
Theatres	SVM	0.83	0.76	**0.80**
	Bayes	0.34	0.89	0.5
Hotels	SVM	0.74	0.89	**0.81**
	Bayes	0.23	0.92	0.37
Schools	SVM	0.96	0.91	**0.94**
	Bayes	0.75	0.96	0.85

Table 1: Evaluation of the algorithm.

correctly); recall is computed as the number of cells correctly identified by the algorithm over the number of cells that contain the name of a POI (i.e., how many of the cells containing the name of a POI are correctly identified); the f-measure is the harmonic mean of precision and recall.

4.2 Facetted Browsing Web Architecture

The facetted web browsing architecture described in this paper follows a line of location-based mobile application frameworks in which the majority is equipped with a full-fledged Web browser that enables us to provide platform-independent graphical user interfaces (GUIs) by means of DHTML-based Rich Internet Applications (RIA) or the like [5,10].

Work in those mobile application frameworks is often concerned with physical location-based issues at query/interaction-time, e.g., supporting wayfinding with tactile cues [9] or interactive experiences for cyclists [12]. Other work concerns mobile search scenarios and incidental information [1] where contexts such as location and time play major roles in information discovery [2] or the design of Web-based mobile services [11].

The Digital Cities facetted web architecture follows a different usage strategy: the ubiquitous access to location-based information from virtually every browser with new information repositories which address POI related information needs on the large scale. Previously, in the context of the query/interaction-time retrieval needs [13], we used Google Maps Local Search [7] and two REST services provided by GeoNames [3] (i.e., the *findNearbyWikipedia* search and the *findNearbyWeather* search). But when comparing different POIs and planning routes and/or restaurant information via the facetted browsing functionality, we need a spatially inclusive and comprehensive set of POIs and category information automatically extracted from other (online) repositories.

The facetted web browsing is realised as a Java-based web-server which provides a HTTP/ReST semantic webservice to the Digital Cities data. The JavaScript client implements the control logic and compose facets, views and lenses to layout them with help of a huge set of widgets.

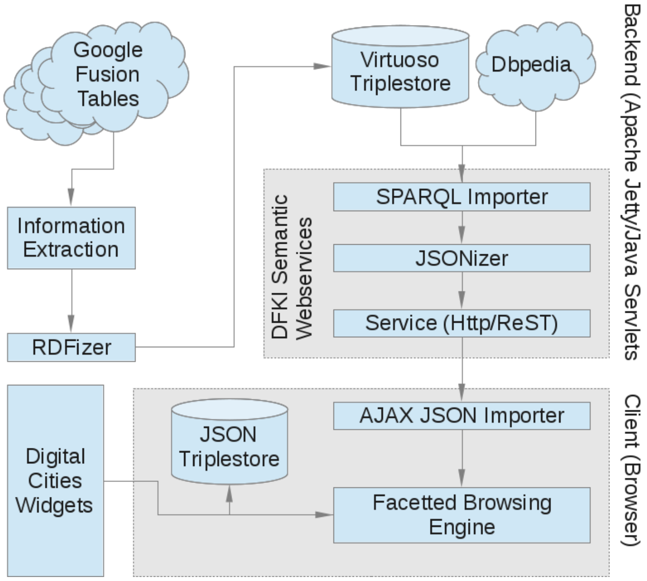


Fig. 7: Technical components of the Facetted Search Web architecture

On start-up, the web-application requests the semantic webservice for available restaurant and museum data. The data is converted and delivered in the Exhibit-JSON format and stored into an In-Memory JavaScript TripleStore provided by the Exhibit framework. The information to be queried for is specified by Exhibit's graph expressions on the Digital Cities RDF graph. Facets to filter the fetched results are defined as HTML-Templates. Filters are defined by Exhibit's path expressions, which allow us to specify paths through an RDF-Graph considering forward and backward properties.

The DBpedia information is provided by the DFKI Semantic webservices which requests the DBpedia SPARQL endpoint at run-time and provide the data as JSON-strings to our semantic widgets. The semantic widgets then sort the DBpedia information about a city by their properties to set up an interactive slideshow. The DFKI semantic webservices in the backend do not hold any application logic, but rather a model-based logic to fetch, store, compose, and convert RDF-data. The RDFizer imports GFT tables from the Information Extraction into RDF triples (NT-Notation) to store the triples into a local TripleStore (Virtuoso) on the server. Virtuoso's native JDBC Interface is used to make the data accessible by SPARQL queries.

Finally, a SPARQL importer fetches RDF data from the DBpedia SPARQL endpoint and the local endpoint by resource-type or URI. Our semantic webservices also provides a converter from RDF into the Exhibit-JSON format. The service component wraps the query mechanism into an application-specific "restful" interface to request for structured museum, restaurant, and city information.

5 Conclusion

The Digital Cities facetted web browsing architecture follows the idea of providing ubiquitous access to location-based information from virtually every browser with new information repositories which address POI related information needs on the large scale. In this paper, we described our live system: starting from several GFT tables about city POIs, we extracted and transferred useful POI data to RDF to be accessible by SPARQL requests in the context of an interactive facetted browsing application. In particular, different views allow us to visualize the objects of interest as tables, thumbnails, or POIs on an interactive map or slideshow which also takes dynamic data from other Linked data sources (DBpedia and flickr) into account. Knowledge extraction from these structured and semi-structured documents on the Web should now be complemented with a special focus on scalability. The facetted browsing tool would highly benefit from further GFT tables to be extracted automatically and used in the context of a digital city search application. Besides further structured data cells, we plan to automatically extract information from textual data cells in the near future. Thereby, the structured data cells of a specific record should improve the performance of the text mining processes of the related unstructured data cells. A demo of the Digital Cities facetted web search can be found at the following URL: `http://digitaleveredelung.lolodata.org:8080/DigitalCities/page/index.html`.

6 Acknowledgements

This research has been supported in part by the THESEUS Programme in the Core Technology Cluster WP4, which is funded by the German Federal Ministry of Economics and Technology (01MQ07016) and the EIT ICT labs activities 2012 The responsibility for this publication lies with the authors.

References

1. Arter, D., Buchanan, G., Jones, M., Harper, R.: Incidental Information and Mobile Search. In: Proceedings of the 9th International Conference on Human Computer Interaction with Mobile Devices and Services. pp. 413–420. MobileHCI '07, ACM, New York, NY, USA (2007)
2. Church, K., Neumann, J., Cherubini, M., Oliver, N.: SocialSearchBrowser: a Novel Mobile Search and Information Discovery Tool. In: Proceedings of the 15th International Conference on Intelligent User Interfaces. pp. 101–110. IUI '10, ACM, New York, NY, USA (2010)
3. GeoNames: WebServices Overview (Jul 2010), `http://www.geonames.org/export/ws-overview.html`
4. Gonzalez, H., Halevy, A.Y., Jensen, C.S., Langen, A., Madhavan, J., Shapley, R., Shen, W., Goldberg-Kidon, J.: Google Fusion Tables: Web-centered Data Management and Collaboration. In: Proceedings of the 2010 International Conference on Management of Data. pp. 1061–1066. SIGMOD '10, ACM, New York, NY, USA (2010)

5. Gruenstein, A., McGraw, I., Badr, I.: The WAMI Toolkit for Developing, Deploying, and Evaluating Web-accessible Multimodal Interfaces. In: Proceedings of the 10th International Conference on Multimodal Interfaces. pp. 141–148. ICMI '08, ACM, New York, NY, USA (2008)
6. Limaye, G., Sarawagi, S., Chakrabarti, S.: Annotating and Searching Web Tables Using Entities, Types and Relationships. Proc. VLDB Endow. 3, 1338–1347 (September 2010)
7. Mapki: (Jul 2010), `http://mapki.com/`
8. Mulwad, V., Finin, T., Syed, Z., Joshi, A.: Using Linked Data to Interpret Tables. In: First International Workshop on Consuming Linked Data (COLD2010) (2010)
9. Pielot, M., Henze, N., Boll, S.: Supporting Map-based Wayfinding with Tactile Cues. In: Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services. pp. 23:1–23:10. MobileHCI '09, ACM, New York, NY, USA (2009)
10. Porta, D., Sonntag, D., Neßelrath, R.: A Multimodal Mobile B2B Dialogue Interface on the iPhone. In: Proc. 4th Workshop on Speech in Mobile and Pervasive Environments (SiMPE) (2009)
11. Riva, C., Laitkorpi, M.: Designing Web-Based Mobile Services with REST. In: Nitto, E., Ripeanu, M. (eds.) Service-Oriented Computing - ICSOC 2007 Workshops, pp. 439–450. Springer-Verlag, Berlin, Heidelberg (2009)
12. Rowland, D., Flintham, M., Oppermann, L., Marshall, J., Chamberlain, A., Koleva, B., Benford, S., Perez, C.: Ubikequitous Computing: Designing Interactive Experiences for Cyclists. In: Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services. pp. 21:1–21:11. MobileHCI '09, ACM, New York, NY, USA (2009)
13. Sonntag, D., Porta, D., Setz, J.: HTTP/REST-based Meta Web Services in Mobile Application Frameworks. In: Proceedings of the 4nd International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM-10). pp. 170–175. IARIA/XPS (Xpert Publishing Services) (2010)
14. Venetis, P., Halevy, A., Madhavan, J., Paşca, M., Shen, W., Wu, F., Miao, G., Wu, C.: Recovering Semantics of Tables on the Web. Proc. VLDB Endow. 4, 528–538 (2011)

Bringing Newsworthiness into the 21st Century

Tom De Nies[1], Evelien D'heer[2], Sam Coppens[1], Davy Van Deursen[1], Erik Mannens[1], Steve Paulussen[2], and Rik Van de Walle[1]

[1] Ghent University - IBBT - ELIS - Multimedia Lab
{tom.denies,sam.coppens,davy.vandeursen,erik.mannens,rik.vandewalle}@ugent.be
[2] Ghent University - IBBT - MICT
{evelien.dheer,steve.paulussen}@ugent.be

Abstract. Due to the ever increasing flow of (digital) information in today's society, journalists struggle to manage and assess this abundance of data in terms of their newsworthiness. Although theoretical analyses and mechanisms exist to determine if something is worth publishing, applying these techniques requires substantial domain knowledge and know-how. Furthermore, the consumer's need for near-immediate reporting significantly limits the time for journalists to select and produce content. In this paper, we propose an approach to automate the process of newsworthiness assessment, by applying recent Semantic Web technologies to verify theoretically determined news criteria. We implemented a proof-of-concept application that generates a newsworthiness profile for a user-specified news item. A preliminary evaluation by manual verification was performed, resulting in 83.93% precision and 66.07% recall. However, further evaluation by domain experts is needed. To conclude, we outline the possible implications of our approach on both the technical and journalistic domains.

Keywords: News, Newsworthiness, Semantic Web, Relevance, Named Entity Recognition

1 Introduction

Every day, journalists have to choose from numerous items to decide which events qualify for inclusion in the media, as they cannot all fit the available space. The media act as 'gatekeepers', who control and moderate the access to news and information. In today's society, connected by the Internet, the amount and flow of information and communication has increased dramatically. In addition, users become involved as creators and distributors of content. The abundance of (digital) information makes it difficult for journalists to manage and assess the events in terms of their alleged newsworthiness.

Within the scholarly field of journalism studies, the selection of news by the media is said to be influenced by a range of factors. As specified in Sect. 2, several criteria have been determined over time to assess news value. However, most of these are based on purely human analysis and reasoning. At the time they were created, large-scale empirical evaluation of these criteria would have been extremely labor intensive, if not impossible. Now, recently developed techniques and services in information technology enable us to revisit newsworthiness research, and construct an automated approach to help journalists in assessing news value. Our goal is to provide a tool that will help journalists in both news selection and reader targeting, and additionally, will aid sociological researchers in verifying and determining news selection criteria.

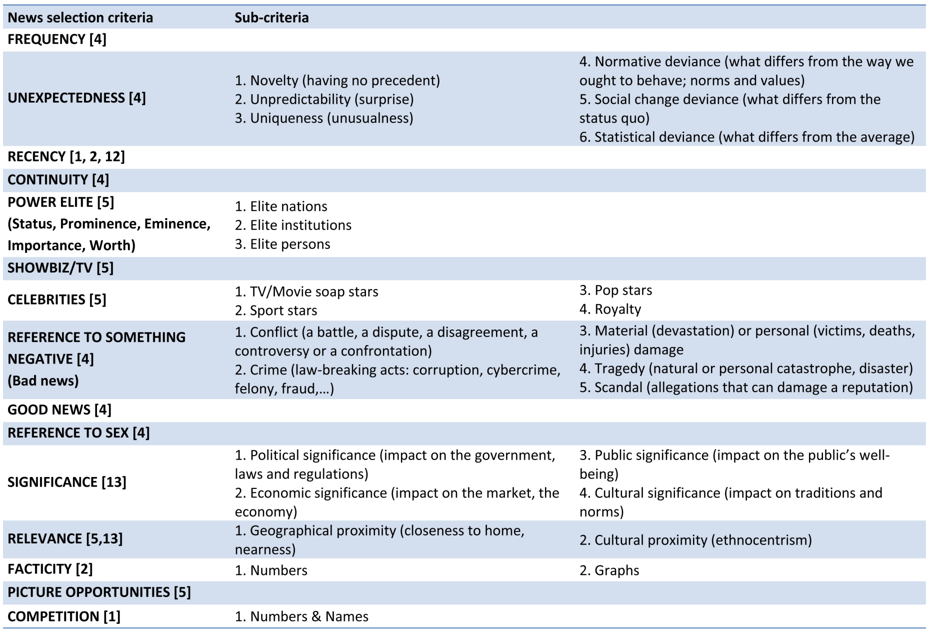
The remainder of the paper is structured as follows: first, we provide a sociological perspective on news value assessment. Next, an overview is given of our proposed approach to revive this perspective using state-of-the-art information processing techniques. We break down the approach into its components, and provide details about each type of analysis, including how it was realized in the proof-of-concept implementation. Finally, a first evaluation of the automatic approach is performed, and the possible improvements and future research opportunities are discussed.

2 A Sociological Perspective

Sociological research on the process of news selection in newsrooms has resulted in various overviews of 'news values' or 'news selection criteria'. The concept of newsworthiness is built on the assumption that certain events get selected by media above others based on the attributes or 'news values' they possess. The more of these news values are satisfied, the more likely an event will be selected. If an event lacks one news value, it can compensate by possessing another. Hence, journalists' criteria for selecting the news are cumulative, making stories significant based on their overall level of newsworthiness. The earliest attempt for a systematic approach of determining newsworthiness by news values, is a taxonomy by Galtung & Ruge [1], that triggered both scholars and practitioners in examining aspects of events that make them more likely to receive coverage [2, 3]. For analytical purposes, the concept of news values is valuable to understand that news selection is more than just the outcome of journalists' 'gut feeling'. To decide what is news and what is not, journalists consciously and unconsciously use a set of selection criteria, that help them assess the newsworthiness of a story or an event [4, 5].

In Table 1, an overview is given of the different news values that are available in the literature of journalism studies, categorized in news values and their subfactors. Here, we will give a brief review of the most important studies in this domain.

In the 1960's, Galtung & Ruge [1] published a theory of news selection, which provided a taxonomy of 12 news values that define how events become news. More specifically, they discerned (1) *frequency*: the time-span of the event to unfold itself, (2) *threshold*: the impact or intensity of an event, (3) *unambiguity*: the clarity of the event (4) *meaningfulness*: the relevance of the event, often in terms of geographical proximity and cultural similarity, (5) *consonance*: the way the event fits with the expectations about the state of the world, (6) *unexpectedness*: the unusualness of the event, (7) *continuity*: further development of a



News selection criteria	Sub-criteria	
FREQUENCY [4]		
UNEXPECTEDNESS [4]	1. Novelty (having no precedent) 2. Unpredictability (surprise) 3. Uniqueness (unusualness)	4. Normative deviance (what differs from the way we ought to behave; norms and values) 5. Social change deviance (what differs from the status quo) 6. Statistical deviance (what differs from the average)
RECENCY [1, 2, 12]		
CONTINUITY [4]		
POWER ELITE [5] (Status, Prominence, Eminence, Importance, Worth)	1. Elite nations 2. Elite institutions 3. Elite persons	
SHOWBIZ/TV [5]		
CELEBRITIES [5]	1. TV/Movie soap stars 2. Sport stars	3. Pop stars 4. Royalty
REFERENCE TO SOMETHING NEGATIVE [4] (Bad news)	1. Conflict (a battle, a dispute, a disagreement, a controversy or a confrontation) 2. Crime (law-breaking acts: corruption, cybercrime, felony, fraud,...)	3. Material (devastation) or personal (victims, deaths, injuries) damage 4. Tragedy (natural or personal catastrophe, disaster) 5. Scandal (allegations that can damage a reputation)
GOOD NEWS [4]		
REFERENCE TO SEX [4]		
SIGNIFICANCE [13]	1. Political significance (impact on the government, laws and regulations) 2. Economic significance (impact on the market, the economy)	3. Public significance (impact on the public's well-being) 4. Cultural significance (impact on traditions and norms)
RELEVANCE [5,13]	1. Geographical proximity (closeness to home, nearness)	2. Cultural proximity (ethnocentrism)
FACTICITY [2]	1. Numbers	2. Graphs
PICTURE OPPORTUNITIES [5]		
COMPETITION [1]	1. Numbers & Names	

Table 1. News selection criteria as identified and specified in literature.

previous newsworthy story, (8) *composition*: a mixture of different kind of news, (9) *reference to elite nations*, (10) *reference to elite persons*, (11) *reference to persons*: events that can be made personal, (12) *reference to something negative*. Bell [6] used Galtung and Ruge's list as a starting point, but redefined some of them and added the value of facticity: a good story needs facts (e.g. names, locations, numbers and figures). Harcup and O'Neill [2] also made a revision of Galtung and Ruge's criteria for the contemporary news offer, with special attention for the entertainment offer available in newspapers. More specifically, the following values were added: (1) Events with *picture opportunities*, (2) *Reference to sex*, (3) *Reference to animals*, (4) *humor*, (5) *showbiz/TV* related events and (6) *good news* (e.g. acts of heroism). In addition, concerning elite persons they made a distinction between *power elite* (e.g. Prime minister) and *celebrities* (e.g. Pop Stars). McGregor [7] also proposed news values reflecting the modern way of news selection, with a focus on TV news. He referred to events that are *visually* accessible and recordable. In addition, when events involve tragedy, victims or children, it is likely to appeal to the *emotions* of the audience. Concerning negative events, *conflict*, *scandal* and *crime* are highlighted. Bekius [8] added *competition* as a value, which explains the extensive coverage of sports. Finally, Shoemaker and Cohen [3] extended the notion of significance, by demarcating four sub-dimensions: *political* significance, *economical* significance, *cultural* significance and *public* significance.

3 A Technical Perspective

While the aspects of news as described in Sect. 2 provide a valuable guideline for the theoretical analysis of news, this remains a task for specialists, labor intensive and prone to human error. However, recent advances in Semantic Web technologies have made new tools and services available on the Web to automatically analyze the content of an article. In Figure 1, these technologies are mapped to the news values from Table 1 they can be used to detect. Each of these analyses will be discussed in detail in Sect. 4. The mapping is structured as follows: each straight, full line indicates that an analysis or detected entity or topic contributes to the connecting news criteria in some way. The dotted, curved lines represent the subdivision of news criteria into several sub-criteria, which are then connected to one or more analyses by full lines. The attentive reader will notice that there are two news criteria in the mapping that were not present in the literature overview in Table 1: *social significance* and *technological significance*. This is because when implementing our approach, we noticed a significant number of articles that complied with these criteria, and had no corresponding news value mapping. Therefore, we added these news values ourselves. Whether they can actually be considered news selection criteria in the newsrooms, remains to be investigated by extensive social research. However, our approach will provide the ideal means to this end.

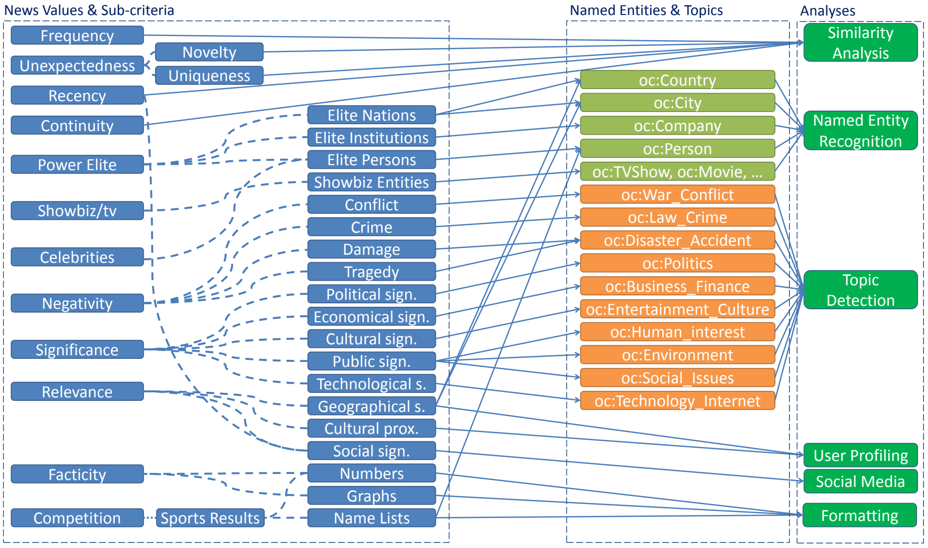


Fig. 1. News values and sub-criteria mapped to Named Entities, Topics and analyses using Semantic Web technologies

4 Determining Newsworthiness

The aim of our research is to provide users (news professionals, such as journalists and media researchers) with information about the newsworthiness of an arbitrary news article. However, it would not be feasible, nor desirable for our system to output a single newsworthiness score. Instead, a score between 0 and 1 will be generated for each news value discussed in Sect. 2. This way, a *newsworthiness profile* is generated, based on the content of the article. To achieve this, we take the content input (plain text), and analyze it in different ways. The results of this analysis are then traced back to the news values, using the mapping illustrated in Figure 1. This mapping leads to six analysis 'clusters', namely: similarity analysis, Named Entity Recognition (NER), topic detection, social media, reader targeting and content format analysis. Each analysis adds a score to one or more corresponding values. After the final analysis, a normalization is performed before returning the final result.

4.1 Named Entity Recognition

A crucial step in our analysis is gathering as much descriptive metadata about the news articles as possible. However, manual addition of metadata by the authors is often neglected. Therefore, we need to automatically generate these metadata ourselves. To do this, we use publicly available tools known as Named Entity Recognition(NER) services. These services accept plain text as input, and output a list of linked Named Entities (NEs), detected in the text. For our implementation, our NER service of choice is OpenCalais[3], a well-established, thoroughly tested [9] and freely available NER service. However, it is our goal to include additional NER services over time, thereby linking to more nodes in the Linked Data cloud, and creating a more complete mapping of Linked Data resources to news values. For a thorough overview of NER services and their performance, we refer to the NERD framework, by Rizzo & Troncy [10]. At the time of writing, OpenCalais is able to detect 21 types of entities from plain text, of which a selection is shown in Figure 1. A complete list of these entity types is provided in the online OC documentation[4]. Conforming to the mapping in Figure 1, these entity types contribute to the *eliteness*, *geographical significance*, *showbiz/TV* and *celebrities* criteria. The amount at which a detected entity contributes to its respective news value(s) is determined by the *relevance score* assigned by OpenCalais. This relevance score represents the importance of an entity in the text it occurs in. For each entity detected in the text, the relevance score is accumulated in the corresponding news value(s).

---

[3] http://www.opencalais.com

[4] http://www.opencalais.com/documentation/linked-data-entities

4.2 Topic Detection

As an addition to the NER, OpenCalais also categorizes the article into one of 17 topics[5]. We associate each of these topics to one or more news criteria, as in Figure 1. The topics of the article contribute to the *negativity* and *meaningfulness* criteria. Because these are very broad criteria, we subdivide negativity into *conflict*, *crime*, *damage* and *tragedy*. Similarly, meaningfulness is subdivided into *political*, *economical*, *cultural*, *public* and *geographical significance*. One of the predefined topics OpenCalais detects, technology, occurred frequently during the testing phase of development, and had no straightforward mapping to one of the values. Therefore, a new news value, termed *technological significance*, was added. Whether this new news value is also used by media practitioners, remains to be investigated in future work. Analogous to the relevance scores in Sect. 4.1, OpenCalais assigns a *confidence score* to each topic. This score is added to the corresponding news criteria for each detected topic.

4.3 Similarity Analysis

Next to NER and topic detection, another important component of automated news analysis is identification of any prior mentions of an article or its content. Comparing the content of an article to strategically chosen reference sets gives us information about the *frequency*, *novelty*, *uniqueness* and *recency* of that kind of content. Here, the challenge is to find a good method for retrieving similar articles, as well as suitable reference sets.

For the retrieval of similar articles in a reference set, we have developed a Named Entity based similarity measure. The approach uses the NEs detected in an article to compute a similarity measure. When comparing two articles A and B, we create two vectors representations a and b of their NEs, where a_i is the weight of NE i in document A (analogous for B), as determined during the NER step (in our case, the relevance score assigned by OpenCalais). The similarity between the documents is then calculated as the *cosine similarity* of the vectors, given by Formula 1.

$$Sim(A,B) = \frac{\sum_i a_i b_i}{\sqrt{\sum_i a_i^2}\sqrt{\sum_i b_i^2}} \tag{1}$$

When no NEs were detected, we revert to the classic “bag of words” approach, using *Term Frequency - Inverse Document Frequency (TF-IDF)* weights for every word in the text. For a more extensive description and evaluation of this NE based similarity measure, we refer to [11], where the same method is used for the clustering of a large set of news articles. Here, the NE based similarity is used to select all news articles from a reference set that are more similar to the input article than a certain threshold.

For the selection of the reference sets, it is important to keep in mind which news values the similarity analysis will influence. As seen in Figure 1, the criteria

---

[5] http://www.opencalais.com/documentation/calais-web-service-api/api-metadata/document-categorization

related to similarity are all relative to recent publications. In other words, for this application, it is better to use a smaller and dynamic reference set of recent news items (e.g. one or two weeks), rather than a large news database, gathered over a long period of time. In our implementation, we use the New York Times "Most Popular" API[6] to acquire the the most viewed articles of the last 7 and 30 days to be used as our reference sets.

After retrieving all articles with similarity to the input article above a certain threshold T (in our case, $T = 0.7$), from both reference sets, we use the size of these retrieved sets to make a decision about the related news values. If the article has more than one similar article, using the reference set of the last 7 days, a positive score is added to the *recency* value. If it has no similar articles in the 7-day set, but it has more than one in the 30-day set, the scores of the *frequency* and *continuity* criteria are increased. Finally, if no similar news items were found, a score is added to the *unexpectedness*. The most suitable amount at which the scores are increased will have to be determined by further testing. In the first implementation, a straightforward "+1" is performed.

4.4 Social Media

An important group of actors in today's media landscape that cannot be ignored, are the social media. Social networks such as Twitter and Facebook play a leading role in the spreading of news stories. These media act as a barometer for news trends, and provide the necessary services and APIs to be utilized as such. These APIs allow us to measure the relevance and public significance of a certain news item, as a form of crowdsourcing. The social media are important indicators of the newsworthiness of an article. Once a topic is trending on Twitter and/or Facebook, the information related to this topic is spread widely in a very short timespan. Therefore, we incorporate a social media API in our approach. More specifically, we cross-reference the Named Entities and topics extracted from the content of the input news item, to trending topics and keywords on Twitter, using the Twitter API[7]. Each trending word or sentence is compared to the string labels of the detected NEs and topics. Additionally, these labels are expanded with their synonyms and semantically similar words, found using a publicly available lexical tool, such as DISCO[8]. If a NE or topic (or one of their synonyms) occurs in the list of currently trending topics on Twitter, the *social significance* and the *recency* score of the news item are increased by one.

4.5 Reader Targeting

While we are aiming towards a general and inherent model of newsworthiness, the influence of the targeted reader's whereabouts and preferences cannot be ignored. Local news might have entirely different news values contributing to

---

[6] http://developer.nytimes.com/docs/read/most_popular_api

[7] http://dev.twitter.com/

[8] http://www.linguatools.de/disco/

its newsworthiness than national or global news. Therefore, we will take these two aspects of the targeted news consumers into consideration. The *cultural proximity* of a news article to a group of users is determined by adding 1 for each entity or topic relating to culture detected in the article, such as TV shows, plays, etc. Similarly, the *geographical proximity* of an article to a user is increased by 1 when specific locations such as cities or regions appear in the article. These criteria give a journalist an indication of which audience his/her article can be targeted towards.

4.6 Content Format Analysis

Research shows that *facticity* plays an important role when it comes to newsworthiness [6]. Although no off-the-shelf techniques are in place for detecting facticity, there are some constructs we can look for. Although far from the only contributing factors to facticity (such as *correctness*, which is beyond the scope of this work), lists and tables are good indicators that facts are being summed up. Lists and tables are elements that are fairly easily recognizable in digital content, due to their notation using a markup language. For example, lists that are included in a news article on a web page will be represented using the HTML <*ul*> tag. However, these HTML lists and tables are often used as a means to create the layout of a web page. To make the distinction from this type of usage, only lists and tables that contain numerical values will contribute to the facticity. Additionally, we look for repetition of certain structural patterns using regular expressions, such as "*:*$" and "-*$" (where '*' represents a wild card, and '$' a new line). Other indicators of facticity are frequent occurrences of currencies (detectable by many NER services), or other numerical values. Each of these detected indicators will contribute to the facticity value in the resulting newsworthiness profile. The amount at which the facticity score is increased, is to be determined empirically.

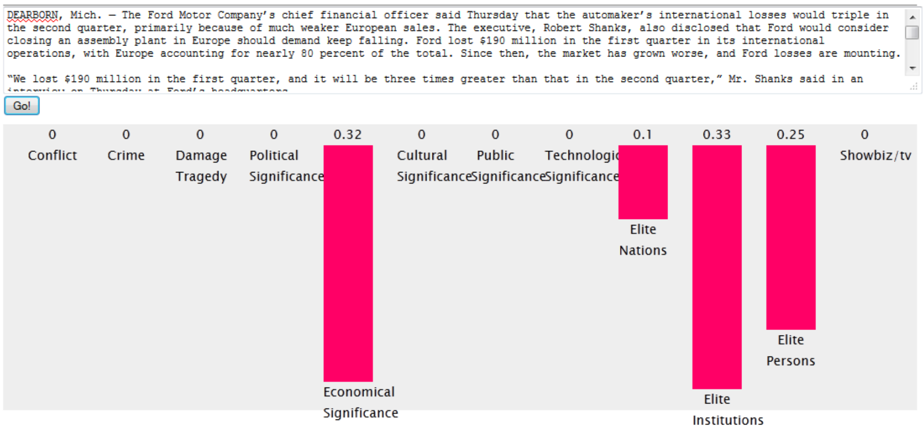
4.7 Normalization and Presentation

When all analyses have been completed, we obtain an array of scores, each linked to a news value. However, these scores have no well-defined bounds, as each type of analysis uses different calculations, precision and weights to add a score to a news value. Therefore, we normalize each news value score s to a normalized score s_n as in Formula 2, where S_0 is the set of non-zero scores in the array.

$$\forall s \in S_0 : s_n = \frac{s}{\sum_{s_i \in S_0} s_i} \tag{2}$$

This implies that the total of all non-zero scores is always 1, and each score indicates the contribution of its news value to the newsworthiness of the article.

Now, the score array is visualized to the user, in the form of a bar chart or similar graphical representation. This way, the user is instantly presented with an overview of the newsworthy aspects of the input article. Figure 2 shows a screen shot of how this is achieved in the proof-of-concept implementation.



DEARBORN, Mich. – The Ford Motor Company's chief financial officer said Thursday that the automaker's international losses would triple in the second quarter, primarily because of much weaker European sales. The executive, Robert Shanks, also disclosed that Ford would consider closing an assembly plant in Europe should demand keep falling. Ford lost $190 million in the first quarter in its international operations, with Europe accounting for nearly 80 percent of the total. Since then, the market has grown worse, and Ford losses are mounting.

"We lost $190 million in the first quarter, and it will be three times greater than that in the second quarter," Mr. Shanks said in an

Fig. 2. Example of visualization of newsworthiness scores in our proof-of-concept application. The analyzed article contains information on the stock of a major automotive company. The software correctly determined the economical significance of the article, and indicates the presence of some important persons, nations and organizations.

5 Evaluation

Until now, analysis of news and assessment of its newsworthiness was performed by domain experts, such as journalists and media researchers. Unfortunately, this means that there are no automated systems to compare to our approach. Therefore, as a first evaluation of our approach, we constructed a gold test set, using publicly available news that is assumed to be newsworthy, and verified the results of our approach manually using the latest version of the proof-of-concept software. Note that even though not all features were implemented yet at the time of writing, we chose to evaluate the output of all implemented features, rather than provide no evaluation at all.

Our test set was created using the New York Times Developer APIs[9]. The New York Times offers a large selection of its data openly for non-commercial use. To obtain a set of articles that can safely be assumed to be newsworthy, we made use of the "Most Popular API". As an initial set, we requested the most viewed articles from the 30 days prior to the submission of this paper (thus ranging from July 8th until August 6th). From this set, we then retained only those URLs which were annotated with more than five concepts, using the "Semantic API". This extra selection step ensures us of suitable data, in which a NER service should stand a good chance of finding Named Entities. Finally, using the "Article Search API", we obtained the summary[10] of each article, and processed it using our approach. The similarity analysis (see Sect. 4.3) was

---

[9] http://developer.nytimes.com/docs/read/{most_popular_api, semantic_API, article_search_api}

[10] Note that the New York Times API does not release the full text of articles.

skipped, since we use the same Most Popular API as a reference set, which would result in each newsworthiness profile containing the "recency" value.

After applying our approach, we obtained a set of 224 articles, each annotated with extracted entities, topics, and a newsworthiness profile. We assessed these newsworthiness profiles manually for correctness ourselves. For each profile, we assessed two things:

1. Are there any news criteria missing in the profile, that are perceived in the article? (*recall*)
2. Are there any excess news criteria in the profile, that are not perceived in the article? (*precision*)

When performed for all newsworthiness profiles, the first assessment gives us the number of profiles without any missing news criteria, or the *recall* of the newsworthiness profiles. The seconds assessment results in the number of profiles without any falsely detected news criteria, or the *precision* of the newsworthiness profiles. After careful evaluation of the 224 newsworthiness profiles, a **precision of 83.93%**, and a **recall of 66.07%** was obtained. The most common mistakes occurred when two very distinct news criteria occurred in the same article. For example, one of the articles describes a football coach harrassing one of his players. This will trigger both the Crime and Competition criteria, while in this case, only Crime is relevant. This could perhaps be solved by defining disjointness rules between the news criteria.

Overall, these preliminary results indicate that we have found the basis for a viable, automated method to extract the contributing news criteria that determine an article's newsworthiness. However, as discussed in the next section, further evaluation by domain experts is still needed when the approach is fully implemented. The processed set of articles, complete with newsworthiness profiles and their evaluations, can be found in machine-understandable JSON format at the following url: `http://users.ugent.be/~tdenies/AVALON/ISWC/evaluation.json`. A graphical interface to browse this data is available at `http://users.ugent.be/~tdenies/AVALON/ISWC/`.

6 Discussion and Future Work

While the initial impression of our approach is positive, a more extensive evaluation will be performed soon. The first evaluation is slightly biased, as it relied on verification of the generated results, because it is a relatively fast and straightforward process. A more thorough, unbiased approach is to ask a panel of domain experts to manually compose a newsworthiness profile, based on the same news values as our approach, and compare it with a profile generated by our approach, using more than one NER service (and corresponding news value mapping). However, before we initiate this extensive, time-consuming evaluation, further optimization of our proof-of-concept software is required, to avoid excess repetition of the process. Once this technological optimization is complete, we will also assess usability with media practitioners in order to discern improvements in terms of ease of use (e.g. the visualization of the newsworthiness scores) and user-perceived reliability (e.g. the professionals' trust in the tool's accuracy). In addition, future research will focus on the question to what extent the selection of news of particular media outlets can be adequately predicted by automatically generated newsworthiness assessments. Such a study has the potential to provide valuable insights as to what extent the inherent subjectivity of human news selection can be approximated by objective parameters and algorithms for newsworthiness assessment.

Our approach also provides an opportunity to gain a new perspective on the research towards news value assessment. The automated approach allows for a large-scale analysis of published works. This way, it becomes possible to verify whether the theoretically determined news criteria still hold for today's media landscape, or even to identify new criteria. As experienced during development, comparing automated and human news selection mechanisms might reveal news values not yet identified in the literature of journalism studies. One specific use case that has recently emerged, is that of citizen journalism. As users become more informed, they often participate actively in the production of news. Examples of this sort of participatory journalism are often found in blogs, opinion pieces and local newspapers. It is to be expected that these citizen journalists will use a distinctly different set of criteria to determine whether something is newsworthy to them. Our approach allows to investigate this objectively.

7 Conclusions

We succeeded in building an application to automatically assess the newsworthiness of a news article, starting from only the content. We used a unique combination of sociological research and technological advances, to create a mapping of long established and recently emerged news values to a set of automated analysis clusters. Through Named Entity Recognition, topic detection, similarity analysis, social media, reader targeting and text format matching, our approach assigns a score to a set of news criteria, to create a newsworthiness profile for an arbitrary piece of text. The generated newsworthiness profiles of 224 news items were manually verified, and exhibited a precision of 83.93% and recall of 66.07%. Of course, further evaluation is needed, in collaboration with domain experts. This approach provides many opportunities for future work, including large-scale analysis of news content by domain experts, potentially uncovering new criteria that contribute to newsworthiness in the current media landscape.

Acknowledgments. The research activities in this paper were funded by Ghent University, IBBT (a research institute founded by the Flemish Government), the Institute for Promotion of Innovation by Science and Technology in Flanders (IWT), the FWO-Flanders, and the European Union, in the context of the IBBT project Smarter Media in Flanders (SMIF). Companies involved are Belga, Concentra, VRT and Roularta, with project support of IWT. We also extend thanks to The New York Times, for their comprehensive open news API.

References

[1] Galtung, J., Ruge, M.: The structure of foreign news. Journal of peace research **2** (1965) 64–90

[2] Harcup, T., O'neill, D.: What is news? Galtung and Ruge revisited. Journalism studies **2** (2001) 261–280

[3] Shoemaker, P., Cohen, A.: News around the world. Content, practitioners, and the public. Recherche **67** (2006) 02

[4] O'sullivan, T., Hartley, J., Saunders, D., Montgomery, M., Fiske, J.: Key concepts in communication and cultural studies. Routledge London (1994)

[5] Schultz, I.: The journalistic gut feeling. Journalism practice **1** (2007) 190–207

[6] Bell, A.: The language of news media. Blackwell Oxford (1991)

[7] McGregor, J.: Terrorism, war, lions and sex symbols: Restating news values. What's news (2002) 111–125

[8] Bekius, W.: Werkboek journalistieke genres. Coutinho (2003)

[9] Iacobelli, F., Nichols, N., Birnbaum, L., Hammond, K.: Finding new information via robust entity detection. In: Proactive Assistant Agents (PAA2010) AAAI 2010 Fall Symposium. (2010)

[10] Rizzo, G., Troncy, R.: NERD: Evaluating Named Entity Recognition Tools in the Web of Data. In: (ISWC'11) Workshop on Web Scale Knowledge Extraction (WEKEX'11). (2011)

[11] De Nies, T., Coppens, S., Van Deursen, D., Mannens, E., Van de Walle, R.: Automatic discovery of high-level provenance using semantic similarity. In: Proceedings of the 4th International Provenance and Annotation Workshop IPAW 2012, LNCS 7525, Springer, Heidelberg. (2012) 97–110