Steffen Lohmann (Ed.)

I-SEMANTICS 2013
Posters & Demos

Proceedings of the I-SEMANTICS 2013
Posters & Demonstrations Track

i-semantics

9th International Conference on Semantic Systems
September 4-6, 2013, Graz, Austria

Title: Proceedings of the I-SEMANTICS 2013 Posters & Demonstrations Track
Editor: Steffen Lohmann
ISSN: 1613-0073
CEUR Workshop Proceedings
(CEUR-WS.org)

Editor's address:

Steffen Lohmann

University of Stuttgart
Universitätsstraße 38
70569 Stuttgart, Germany
steffen.lohmann@vis.uni-stuttgart.de

Preface

For the ninth time, *I-SEMANTICS* takes place in Graz, Austria, this year. The *International Conference on Semantic Systems* provides a forum for the exchange of latest scientific results in research areas such as semantic technologies, social software, semantic content engineering, logic programming, Linked Data, and the Semantic Web. The conference has become a major event in the field, attracting more than 400 participants every year.

The I-SEMANTICS Posters & Demonstrations Track complements the main conference track. It provides an opportunity to present late-breaking research results, smaller contributions, and innovative work in progress. It gives conference attendees the possibility to learn about on-going work and encourages discussions between researchers and practitioners in the field. For presenters, it provides an excellent opportunity to obtain feedback from peers.

Each submission to the Posters & Demonstrations Track was sent for review to three members of the program committee. Based on the reviews, we selected twelve posters and demos for presentation at the conference. The papers to these contributions are included in the proceedings.

We thank all authors for their contributions and the members of the program committee for their valuable work in reviewing the submissions. We are also grateful to the staff of the Know-Center, in particular Patrick Höfler who supported us in the local organization of this track.

August 2013

Steffen Lohmann
(Posters & Demos Chair)

Program Committee

Panos Alexopoulos, iSOCO, Spain

Christian Blaschke, Semantic Web Company, Austria

Kalina Bontcheva, University of Sheffield, United Kingdom

Adrian Brasoveanu, MODUL University Vienna, Austria

Simone Braun, FZI Research Center for Information Technologies, Germany

Irene Celino, CEFRIEL, Italy

Alicia Cortés Fernández, Instituto Tecnológico de Apizaco, Mexico

Aba-Sah Dadzie, University of Sheffield, United Kingdom

Claudia D'Amato, University of Bari, Italy

Thomas Gottron, University of Koblenz-Landau, Germany

Siegfried Handschuh, Digital Enterprise Research Institute (DERI), Ireland

Laura Hollink, VU University Amsterdam, The Netherlands

Tim Hussein, University of Duisburg-Essen, Germany

Christoph Lange, University of Birmingham, United Kingdom

Jindrich Mynarz, University of Economics, Prague, Czech Republic

Stefan Negru, Alexandru Ioan Cuza University, Romania

Alexandre Passant, seevl.net / MDG Web Ltd., Ireland

Heiko Paulheim, TU Darmstadt, Germany

Thomas Riechert, University of Leipzig, Germany

Nadine Steinmetz, HPI Potsdam, Germany

Andreas Thalhammer, STI Innsbruck, Austria

Contents

Diversity-Aware Clustering of SIOC Posts

Andreas Thalhammer, Ioannis Stavrakantonakis, and Ioan Toma

University of Innsbruck, Technikerstr. 21a, A-6020 Innsbruck
{andreas.thalhammer, ioannis.stavrakantonakis, ioan.toma}@sti2.at

Abstract. Sentiment analysis as well as topic extraction and named entity recognition are emerging methods used in the field of Web Mining. Next to SQL-like querying and according visualization, new ways of organization have become possible. In this demo paper we apply efficient clustering algorithms that stem from the image retrieval field to `sioc:Post` entities, blending similarity scores of sentiment and covered topics. We demonstrate the system with a visualization component that combines different diversity aspects within microposts by Twitter users and a static news article collection.

1 Introduction

Named entity recognition, automatic tagging, and sentiment detection in microposts, news articles, blog posts, forum posts etc. provide us new ways of interacting with content. Not only is it possible to retrieve answers from queries like *"select all positive articles that mention Barack Obama"* but these features offer a new way of content organization: combining sentiment and topic similarity in a single clustering approach. This enables the user to browse datasets in a novel way, for example getting overviews on positive and negative opinions on the topic *"champions league final"* or retrieving different topic clusters in negative Tweets from a specific user.

In this work, we demonstrate the application of two efficient clustering algorithms that stem from the image retrieval domain to sentiment analysis in combination with topic extraction and named entity recognition. We apply our approach on two use cases: microposts and news articles. Moreover, the readers are invited to try the system with live Twitter data to find new insights about the polarity and topic distribution of politicians' Tweets as well as their own.

2 Related Work

The contribution of our work is twofold, from a cluster dimension perspective (i.e., sentiment and topics are covered) as well as from a domain perspective (i.e., news articles and Tweets are covered). In this short paper we are not able to provide an extensive overview of the state of the art but we would like to contextualize our approach along with two related approaches.

[3] presents a study on automatically clustering and classifying Tweets. The outcomes of the paper stress out that employing a supervised methodology based

on hash-tags could produce better results than the traditional unsupervised methods. Furthermore, the authors present a methodology for finding the most representative Tweet in a cluster. Automatic detection of topics discussed in Tweets is pointed out as one of the interesting problems in Tweet analysis.

[2] proposes an emotion-oriented clustering approach in accordance to sentiment similarities between blog search result titles and snippets. The authors propose an approach for grouping blog search results in sentiment clusters, which is related to the grouping that we perform in the retrieved articles when we choose to cluster them based on the sentiment rather than the topic. The authors' goals are similar to ours as the approach focuses on very short text portions, which is also covered by our method as we cluster Tweets which are no longer than 140 characters. The sentiment detection relies on the SentiWordNet[1] which is built on top of WordNet and it provides sentiment scores on the glosses of WordNet.

In comparison to [3] and [2] which focus on clustering either by topics or sentiments, our approach combines those elements in a flexible way. For this, we introduce a straight-forward combination of topic and sentiment similarity measures that can be flexibly adapted to be more specific towards either topic or sentiment. Similarly to [2] we try to cover clusters of microposts as well as longer articles.

3 Data Extraction, Modeling, and Storage

We utilize the Twitter API to access the microposts and a static news corpus of the RENDER project[2]. The extracted Twitter data is processed using the Enrycher service[3] and stored in a Sesame[4] or OWLIM[5] triple store. The news data is already processed with Enrycher and already available in the correct format in an OWLIM triple store. As a data model we are utilizing the sioc [1] vocabulary in combination with the Knowledge Diversity Ontology[6] (KDO) [4]. KDO was developed in the context of the RENDER project and features assigning sentiments to sioc posts. Moreover we make use of the newly introduced type `kdo:NewsArticle` and the class `sioc-types:MicroblogPost`, both being subclasses of `sioc:Post`. In accordance to the respective document, the Enrycher service [6] assigns to instances of these subclasses a range of `sioc:topics` as well as a sentiment (i.e., `kdo:hasSentiment`). The data model as well as instances are stored in and retrieved from a triple store implementing the SAIL[7] interface (e.g. OWLIM).

[1] SentiWordNet – `http://sentiwordnet.isti.cnr.it/`

[2] RENDER News Corpus – `http://rendernews.ontotext.com/`, RENDER project – `http://render-project.eu`

[3] Enrycher – `http://enrycher.ijs.si`, `http://ailab.ijs.si/tools/enrycher/`

[4] Sesame - `http://www.openrdf.org/`

[5] OWLIM – `http://owlim.ontotext.com/`

[6] KDO – `http://kdo.render-project.eu/`

[7] SAIL API – `http://www.openrdf.org/doc/sesame2/system/ch05.html`

4 Diversity-Aware Clustering

Van Leuken et al. introduce “visual diversification of image search results” in [5]. The involved clustering algorithms are reported to be effective and efficient. The introduced similarity measures are based on visual similarity of images. For our document-based approach, we employ a combination of two similarity measures, namely topic and sentiment similarity. The final score is calculated with a flexible weighting component γ (with $0 \leq \gamma \leq 1$). We calculate the similarity of two `sioc:Posts` p_1 and p_2 as follows:

$$sim(p_1, p_2) = \gamma \cdot jacc(p_1, p_2) + (1 - \gamma) \cdot sent(p_1, p_2) \tag{1}$$

In formula 1 the functions *jacc* and *sent* need yet to be defined. *jacc* is basically a simple Jaccard similarity index over topics:

$$jacc(p_1, p_2) = \frac{|topics(p_1) \cap topics(p_2)|}{|topics(p_1) \cup topics(p_2)|} \tag{2}$$

We assume the extracted sentiment scores to be in the interval of $[0, 1]$ with 1 being most positive and 0 being most negative. The similarity score *sent* takes this into account, having the highest similarity of 1 if the two scores are equal. This similarity score is calculated as follows:

$$sent(p_1, p_2) = 1 - |score(p_1) - score(p_2)| \tag{3}$$

For the case that the scores are not in the mentioned interval, they are normalized as follows:

$$score(p) = \frac{score(p) - min(score(p))}{max(score(p)) - min(score(p))} \tag{4}$$

We utilize the Folding and Maximum algorithm from [5]. These algorithms were originally designed to cluster in accordance to visual similarity of images. Rather than using image histograms, we apply these algorithms to textual features of posts, using the similarity measure from above (see Formula 1).

The Folding algorithm assumes a ranked list as input. There are two disjoint lists maintained, the representatives and the rest. At the start, the ranked input is the rest. The algorithm selects the first element of the rest (i.e., the ranked input list) as a representative. In the following, each element of the rest is compared to the representatives and added to the representatives list in case its similarity to all existing representatives is less than a certain reference point (i.e., a variable ϵ). When all representatives are established, each element in the rest is assigned to the cluster of which the representative is most similar to it.

The Maximum algorithm is similar to Folding but has some distinct features. The Maximum algorithm belongs to the class of randomized algorithms. Again there are two disjoint lists, the representatives and the rest which is assigned to the input at the beginning. The first element of the representatives is selected randomly from the rest. Then, the algorithm adds the element which

```
Data: List L containing sioc posts
Result: double value of □
sumAll := 0;
for each sioc:Post s1 in L do
    sum := 0;
    for each sioc:Post s2 in L do
        if s1 != s2 then
            Sum := Sum + sim(s1, s2);
    Avg := Sum / (size(L) -1);
    SumAll := SumAll + Avg;
return SumAll / size(L);
```

Algorithm 1: ϵ estimation

has minimum maximum similarity (or maximum minimum distance) to the representatives. If this minimum maximum similarity is at some point less than ϵ, all representatives are found and the remaining elements in the rest list are assigned to the clusters with closest representatives.

Both algorithms produce clusters, each with a selected representative. However, as a last point, it remains open how to select an appropriate value for ϵ. In this step we determine the average similarity of a `sioc:Post` to another (see Algorithm 1).

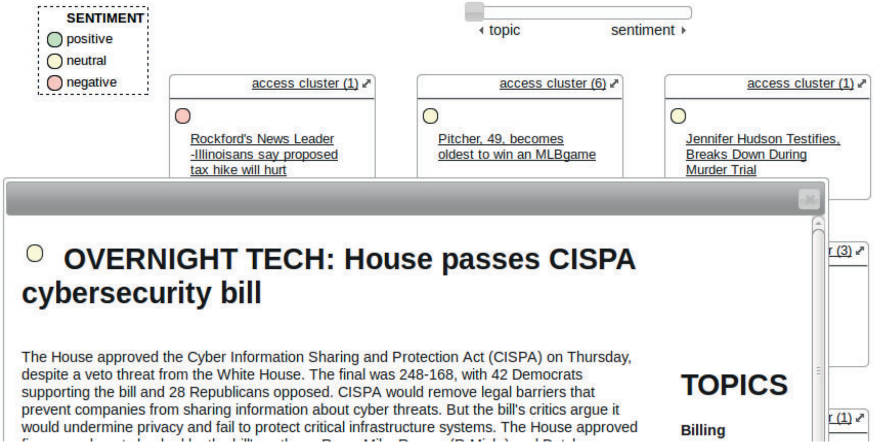
5 Implementation

We implemented the diversity-aware ranking service with Oracle GlassFish 3.x. The source code is available as a github project[8] and a deployment can be found at `http://ranking.render-project.eu/`. There, users can specify a variety of parameters and retrieve the JSON output for the clustering. For a better user experience, we introduce a jQuery-based visualization component that is demonstrated at `http://ranking.render-project.eu/tweetVis.html` (Twitter) and `http://ranking.render-project.eu/vis.html` (news). Figure 1 shows the news visualization component. The slider at the top changes the γ value of the similarity measure (see Formula 1) either towards sentiment similarity or topic similarity.

6 Conclusion

We have implemented a diversity-aware ranking service that enables clustering and retrieval of sioc posts along the two dimensions: sentiment and topic. We exemplify our approach on live Twitter data and a static news dataset. This work is also meant to initiate new directions to look at content organization, navigation, and presentation.

[8] Source code – `https://github.com/athalhammer/RENDER-ranking-service`



Rockford's News Leader -Illinoisans say proposed tax hike will hurt

Pitcher, 49, becomes oldest to win an MLBgame

Jennifer Hudson Testifies, Breaks Down During Murder Trial

OVERNIGHT TECH: House passes CISPA cybersecurity bill

The House approved the Cyber Information Sharing and Protection Act (CISPA) on Thursday, despite a veto threat from the White House. The final was 248-168, with 42 Democrats supporting the bill and 28 Republicans opposed. CISPA would remove legal barriers that prevent companies from sharing information about cyber threats. But the bill's critics argue it would undermine privacy and fail to protect critical infrastructure systems. The House approved

Fig. 1. The news visualization component.

Acknowledgements We are grateful for the feedback from Daniele Pighin (Google Zurich). This research was partly funded by the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 257790 (RENDER project).

References

1. John G. Breslin, Andreas Harth, Uldis Bojars, and Stefan Decker. Towards semantically-interlinked online communities. In *The Semantic Web: Research and Applications*, volume 3532 of *Lecture Notes in Computer Science*, pages 500–514. Springer Berlin Heidelberg, 2005.
2. Shi Feng, Daling Wang, Ge Yu, Chao Yang, and Nan Yang. Sentiment clustering: A novel method to explore in the blogosphere. In *Proceedings of the Joint International Conferences on Advances in Data and Web Management*, APWeb/WAIM '09, pages 332–344, Berlin, Heidelberg, 2009. Springer-Verlag.
3. K. D. Rosa, R. Shah, B. Lin, A. Gershman, and R. Frederking. Topical clustering of tweets. *Proceedings of the ACM SIGIR: SWSM*, 2011.
4. Andreas Thalhammer, Ioan Toma, Rakebul Hasan, Elena Simperl, and Denny Vrandečić. How to represent knowledge diversity. Poster at 10th intl. Semantic Web Conf. (ISWC11), 10 2011.
5. Reinier H. van Leuken, Lluis Garcia, Ximena Olivares, and Roelof van Zwol. Visual diversification of image search results. In *Proc. of the 18th intl. conf. on World Wide Web*, WWW '09, pages 341–350, New York, NY, USA, 2009. ACM.
6. Tadej Štajner, Delia Rusu, Lorand Dali, Balž Fortuna, Dunja Mladenić, and Marko Grobelnik. Enrycher: service oriented text enrichment. In *Proc. of the 11th intl. multiconference Information Society*, IS-2009, 2009.

Mobile Location-Driven Associative Search in DBpedia with Tag Clouds

Bjørnar Tessem, Bjarte Johansen, and Csaba Veres

Department of Information Science and Media Studies,
Postbox 7802, University of Bergen, 5020 Bergen, Norway
bjornar.tessem@uib.no bjarte.johansen@uib.no
csaba.veres@uib.no

Abstract. A primary contextual source for today's context-sensitive mobile phone apps is the user's location. The recent surge in the availability of open linked data can provide location-oriented semantic context, still wanting to be explored in innovative ways. In PediaCloud, the Android tool described here, we show how we can use the associative structure of the Semantic Web at a geographical location, visualize location information with tag clouds, and allow users to follow the associations of the Semantic Web enabled by the tag cloud, with the aim of enabling the users to construct an understanding of the "place" around them. The data we use are found through DBpedia, a project a project aimed to lift the information in WikiPedia into the Semantic Web.

Keywords: tag clouds, mobile, location, semantic web, DBpedia

1 Introduction

Exploiting location context has become a major theme for research on mobile technologies and is widely applied in commercial applications. These new technologies have implications for the realisation of the concept of *place*, which in sociology is understood as not only the location itself, but also the physical surroundings, and a persons attribution of meaning to the surroundings[4]. The location-based mobile tools guide the users to information about the surroundings, and enable new ways of constructing a user's understanding of place. Examples of such tools are Google maps and Google places, which in order to enhance the experience of *places*, connect to located information from among others Wikipedia, and its linked data extraction, DBpedia[1].

A potential technique for presenting located information are tag clouds, which are a means of visualizing natural language information. In a tag cloud a collection of words is drawn in a bounded area, each word with a font proportional to a weight computed from the text collection, often the count of that particular word. The size of the words gives the user an impression of what topics are important in the text collection.

With the PediaCloud tool we aim to show how one can create tag clouds of located information from DBpedia. The goal is to use not only located (primary) resources, but also to collect DBpedia entities that have some semantic link to the located resources

[1] http://dbpedia.org

(secondary resources), and build the tag clouds from this combined set of information. This should give users a richer sense of the semantic aspects of a location, the idea being that both primary and secondary resources contribute to a user's understanding of the place. It is also a goal to create a mobile semantic web application without a dedicated backend for the intermediate organisation of information, instead aiming to use DBpedia as an example of an existing semantic web resource that can be consulted directly.

2 The Workings of PediaCloud

The DBpedia data collected from Wikipedia articles contain features like location, abstract, categories, and other relations to many types of entities. For the PediaCloud application we use Wikipedia articles with a location, and their abstracts, and in addition the abstracts of the linked secondary resources. The collected abstracts are the source of the tag clouds we build. The construction of the tag cloud essentially goes through the following steps:

1. Get user location
2. Get all articles (primary) within fixed radius of location.
3. Get all articles (secondary) that the primary articles link to.
4. Find frequency of each word (wf) in each article's abstract.
5. Weight the wf of the words from the primary articles as a function of distance, scaling linearly with weight 1.0 at the user's location and 0.0 at the radius.
6. Weight the wf of the words from the secondary articles by the cosine similarity to the primary article it is linked from, multiplied by the weight of the primary article.
7. Add the weighted wf for each word together.
8. Select the highest n scoring words.
9. Use the word score to create tag cloud for current area.

Figure 1 illustrates how we weight the words in the tag cloud.

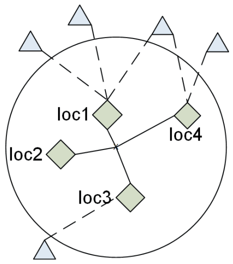
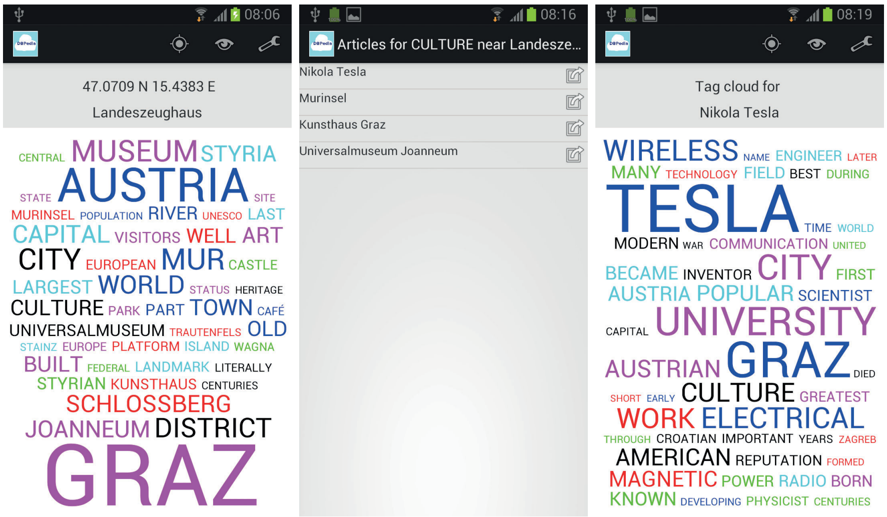


Fig. 1. The centre indicates the user's position, and the located articles are illustrated with diamonds at their locations loc1, loc2, ... Light gray triangles outside the circle indicate secondary resources. Located articles contributes according to their distance, secondary according to the distance of their located origin multiplied by the cosine similarity (indicated by dotted lines).



08:06

47.0709 N 15.4383 E

Landeszeughaus

CENTRAL MUSEUM STYRIA
STATE AUSTRIA SITE
MURINSEL POPULATION RIVER UNESCO LAST
CAPITAL VISITORS WELL ART
CITY EUROPEAN MUR CASTLE
LARGEST WORLD STATUS HERITAGE
CULTURE PARK PART TOWN CAFÉ
UNIVERSALMUSEUM TRAUTENFELS OLD
STAINZ EUROPE PLATFORM ISLAND WAGNA
BUILT FEDERAL LANDMARK LITERALLY
STYRIAN KUNSTHAUS CENTURIES
SCHLOSSBERG
JOANNEUM DISTRICT
GRAZ

08:16

Articles for CULTURE near Landesze...

Nikola Tesla

Murinsel

Kunsthaus Graz

Universalmuseum Joanneum

08:19

Tag cloud for

Nikola Tesla

WIRELESS NAME ENGINEER LATER
MANY TECHNOLOGY FIELD BEST DURING
TESLA TIME WORLD
MODERN WAR COMMUNICATION UNITED
BECAME INVENTOR CITY FIRST
AUSTRIA POPULAR SCIENTIST
CAPITAL UNIVERSITY
AUSTRIAN GRAZ DIED
SHORT EARLY CULTURE GREATEST
WORK ELECTRICAL
THROUGH CROATIAN IMPORTANT YEARS ZAGREB
AMERICAN REPUTATION FORMED
MAGNETIC POWER RADIO BORN
KNOWN DEVELOPING PHYSICIST CENTURIES

2a 2b 2c

Fig. 2. a: Tag cloud for a user located at the Graz Main Square. The closest located Wikipedia article is about Landeszeughaus. **b**: Ranked Wikipedia articles with the tag "CULTURE". **c**: Tag cloud for "Nikola Tesla".

The tag cloud for the Main square of Graz is shown in Fig. 2a. The user can select any word in the tag cloud, for example ”CULTURE”. This will show the list of WikiPedia articles that contain the word ”culture” (Fig. 2b). Now, the user may choose to look at the Wikipedia article for a resource, but may also select one of these articles as a focused resource (as an alternative to the user's location) in a new tag cloud. This is shown in Fig. 2c where "Nikola Tesla" was chosen as a focal point. Note that the selected article no longer has a spatial component. We use the same collection of articles as for the previous tag cloud. However, the weight of the different abstracts will only be based on the cosine similarity to the focused resource. The word weights in the tag cloud thus depends on the choice of focused resource, giving the user a sense of what words are most prominent in the collection given this special focus.

The effect of using cosine similarity combined with word frequency to weight the words as opposed to using only word frequency is shown in Table 1, where we show the ten most prominent words and their weights in two tag clouds generated for the "Nikola Tesla" resource. We notice that words relating to Nikola Tesla's achievements (ELECTRICAL, WIRELESS) get a place in the list when we use a cosine similarity approach as opposed to when we use word frequency only. The word frequency approach mainly results in an emphasis on general geographical words and nationalities. With the use of cosine similarity, larger fonts are given to words that are more informative.

The query we send to DBpedia's sparql endpoint[2] is a single SELECT with UNION gathering data for both primary and secondary resources at the same time (steps 2 and 3 in the process). From the returned data the tool is able to compute the tag cloud for

[2] http://DBpedia.org/sparql

Nikola Tesla: Word frequency		Nikola Tesla: Word freq. * Cos.sim.	
GRAZ	57.0	TESLA	9.00
UNIVERSITY	44.0	GRAZ	6.78
CITY	30.0	UNIVERSITY	5.42
AUSTRIA	24.0	CITY	4.83
AUSTRIAN	21.0	ELECTRICAL	4.00
FIRST	16.0	WIRELESS	4.00
KNOWN	13.0	WORK	3.81
CROATIAN	13.0	CULTURE	3.60
CAPITAL	12.0	AUSTRIAN	3.32
WORLD	12.0	AMERICAN	3.23

Table 1. Weights and ranking for the 10 highest ranked words in a tag cloud for the same resource ("Nikola Tesla"), but with different weighting approaches.

the mobile screen. A positive effect of doing one single query is that it saves response time due to less network connection time.

We would have preferred to run the data gathering as a single CONSTRUCT in order to store all relevant triples locally, possibly on a triple store, but we ran in to a memory limit at the DBpedia endpoint when sending the query, so we had to go for the SELECT version. We considered installing a triple store with a query engine on the device to support the CONSTRUCT version, but discovered early that even though many of the triple stores and query engines are written in Java they are currently not working on the Android platform.

We are getting interesting results with the current implementation, but there are weaknesses that we would like to fix. One problem is that some words describing large enclosing areas (like ”GRAZ” and ”AUSTRIA”) often get very high weights, but may not be very informative in the sense of getting a cultural and historical overview of a location. We believe that we can reduce the weight of these words by modifying the weighting algorithm we use, for instance by using tf-idf in conjunction with the already implemented cosine similarity.

3 Related Work and Conclusion

DBpedia has been used as a source of information in DBpedia mobile which is a tool presenting DBpedia resources close to the user at a map [2]. Ruta et al. [7] also use DBpedia data, and combine semantic similarity with location closeness to give DBpedia sources an overall match to a search criteria. MapXplore [9] is a tool that uses DBpedia to present classification and other factual data about points of interest, to users. MapXplore uses a category browser for locating relevant points of interest for users, in order to give them an overall impression of the important concepts at a place. For example, Bergen in Norway features prominently with the concept ”mountain”, whereas Dubai features with the category ”skyscraper”. van Aart et al. [8] use data from a variety of sources and connect non-located resources to a location through links to a located resource. Mäkelä et al. [5] do the same and put these data into a backend store and make them accessible through a mobile application. Paelke et al. [6] and Baldauf et al. [1] both use tag clouds on mobiles to present information from resources that are tagged with location data. Dörk et al. [3] also use tag clouds in a web application allowing location-based exploratory web searches with visualization tools.

PediaCloud integrates ideas from these related projects as it focuses on located information from DBpedia, and further, its visualisation through tag clouds. A particular feature of PediaCloud is the use of secondary resources in constructing the tag cloud, and that the tag cloud changes depending on the user's choice of focus, either the user's location or a particular DBpedia resource. The weighting of tag cloud words are computed from word counts combined with cosine similarity and geographical distance, resulting in a higher emphasis on the more informative tags. PediaCloud also does not depend on a dedicated backend. The tool gathers information from a main Semantic Web endpoint, and computes the visual presentation locally.

References

1. Baldauf, M., Fröhlich, P., Reichl, P.: The ambient tag cloud: A new concept for topic-driven mobile urban exploration. In: Proceedings of the European Conference on Ambient Intelligence. AmI '09, Berlin, Heidelberg, Springer-Verlag (2009) 44–48
2. Becker, C., Bizer, C.: Exploring the geospatial semantic web with DBpedia Mobile. Web Semantics: Science, Services and Agents on the World Wide Web **7**(4) (2012)
3. Dörk, M., Williamson, C., Carpendale, S.: Navigating tomorrow's web: From searching and browsing to visual exploration. ACM Trans. Web **6**(3) (October 2012) 13:1–13:28
4. Gieryn, T.F.: A space for place in sociology. Annual Review of Sociology **26**(1) (2000) 463–496
5. Mäkelä, E., Lindblad, A., Väätäinen, J., Alatalo, R., Suominen, O., Hyvönen, E.: Discovering places of interest through direct and indirect associations in heterogeneous sources — the travelsampo system. In: Terra Cognita 2011: Foundations, Technologies and Applications of the Geospatial Web, CEUR Workshop Proceedings, Vol-798 (2011)
6. Paelke, V., Dahinden, T., Eggert, D., Mondzech, J.: Location based context awareness through tag-cloud visualization. In: Advances in Geo-Spatial Information Science. CRC Press (2012) 265–273
7. Ruta, M., Scioscia, F., Di Sciascio, E., Piscitelli, G.: Location-based semantic matchmaking in ubiquitous computing. In: Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 03. WI-IAT '10, Washington, DC, USA, IEEE Computer Society (2010) 124–127
8. van Aart, C., Wielinga, B., van Hage, W.R.: Mobile cultural heritage guide: location-aware semantic search. In: Proceedings of the 17th international conference on Knowledge engineering and management by the masses. EKAW'10, Berlin, Heidelberg, Springer-Verlag (2010) 257–271
9. Veres, C.: MapXplore: Linked data in the app store. In Sequeda, J., Harth, A., Hartig, O., eds.: COLD. Volume 905 of CEUR Workshop Proceedings., CEUR-WS.org (2012)

Schema.org for the Semantic Web with MaDaME

Csaba Veres, Eivind Elseth

Department of Information Science and Media Studies,
Postbox 7802, University of Bergen, 5020 Bergen, Norway
csaba.veres@infomedia.uib.no, eivind.elseth@student.uib.no

Abstract. Schema.org is a high profile initiative to introduce structured markup into web sites. However, the markup is designed for use cases relevant to search engines, which limits their general usefulness. MaDaME is a tool to help web developers to annotate their web pages with schema.org annotations, but in addition automatically injects semantic metadata from SUMO and WordNet. It is unlike previous tools in that it assumes no knowledge of the metadata standards. Instead, users provide disambiguated natural language terms, and the tool automatically picks the most appropriate metadata terms from the different vocabularies.

Keywords: schema.org, wordnet, semantic web, markup, search

1 Introduction

Schema.org was launched on June 2, 2011, under the auspices of a powerful consortium consisting of Google, Bing, and Yahoo! (they were subsequently joined by Yandex). They established the http://schema.org web site whose main purpose is to document an extensive type schema which is meant to be used by web masters to add structured metadata to their content. In a sense schema.org provides extended semantics for rich snippets1 with the motivation that markup can be used to display more information about web sites on the search page which may result in more clicks, and perhaps higher rankings in the long run.

The schema was designed specifically for the use cases developed by the search engines, and both the semantics and preferred syntax reflect that choice. In terms of semantics, the schema has some non-traditional concepts to fulfill its role. For example there is a general class of *Product* but no general class for *Artifact*. There are also odd property ascriptions from the taxonomy structure, so, for example, Beach has *openingHours* and *faxNumber*. These oddities exist because, we are told, they reflect what people are predominantly looking for when they perform a search. In terms of syntax, there is a very strong message that developers should use the relatively new Microdata format, designed specifically for the schema, rather than the vastly more popular RDFa web standard [1]. The choice is dictated by simplicity, because Microdata has just those elements required for the schema. But this choice is unfortunate

[1] http://goo.gl/RAJy8

because it makes metadata from schema.org incompatible with many other sources of metadata like Facebook's OGP.[2] [2] lists five key reasons why RDFa Lite 1.1 should be the preferred syntax over Microdata. RDFa is feature equivalent to Microdata, and it is supported by all major search crawlers including Facebook, while Microdata is not. For the purposes of expressing schema.org, RDFa is no more complex than Microdata. But most importantly from the perspective of general semantic markup, RDFa is designed to naturally mix vocabularies while Microdata makes it much more difficult to do so. Thus if annotating web pages with multiple vocabularies is the desired goal, then RDFa Lite 1.1 is the best choice.

MaDaME (Meta Data Made Easy) is a markup tool developed for two specific purposes. First, it must to help web developers who were not familiar with the schema.org to mark up their web sites as easily as possible. This is important because the idiosyncratic nature of the schema can make concepts hard to navigate. It is especially important if a web developer wants to mark up a site for which there is no existing type in schema.org. For example a web master might be designing a web site about caves for tourists to visit, but schema.org does not have a type for *cave*. We wanted to help developers find the best markup in these cases, without requiring them to study the schema itself. The second important motivator was to make the markup episode as fruitful as possible, since it is not easy to motivate people to provide structured data about their web site. This means the markup should be useful in as many use cases as possible. We achieve this by producing RDFa markup and mixing different vocabularies to describe the same object. While there are existing efforts to provide tool support for schema.org markup, including a tool from Google,[3] all of them require some knowledge of the schema, and none of them provide rich markup for a more general semantic web.

2 MaDaME

MaDaME has at its core a mapping file between WordNet word senses and schema.org types. WordNet can for our purposes be regarded as a comprehensive electronic dictionary which defines word senses through numerous relationships to other words [3]. Web developers simply look up the word which expresses the content of their site, and they are given the best matching schema.org markup. Obviously not all words will have direct mappings to schema.org, so we also have an algorithm to infer the best match for those.

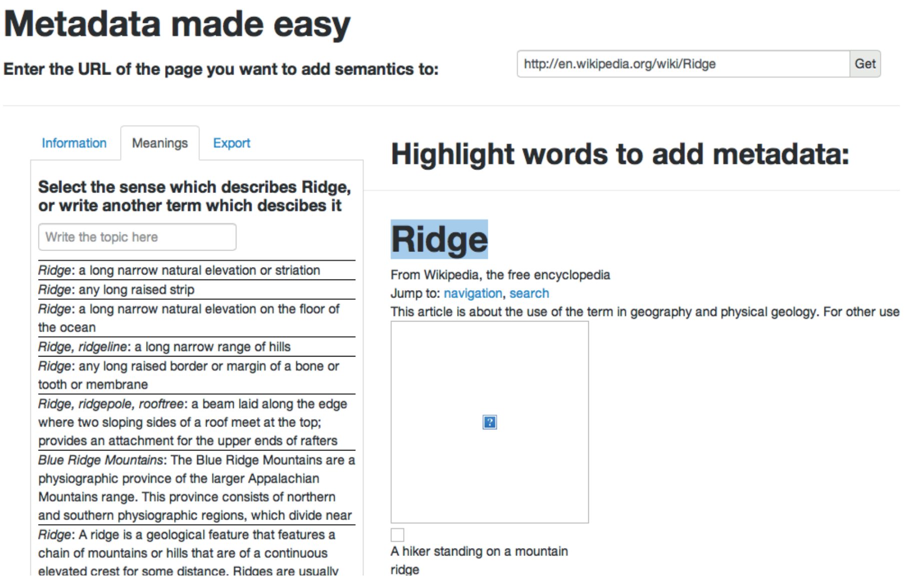
To import a page into the web app the user will write the URL of the web site he wants to mark up into the URL input field. The page will then be loaded into the web app after some preprocessing. The preprocessing consists of commenting out scripts and iframes which might not run correctly. The user then selects words, phrases, or images to tag by highlighting them on the page. When a word item has been highlighted, its possible senses in WordNet are retrieved. The user picks one of these senses by clicking on it. In fig. 1 we can see the word *ridge* highlighted, and the corresponding disambiguation options. The sense the user picks is sent back to the server for map-

[2] http://ogp.me

[3] http://goo.gl/7DGr5D

ping to schema.org, as well as a selection of other ontologies. So far we have only implemented SUMO [4] and WordNet itself. The Schema.org mappings can be further refined by filling out the properties defined by the schema, using a popup form.

When the users finish marking up the document they are given a link to a newly created webpage containing their original page plus the meta data they have created. In most cases where the web site is simple HTML there will be no need to manually modify any code. From here they can save the document and upload it to their own server.



Metadata made easy

Enter the URL of the page you want to add semantics to: http://en.wikipedia.org/wiki/Ridge

Select the sense which describes Ridge, or write another term which descibes it

Write the topic here

Ridge: a long narrow natural elevation or striation

Ridge: any long raised strip

Ridge: a long narrow natural elevation on the floor of the ocean

Ridge, ridgeline: a long narrow range of hills

Ridge: any long raised border or margin of a bone or tooth or membrane

Ridge, ridgepole, rooftree: a beam laid along the edge where two sloping sides of a roof meet at the top; provides an attachment for the upper ends of rafters

Blue Ridge Mountains: The Blue Ridge Mountains are a physiographic province of the larger Appalachian Mountains range. This province consists of northern and southern physiographic regions, which divide near

Ridge: A ridge is a geological feature that features a chain of mountains or hills that are of a continuous elevated crest for some distance. Ridges are usually

Highlight words to add metadata:

Ridge

From Wikipedia, the free encyclopedia

Jump to: navigation, search

This article is about the use of the term in geography and physical geology. For other use

A hiker standing on a mountain ridge

Fig. 1. A screenshot of MaDaME with options for *ridge* on the left of the screen

All of the markup is in the RDFa Lite 1.1 syntax, which is the current W3C recommendation,[4] and has the necessary features to handle multiple namespaces and multiple types elegantly.

The algorithm for finding markup for the selected senses is in two stages. The first stage is to build an extended tree of WordNet senses. This is done by using a perl library (the WordNet::QueryData library from CPAN) which is capable of querying the WordNet database. We have written a script that when given a WordNet sense will find the hypernyms of the sense (more general senses), and all the hyponyms (more specific) of these ancestor nodes. We call this the *mapping tree*, which intuitively contains all the words in the semantic neighbourhood of the original word.

In the second stage we find mappings for the user selected synsets. If a direct mapping to schema.org exists then this is simply added to the markup. For novel words we

[4] http://www.w3.org/TR/rdfa-lite/

use mappings for the closest available related sense from the *mapping tree* which does have a direct mapping. We tried several versions of the mapping algorithm, and the most successful one turned out to be a simple depth-first traversal of the *mapping tree* until a sense is found with a direct mapping to the schema. For a simple example, consider the concept *ridge* which is not represented in schema.org. The correct sense of wn:ridge has the hypernym wn:geological_formation, which has a direct mapping to schema:Landform. Therefore *ridge* is marked up as schema:Landform. SUMO has direct mappings for a very large number of WordNet senses and *ridge* has a corresponding mapping in SUMO as sumo:UplandArea, so the concept *ridge* would acquire mappings schema:Landform as well as sumo:UplandArea. More generally, any vocabulary that is mapped to WordNet could be used to provide metadata. In future releases we plan to provide facilities for advanced users to incorporate their own mapping files to an ontology of their choice.[5]

3 Results

We performed an automatic evaluation of 4350 random nouns in WordNet to see how they mapped to schema types, by measuring the average depth of the mapped type in the schema.org taxonomy. The result was a somewhat disappointing 0.689, which means that most words were mapped to schema:Thing or one of its immediate specialisations.

To test how this compares to real world usage we sampled a set of five web sites that had used schema.org markup. We ended up with a restaurant review from the Telegraph, a tour operators customer feedback page, a tourist agency home page, the home page of a marketing company and a movie review sites review of a film. When we manually added markup by selecting key words in the text we achieved 100% agreement. While this is clearly a small study, it does suggest that the schema.org markup we will see “in the wild” will represent concepts from the top nodes of the type hierarchy. The relatively shallow mappings may be a reflection of the schema itself, rather than a criticism of our mapping algorithm.

4 Related Work

There are existing approaches for annotating web pages with semantic markup, especially schema.org. These can broadly be categorised as manual or automatic annotation tools.

The schema.rdfs.org web site links to a number of publishing tools[6]. The two major form-based tools, Schema Creator and Microdata Generator, both provide a forms based interface for entering detailed properties, not unlike the MaDaME interface. However in these tools the web author must find the appropriate schema types by

[5] The tool can be tried at http://csaba.dyndns.ws:3000.

[6] http://schema.rdfs.org/tools.html

browsing a sub set of the most common types that are presented in these tools. They both differ from our approach because they expect the author to make decisions about which schema types to use. Similarly, major content management platforms like Drupal, Joomla!, WordPress and Virtuoso provide mechanisms for adding schema.org types to their content.

Amongst automatic annotation tools, [5] presents a tool that can add schema.org types automatically, but only to web pages about patents. Their approach uses underlying domain knowledge to extract key terms and a patent knowledge base to generate structured microdata markup for web pages. It remains to be seen if this approach could scale to web sites in general.

5 Conclusion

Schema.org is a promising initiative from the search engines in that it exposes structured metadata to a vast new audience of web developers. However, this requires some learning of the syntax and vocabulary of the particular markup, which could limit the breadth of metadata that will appear from web developers. MaDaME is a tool that helps web masters use the schema because it removes the requirement to learn a new vocabulary and syntax, while providing the necessary markup. The markup can be extended to other proprietary standards like Facebook's OGP, so web sites could be annotated with both standards at no extra effort. But we see MaDaME's most important contribution as one to the semantic web effort because it piggybacks on the major search engine backed initiative, to include markup from popular ontologies that can be used for diverse semantic applications.

References

1. P. Mika and T. Potter, "Metadata Statistics for a Large Web Corpus," WWW2012 Workshop on Linked Data on the Web (LDOW '12), Lyon, France, 16-Apr-2012. Online Available: http://ceur-ws.org/Vol-937/ldow2012-inv-paper-1.pdf [Accessed: 11-Jul-2012].
2. M. Sporny, "Mythical Differences: RDFa Lite vs. Microdata | The Beautiful, Tormented Machine," manu.sporny.org. Online Available: http://manu.sporny.org/2012/mythical-differences. [Accessed: 17-Jul-2013].
3. G. A. Miller, "WordNet: a lexical database for English," Communications of the ACM, vol. 38, no. 11, Nov. 1995, pp. 39–41
4. I. Niles and A. Pease, "Towards a Standard Upper Ontology," Proceedings of the International Conference on Formal Ontology in Information Systems (FOIS '01), ACM, 2001, pp. 2–9.
5. A. Norbaitiah and D. Lukose, "Enriching Webpages with Semantic Information," Proc. Int'l Conf. on Dublin Core and Metadata Applications 2012, Sep. 2012, pp. 1–11

Knowledge Tagger: Customizable Semantic Entity Resolution using Ontological Evidence

Panos Alexopoulos, Boris Villazon-Terrazas, and José-Manuel Gómez-Pérez

iSOCO, Avda del Partenon 16-18, 28042, Madrid, Spain,
{palexopoulos,bvillazon,jmgomez}@isoco.com

Abstract. Knowledge Tagger performs Named Entity Resolution (NER) in texts using relevant domain ontologies and semantic data as background knowledge. Its distinguishing characteristic is its disambiguation-related customization capabilities as it allows users to define and apply custom disambiguation evidence models, based on their knowledge about the domain(s) and expected content of the texts to be analyzed. In this demo we explain the structure and content of such evidence models and we demonstrate how, given a concrete resolution scenario, one may use our system to define and apply them to texts pertaining to this scenario.

1 Introduction

In this paper we demonstrate Knowledge Tagger[1], a system that utilizes background semantic information, typically in the form of Linked Data, to accurately determine the intended meaning of detected semantic entity references within texts. The system is based on a novel corresponding framework [1] that we have developed and which is particularly applicable to constrained scenarios where knowledge about what entities and relations are expected to be present in the texts to be analyzed is available.

More specifically, through a structured semi-automatic process the framework enables i) the exploitation of this a priori knowledge for the selection of the subset of domain semantic information that is optimal for the disambiguation scenario at hand, ii) the use of this subset for the generation of corresponding evidence and iii) the use of this evidence for the disambiguation of entities within the scenario's texts. As we have already shown in [1] this process allows our system to adapt to the particular characteristics of different domains and scenarios and be more effective than other similar systems primarily designed to work in open domain and unconstrained scenarios like, for example, DBPedia Spotlight [3], AIDA [2] or the systems included in NERD [4].

2 Framework and System Overview

Knowledge Tagger's underlying framework is based on the intuition that a given ontological entity is more likely to represent the meaning of an ambiguous term when there are many ontologically related to it entities in the text. The latter can be seen as

[1] http://glocal.isoco.net/disambiguator/demo

evidence whose quantitative and qualitative characteristics can be used to determine the most probable meaning of the term. Nevertheless, which entities and to what extent should serve as evidence in a given scenario depends on the domain and expected content of the texts that are to be analyzed. For that, the key ability our system provides to its users is to construct and use, in a semi-automatic manner, custom ontology-based disambiguation evidence models.

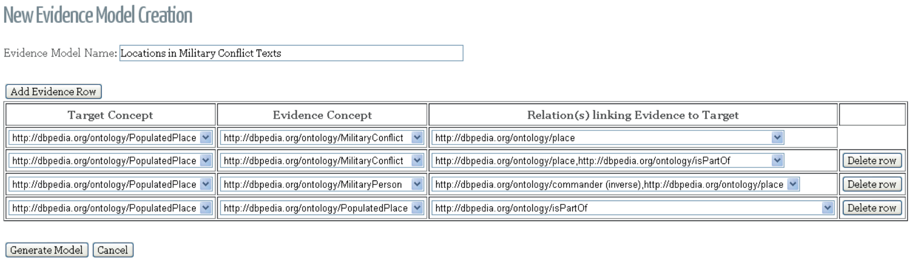
Such models define for given ontology entities which other entities and to what extent should be used as evidence towards their correct meaning interpretation (see Table 1). Their construction depends on the characteristics of the domain and the texts. For example, assume we want to disambiguate location references within textual descriptions of military conflicts like the following: *"Siege of Tripolitsa occured near Tripoli with Theodoros Kolokotronis being the leader of the Greeks against Turkey"*. The nature of these texts allows us to expect to find in them, among others, military conflicts, locations where these conflicts took place and people and groups that participated in them. This in turn allows us to use these entities as evidence for disambiguating one another. For example, in the above text the term "Tripoli" is mentioned along with terms like "Siege of Tripolitsa" (a battle that took place in Tripoli, Greece) and "Theodoros Kolokotronis" (the commander of the Greeks in this siege). Thus, it is fair to assume that this term refers to the Greek town of Tripoli rather than, for example, to Tripoli of Libya. Generalizing this, we may define the location disambiguation evidence model of Table 2 where, for instance, a populated place can be disambiguated by the military conflicts that took place in it (row 1) and by the military persons that fought in conflicts that took place in it (row 3).

Table 1. Examples of Target-Evidential Entity Pairs for the Miltary Conflict Scenario

Location	**Evidential Entity**	**dem**
dbpedia:Columbus,_Georgia	James H. Wilson	1.0
dbpedia:Columbus,_New Mexico	dbpedia:Pancho_Villa	1.0
dbpedia:Beaufort_County,_South_Carolina	dbpedia:James_Montgomery_(colonel)	0.25
dbpedia:Beaufort_County,_North_Carolina	dbpedia:John_G._Foster	1.0

Table 2. Sample Disambiguation Evidence Model for Military Conflict Texts

Target Concept	**Evidence Concept**	**Relation(s) linking Evidence to Target**
dbpedia-owl:PopulatedPlace	dbpedia-owl:MilitaryConflict	dbpprop:place
dbpedia-owl:PopulatedPlace	dbpedia-owl:MilitaryConflict	dbpprop:place, dbpedia-owl:isPartOf
dbpedia-owl:PopulatedPlace	dbpedia-owl:MilitaryPerson	is dbpprop:commander of, dbpprop:place
dbpedia-owl:PopulatedPlace	dbpedia-owl:PopulatedPlace	dbpedia-owl:isPartOf



New Evidence Model Creation

Evidence Model Name: Locations in Military Conflict Texts

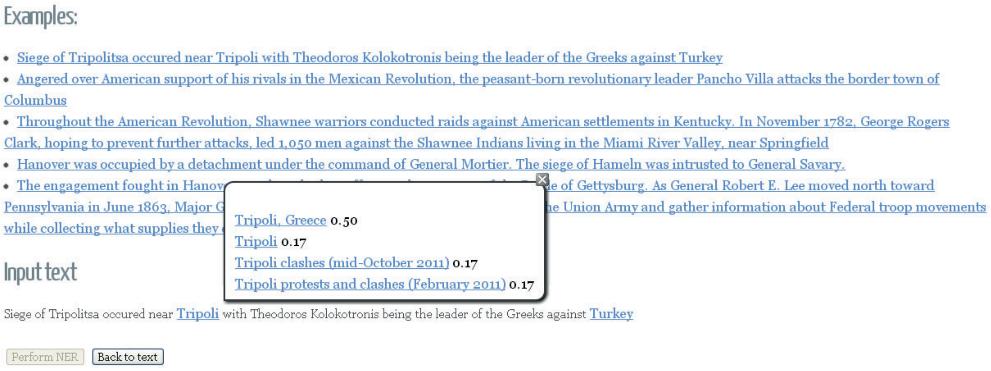
Target Concept	Evidence Concept	Relation(s) linking Evidence to Target	
http://dbpedia.org/ontology/PopulatedPlace	http://dbpedia.org/ontology/MilitaryConflict	http://dbpedia.org/ontology/place	
http://dbpedia.org/ontology/PopulatedPlace	http://dbpedia.org/ontology/MilitaryConflict	http://dbpedia.org/ontology/place,http://dbpedia.org/ontology/isPartOf	Delete row
http://dbpedia.org/ontology/PopulatedPlace	http://dbpedia.org/ontology/MilitaryPerson	http://dbpedia.org/ontology/commander (inverse),http://dbpedia.org/ontology/place	Delete row
http://dbpedia.org/ontology/PopulatedPlace	http://dbpedia.org/ontology/PopulatedPlace	http://dbpedia.org/ontology/isPartOf	Delete row

Fig. 1. New Evidence Model Creation Form

To define this model in the Knowledge Tagger demo we work as follows. First we press the **"Create New Evidence Model"** button to reveal the model creation form. Then we give a name for the new model (e.g. "Locations in Military Conflict Texts") and we start filling the table form with the information of Table 2 (see Figure 1). First we select the target concept (e.g. "PopulatedPlace"), then the one to be used as evidence (e.g. "MilitaryConflict") and then the (automatically calculated) relation path between them that we want to consider. For simplicity, in this demo we consider paths of maximum length two.

When the model is complete we press the **"Generate Model"** button to store the model into the server and generate target-evidence entity pairs. Each pair is accompanied by a degree that quantifies the evidential entity's strength for the given target. (see table 1). For example, James Montgomery acts as evidence for the disambiguation of Beaufort County, South Carolina because he's fought a battle there while his evidential power for that location is 0.25, practically because there are 3 other military persons in the ontology also named Montgomery. The exact way this strength is calculated may be found in [1]. In any case, depending on the size of the underlying ontology, the generation of the target-evidence pairs can take a while but it's a process that will need to be performed only once. For this example, the creation of the model takes about 30 seconds in a standard server environment.

When the generation process is finished, the new model appears as an option in the list of defined evidence models and can be used to perform entity detection and disambiguation. To do that we select the model and then use the "Input Text" form to perform NER to texts relevant to the scenario the model has been defined for. By pressing the **"Perform NER"** button the system works as follows: First it extracts from the text terms that possibly refer to the target entities as well as those that refer to their respective evidential entities. Then the disambiguation evidence model is used to compute for each extracted term the confidence that it refers to a particular target entity. The target entity with the highest confidence is expected to be the correct one. Figure 2 shows the results of executing this process on the above text about Siege of Tripolitsa.



Examples:

- Siege of Tripolitsa occured near Tripoli with Theodoros Kolokotronis being the leader of the Greeks against Turkey
- Angered over American support of his rivals in the Mexican Revolution, the peasant-born revolutionary leader Pancho Villa attacks the border town of Columbus
- Throughout the American Revolution, Shawnee warriors conducted raids against American settlements in Kentucky. In November 1782, George Rogers Clark, hoping to prevent further attacks, led 1,050 men against the Shawnee Indians living in the Miami River Valley, near Springfield
- Hanover was occupied by a detachment under the command of General Mortier. The siege of Hameln was intrusted to General Savary.
- The engagement fought in Hanov e of Gettysburg. As General Robert E. Lee moved north toward Pennsylvania in June 1863, Major G he Union Army and gather information about Federal troop movements while collecting what supplies they

Tripoli, Greece 0.50
Tripoli 0.17
Tripoli clashes (mid-October 2011) 0.17
Tripoli protests and clashes (February 2011) 0.17

Input text

Siege of Tripolitsa occured near Tripoli with Theodoros Kolokotronis being the leader of the Greeks against Turkey

Fig. 2. Semantic Entity Resolution Example

3 Conclusions and Future Work

Knowledge Tagger does not aim to be independent of the content or domain of the input texts but rather adaptable to them. That's exactly its main differentiating feature from other similar systems as our purpose was not to build another generic disambiguation system but rather a reusable framework that can be adapted to the particular characteristics of the domain and application scenario at hand and exploit them to increase the task's effectiveness.

The current version of the system's user interface is still in an early stage of development. A first line of future work will focus on adding more domain knowledge to the system's repository (other than the football and history datasets we already have) so that users are able to build evidence models for a larger range of domains. Moreover, we intend to allow users to use their own semantic data by linking our system to their repository.

References

1. Alexopoulos, P., Ruiz, C., Gomez-Perez, J.M.: Scenario-Driven Selection and Exploitation of Semantic Data for Optimal Named Entity Disambiguation. In Proceedings of the Semantic Web and Information Extraction Workshop (SWAIE 2012), Galway, Ireland, October 8-12, 2012.
2. Hoffart, J., Yosef, M.A., Bordino, I., Frstenau, H, Pinkal, M., Spaniol, M., Taneva, B., Thater, S., Weikum, G.: Robust disambiguation of named entities in text. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Stroudsburg, PA, USA, 782-792.
3. Mendes, P.N., Jakob, M., Garcia-Silva, A., Bizer, C.: DBpedia spotlight: shedding light on the web of documents. In Proceedings of the 7th International Conference on Semantic Systems, ACM, New York, USA, 1-8, 2011.
4. Rizzo G., Troncy, R.: NERD: A Framework for Evaluating Named Entity Recognition Tools in the Web of Data. In 10th International Semantic Web Conference, Demo Session, pages 1-4, Bonn, Germany, 2011.

Introducing a Diversity-Aware Drupal Extension

Simon Hangl, Ioan Toma, and Andreas Thalhammer

University of Innsbruck, Technikerstr. 21a, A-6020 Innsbruck
{simon.hangl, ioan.toma, andreas.thalhammer}@sti2.at

Abstract. This demonstration paper introduces a diversity-aware extension for the content management system Drupal. It shows how different aspects, such as automatically recognized entities, topics and sentiment scores can be leveraged in a Web user interface. We introduce a coherent approach that enables readers to navigate to further related articles. In particular, we demonstrate new ways to quickly grasp what the articles' sentiments are and which topics they cover before the actual click.

1 Introduction

Nowadays an impressive amount of data is being produced and consumed online each day introducing new challenges for technologies and tools that handle the information management life cycle, from filtering, ranking and selecting, to presenting and aggregating information. Furthermore, existing technologies and tools are based on principles that do not reflect the plurality of opinions and viewpoints captured in the information. Developing methods and software extensions to tools that leverage content analysis at large scale has become a necessity, which the RENDER project[1] is addressing. As a part of this contribution, we introduce a diversity-enabled Drupal module. Drupal is a very popular Content Management System (CMS) with – as of August 2013 – more than 983,000 users and more than 28,000 developers contributing.[2]

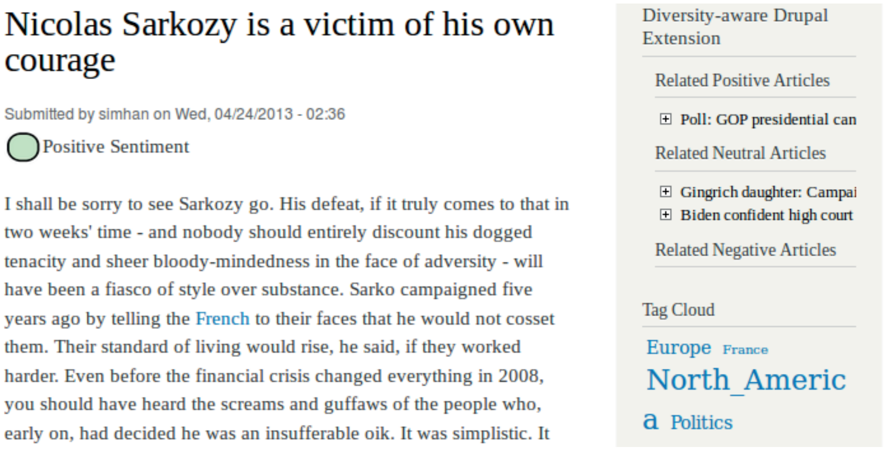
The *Diversity Enricher* Drupal extension has been developed as a show case for diversity-enabling technologies. It supports diversity-aware navigation, organization, and presentation of Drupal articles. A demo deployment can be found at `http://render-project.eu/drupal`.

2 Functionality

The *Diversity Enricher* module provides several functionalities that present and process information that can be considered to enrich Drupal articles with more *diverse* information.

[1] RENDER project – `http://render-project.eu`

[2] Numbers taken from the `http://drupal.org` landing page. Retrieved on August 8, 2013



Nicolas Sarkozy is a victim of his own courage

Submitted by simhan on Wed, 04/24/2013 - 02:36

🟢 Positive Sentiment

I shall be sorry to see Sarkozy go. His defeat, if it truly comes to that in two weeks' time - and nobody should entirely discount his dogged tenacity and sheer bloody-mindedness in the face of adversity - will have been a fiasco of style over substance. Sarko campaigned five years ago by telling the French to their faces that he would not cosset them. Their standard of living would rise, he said, if they worked harder. Even before the financial crisis changed everything in 2008, you should have heard the screams and guffaws of the people who, early on, had decided he was an insufferable oik. It was simplistic. It

Diversity-aware Drupal Extension

Related Positive Articles

⊞ Poll: GOP presidential can

Related Neutral Articles

⊞ Gingrich daughter: Campai

⊞ Biden confident high court

Related Negative Articles

Tag Cloud

Europe France

North_Americ

a Politics

Fig. 1. Article view with diversity aspects

2.1 Diversity Information Extraction

One of the most important things about the Drupal extension is the fact that no user interaction is needed to extract the necessary diversity information. This information is generated by the *Enrycher* service[3] which is publicly available. Enrycher utilizes natural language processing techniques to extract diversity information such as *topics*, *sentiments*, *sentiment scores* or *named entities* captured by the article text. This information is then described by using SIOC [1] in combination with the *Knowledge Diversity Ontology*[4] (KDO) [4].

2.2 Links to Related Articles and Topics

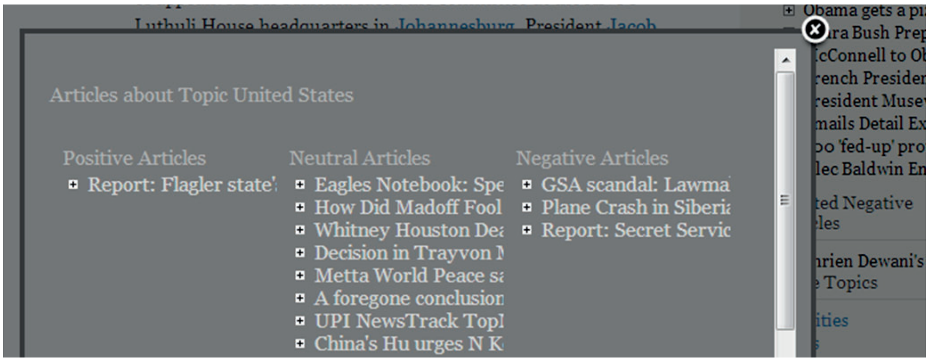
Figure 1 shows an overview on the extension. On the left hand side the original article is presented, whereas on the right hand side the main functionality of the extension is located. There, related articles within the Drupal database are listed, split up according to their extracted overall sentiment. An article is considered to be related, if it has at least one topic in common with the currently shown one and is located in the same cluster of the *Diversity-Aware Ranking Service*[5].

The topics of the related articles can be shown by clicking on the + button in front of the article titles. In addition, tags extracted from the currently shown article are presented in a tag cloud below the related article's tree. The size of the respective tag is determined by its number of occurrences in the triple store. Named entities are recognized within the text and get marked. It is possible to click on all tags, named entities, and topics in order to get articles with the same tag/topic (see Figure 2). As a further diversity feature, each article's sentiment is displayed between the title and the actual text.

[3] Enrycher – `http://enrycher.ijs.si`

[4] KDO – `http://kdo.render-project.eu/`

[5] Diversity-Aware Ranking Service – `http://ranking.render-project.eu`



Articles about Topic United States

Positive Articles	Neutral Articles	Negative Articles
⊞ Report: Flagler state'	⊞ Eagles Notebook: Spe	⊞ GSA scandal: Lawma
	⊞ How Did Madoff Fool	⊞ Plane Crash in Siberi
	⊞ Whitney Houston Dea	⊞ Report: Secret Servic
	⊞ Decision in Trayvon M	
	⊞ Metta World Peace sa	
	⊞ A foregone conclusion	
	⊞ UPI NewsTrack TopI	
	⊞ China's Hu urges N K	

Fig. 2. Related articles with topic "United States"

2.3 Export Options

The diversity data produced by the Enrycher service can be exported in the following formats: RDF+XML, JSON, and Turtle.

2.4 Import articles from a Sesame triple store

Another important function of the extension is the ability to import additional articles into the Drupal database, if they are stored in an Sesame store and are described with the SIOC [1] and KDO [4] ontologies. This option is only available through the administration interface.

3 Key technologies and implementation

The Drupal extension makes use of several technologies and tools. This section describes how the main parts of the *Diversity Enricher* Drupal extension interact.

Diversity Mining Web Services (Enrycher) The main functionality of the Enrycher service has already been described in Section 2.1. However, the Enrycher service could be replaced by any Web service that is SIOC and KDO compliant. This means, that the service has to support a subset of the SIOC and KDO functionalities - namely the extraction and proper output of *topics*, *sentiments* and *sentiment scores*.

Sesame triple store Sesame is used as data store back-end. All other components operate on the Sesame store by using SPARQL queries. The Enrycher service returns RDF data which can be directly submitted to the Sesame store. The *Diversity-Aware Ranking Service* component and the Drupal tool read from the store using a set of predefined queries.

Diversity-Aware Ranking Service This service is used to retrieve related articles with differing sentiments. It operates on a Sesame triple store. The core of the ranking service is a clustering algorithm that operates using a distance metric based on topics and sentiment scores. Articles that have at least one topic in common with the current article are preselected and then clustered by topic. All articles that are in the same cluster as the currently browsed one are then marked as related.

Drupal Integration The tool is connected to Drupal with so-called *hooks*. An implemented hook is called each time a certain event occurs. The hooks of the *Diversity Enricher* module are

- New Article Created: As soon as an article is created, the raw text data is submitted to Enrycher, which extracts the diversity information. This information is then stored to the local Sesame store.
- Article Viewed: If the article is viewed the first time and it has been in the database before the Drupal extension has been activated, this hook acts the same as in the case for *New Article Created*. Additionally, the information needed to present is generated (by using ranking, SPARQL queries) and then presented beside the raw article text.
- Article Changed: If an article is changed, Enrycher is again asked for diversity information and the store is updated with the new enrichment.
- Article Deleted: If the article is removed, all links to the diversity information is deleted from the Drupal database.

4 Related work

The integration of semantic technologies into CMSs brings clear benefits especially for improving search, integration and intelligent management of the content. During the last years several approaches have been published on how semantics can be used within CMSs in general and Drupal in particular.

Since version 7, Drupal natively supports RDF representation of posts, making use of vocabularies like SIOC, FOAF, Dublin Core, and SKOS. Although the new RDF module in Drupal easily enables publishing LOD, it does not provide means for the automatic creation of links to relevant LOD resources.

The approaches described in [2] and [3] enable the production and consumption of Linked Data in CMSs. In [2], two Drupal modules are introduced, one for creating RDFa annotations and another one for generating a SPARQL endpoint for any Drupal site out of the box. The RDFa export module also enables content providers to use their own vocabulary with RDF mappings management. [3] presents RDFaCE, a WYSIWYM (What You See Is What You Mean) editor that extends traditional WYSIWYG editors by RDF statement and RDFa output capabilities. This also enables the reuse of Linked Data sources such as DBpedia. Both approaches focus on the manual or semi-automatic annotation of articles with named entities and topics.

VIE.js[6] is a JavaScript-based semantic interaction framework. It facilitates annotation and interaction with textual and RDFa-annotated content on Web pages. It is used in combination with Apache Stanbol[7] that supports the extension of CMSs with semantic services. Another annotation framework is given by the OpenCalais[8] Drupal extension that uses the OpenCalais API of Thomson Reuters to annotate posts with named entities, facts, and events.

While the above approaches focus on the named entity or topic aspects, we introduce a new dimension given by the active utilization of automatic sentiment extraction. Eventually, this is expected to support the content creation and perception process (given a more fine-grained sentiment and opinion extraction). Also, in contrast to the above approaches, our approach focuses on providing a complete and fully automatic cycle to support the management of diversity; from text analysis and annotation to different visualization methods within Drupal.

5 Current Work

We developed a diversity-aware Drupal extension coined *Diversity Enricher*. The module is currently available at `http://drupal.org/sandbox/sti-innsbruck/1991696`. As of the time of writing (i.e., August 12, 2013) the extension is within a review process to achieve "full project status" within the `http://drupal.org` Web portal. Amongst our next steps will be the qualitative evaluation of the *Diversity Enricher* Drupal module.

Acknowledgment This research was partly funded by the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 257790 (RENDER project).

References

1. John G. Breslin, Andreas Harth, Uldis Bojars, and Stefan Decker. Towards semantically-interlinked online communities. In *The Semantic Web: Research and Applications*, volume 3532 of *Lecture Notes in Computer Science*, pages 500–514. Springer Berlin Heidelberg, 2005.
2. Stephane Corlosquet, Renaud Delbru, Tim Clark, Axel Polleres, and Stefan Decker. Produce and consume linked data with drupal! In *Proc. of the 8th Intl. Semantic Web Conf. (ISWC2009)*, Lecture Notes in Computer Science, pages 763–778. Springer, 2009.
3. Ali Khalili, Sören Auer, and Daniel Hladky. The rdfa content editor - from wysiwyg to wysiwym. In *Proc. of COMPSAC 2012*, pages 531–540. IEEE Computer Society, 2012.
4. Andreas Thalhammer, Ioan Toma, Rakebul Hasan, Elena Simperl, and Denny Vrandečić. How to represent knowledge diversity. Poster at the 10th intl. Semantic Web Conf. (ISWC2011), 10 2011.

[6] VIE.js Semantic Interaction Framework – `http://viejs.org/`

[7] Apache Stanbol – `http://stanbol.apache.org/`

[8] OpenCalais Drupal module – `http://drupal.org/project/opencalais`

Linked Soccer Data

Tanja Bergmann[1], Stefan Bunk[1], Johannes Eschrig[1], Christian Hentschel[2],
Magnus Knuth[2], Harald Sack[2], and Ricarda Schüler[1]

[1]firstname.lastname@student.hpi.uni-potsdam.de
[2]firstname.lastname@hpi.uni-potsdam.de

Hasso Plattner Institute for Software Systems Engineering, Potsdam, Germany

Abstract. The sport domain is strongly under-represented in the Linked Open Data Cloud, whereas sport competition results can be linked to already existing entities, such as events, teams, players, and more. The provision of Linked Data about sporting results enables extensive statistics, while connections to further datasets allow enhanced and sophisticated analyses. Moreover, providing sports data as Linked Open Data may promote new applications, which are currently impossible due to the locked nature of today's proprietary sports databases. We present a dataset containing information about soccer matches, teams, players and so forth crawled from from heterogeneous sources and linked to related entities from the LOD cloud. To enable exploration and to illustrate the capabilities of the dataset a web interface is introduced providing a structured overview and extensive statistics.

Keywords: Linked Data, Soccer, Information Extraction, Triplification

1 Introduction

The Linked Open Data (LOD) Cloud includes 870 datasets containing more than 62 billion triples[1]. The majority of triples describes governmental (42 %) and geographic data (19 %), whereas Linked Data about sports is strongly under-represented. Sport competition results are collected by various authorities and other parties, they are connected to events, teams, players, etc. Providing also Linked Data about sports and sporting results enables extensive statistics, while connections to further datasets allow enhanced and sophisticated analyses. Moreover, providing sports data as Linked Open Data may promote new applications, which are currently impossible due to the locked nature of today's proprietary sports databases. By enabling linkage to additional resources such as geographical, weather, or social network data, interesting statistics for the sport enthusiast can be easily derived and provide further information that would be hidden otherwise.

In this paper we describe an extensive RDF dataset of soccer data providing soccer matches, teams, and player information, collected from heterogeneous

[1] http://stats.lod2.eu/

sources and linked to LOD datasets like the DBpedia. The raw data was collected via APIs and crawling from authorities' websites, like UEFA.com or Fussballdaten.de, and is linked to further web resources for supportive information, such as Twitter postings for most recent information, Youtube videos for multimedia support, and weather information. Based on this aggregated new dataset we have implemented an interactive interface to explore this data.

2 Related Work

The BBC Future Media and Technology department applies semantic technologies according to their Dynamic Semantic Publishing (DSP) strategy [2] to automate the publication, aggregation, and re-purposing of inter-related content objects. The first launch using DSP was the BBC Sport FIFA World Cup 2010 website[2] featuring more than 700 team, group and player pages. But, the data used by the system internally is not published as Linked Data.

An extensive dataset of soccer data is aggregated by footytube. According to their website[3] the data is crawled from various sources and connected by semantic technologies, though the recipes are not described in detail. Footytube's data include soccer statistics about soccer matches and teams, as well as related media content, such as videos, news, podcasts, and blogs. The data is accessible via the openfooty API but is subject to restrictions that interdict the re-publishing as Linked Data.

Generally, it is hard to find open data about sport results, since exploitation rights are possessed by responsible administrative body organizations. An approach to liberate sport results are community-based efforts, such as OpenLigaDB[4], which collect sport data for public use. Van Oorschot aims to extract in-game events from Twitter [3]. As to the authors' best knowledge, the presented dataset provides the first extensive soccer dataset published as Linked Data, consisting of more than 9 million triples.

3 Linked Soccer Dataset

Our intention was to create a dataset including reliable information about soccer events covering as many historical data as available including recent competition results. For this purpose DBpedia as cross domain dataset is not sufficient, since soccer data in DBpedia is incomplete and unreliable.

The dataset is aggregated from raw data originating from Fussballdaten.de[5], Uefa.com[6], DBpedia[7], the Twitter feed of the Kicker magazine[8], the Sky Sport

[2] http://news.bbc.co.uk/sport2/hi/football/world_cup_2010/default.stm

[3] http://www.footytube.com/aboutus/search-technology.php

[4] http://www.openligadb.de/

[5] http://www.fussballdaten.de/

[6] http://www.uefa.com/

[7] the original http://dbpedia.org/ and German DBpedia http://de.dbpedia.org/ have been applied for matching

[8] http://twitter.com/kicker_bl_li

HD Youtube Channel[9], and weather information from Deutscher Wetterdienst[10]. Fussballdaten.de, Uefa.com, and Kicker.de offer match results and player information. The Twitter feed is used both for parsing live match data (Kicker updates its feed with live results) and to analyse free text tweets for latest news about players or teams. The time frame of our data collection ranges from the 1960s until today and is updated constantly. Updates are scheduled every matchday, while the Twitter feeds are refreshed every 30 seconds during running games. Additional leagues can be included by setting up new crawlers, or by providing an interface for manual submission. Currently, the dataset contains information about 1. and 2. Bundesliga, the Champions League, European and World Championships.

The data from these sources is converted and persistently stored as RDF triples describing resources such as soccer player, soccer teams, matches, associations, different types of in-game-events, and seasons. Each entity is referenced by a unique URI, which unites all facts, from whatever source they originate, about the entity.

For describing the information about soccer data we have created a vocabulary *Soccer Voc*[11], which extends the *BBC Sport Ontology*[1] with soccer specific classes and properties.

The dataset comprises descriptions of about 57,000 soccer players, 1,500 teams, 1,400 clubs, 1,500 referees, 1,800 managers, 700 stadiums, 38,000 matches, 97,000 goals, and 207 seasons or competition series. In total 9 million triples have been generated up to now. About 3.35 million triples originate from raw data from Fussballdaten.de and 2.10 million triples from the UEFA.com website.

In order to evaluate the quality of the matching, a percentage of matched entities has been reviewed. The correctness of these matches was confirmed by manually comparing the results to a data sample. For Bundesliga, all teams (54) and about 78 % of all players (6,790) have been matched successfully to DBpedia entities. Missing matches were mostly due to missing player entities in DBpedia.

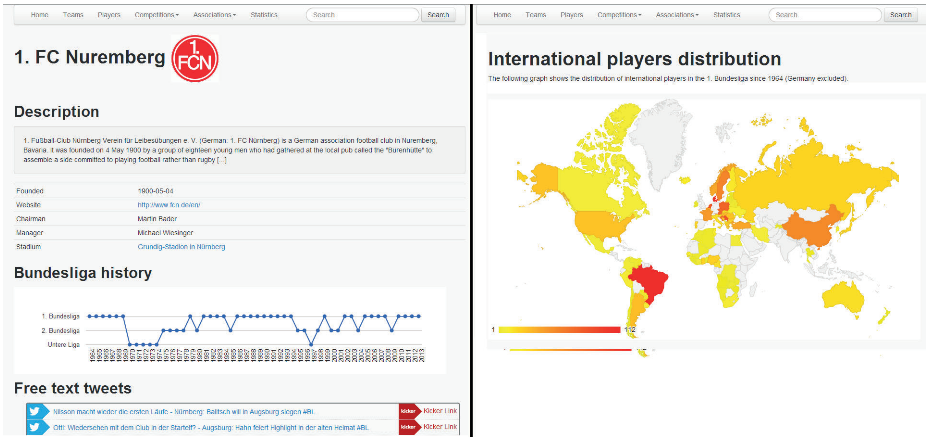
4 Application

The soccer dataset comprises a diverse amount of information, both historic and present data. As the data set contains data about every match played, it is possible to create queries for all types of entities in a soccer match, e. g. all games of a particular referee, or all games played in a specific stadium. By querying the data, the user can find interesting statistics about the world of soccer, or find information about his or hers favorite club.

The dataset can be accessed via a demonstrator website[12], where each entity is presented on its own page with relevant information, statistics, and links to

[9] http://www.youtube.com/user/SkySportHD
[10] http://www.dwd.de/
[11] http://purl.org/hpi/soccer-voc/
[12] http://mediaglobe.yovisto.com/SoccerLD/

related entities. Additionally, a variety of possible complex queries are demonstrated, such as "Which player is most important for his team?", "From which foreign country do most players in the last Bundesliga season come from?", or "Which team performs best in rainy weather?". In Figure 1, two different views of the website are shown.



1. FC Nuremberg

Description

1. Fußball-Club Nürnberg Verein für Leibesübungen e. V. (German: 1. FC Nürnberg) is a German association football club in Nuremberg, Bavaria. It was founded on 4 May 1900 by a group of eighteen young men who had gathered at the local pub called the "Burenhütte" to assemble a side committed to playing football rather than rugby [...]

Founded	1900-05-04
Website	http://www.fcn.de/en/
Chairman	Martin Bader
Manager	Michael Wiesinger
Stadium	Grundig-Stadion in Nürnberg

Bundesliga history

Free text tweets

Nilsson macht wieder die ersten Läufe - Nürnberg: Balitsch will in Augsburg siegen #BL — Kicker Link

Ottl: Wiedersehen mit dem Club in der Startelf? - Augsburg: Hahn feiert Highlight in der alten Heimat #BL — Kicker Link

International players distribution

The following graph shows the distribution of international players in the 1. Bundesliga since 1964 (Germany excluded).

Fig. 1. *Left*: Information about a German soccer club, among other a graph showing promotions and relegations (generated from match data) and free text tweets belong to this club, *Right*: Map visualization about the distribution of international players in the Bundesliga since 1963, generated from player data.

5 Conclusion and Outlook

We presented a rich soccer dataset, which is to our best knowledge the first comprehensive linked soccer dataset. We published non-restricted parts of the dataset, the publication of the dataset as a whole is prevented by legal rights belonging to the respective authorities. Applications based on this data not only allow for typical statistical information about players and matches but also exploit the advantages of Linked Data principles in order to provide additional information currently not considered by available soccer datasets. We have developed and deployed a website in order to conveniently browse the dataset and provide various statistics that exemplify the advantage of aggregating multiple resources as Linked Data.

Possible additions could include advanced and more detailed data such as the number of ball contacts, played passes, or the distance covered by a player during a match. Integrating such data even more sophisticated queries could be answered. Further extensions of the dataset include also articles from sport magazines like interviews, team presentations, or background stories of players.

References

1. S. Oliver. Enhancing the BBC's world cup coverage with an ontology driven information architecture. In P. F. Patel-Schneider, Y. Pan, P. Hitzler, P. Mika, L. Zhang, J. Z. Pan, I. Horrocks, and B. Glimm, editors, *9th International Semantic Web Conference (ISWC2010)*, November 2010.
2. J. Rayfield. Dynamic semantic publishing. In W. Maass and T. Kowatsch, editors, *Semantic Technologies in Content Management Systems*, pages 49–64. Springer Berlin Heidelberg, 2012.
3. G. van Oorschot, M. van Erp, and C. Dijkshoorn. Automatic extraction of soccer game events from twitter. In M. van Erp, L. Hollink, W. R. van Hage, R. Troncy, and D. A. Shamma, editors, *Proceedings of the Workhop on Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE 2012)*, volume 902, pages 21–30, Boston, USA, 11 2012. CEUR.

Resource Description Graph Views for Configuring Linked Data Visualizations

Bettina Steger[1], Thomas Kurz[2], and Sebastian Schaffert[2]

[1] Fachhochschule Salzburg, Campus Urstein Süd 1, 5412 Puch/Salzburg, Austria
bsteger.mmt-m2011@fh-salzburg.ac.at
[2] Salzburg Research, Jakob-Haringer-Straße 5/II, 5020 Salzburg, Austria
firstname.lastname@salzburgresearch.at

Abstract. The Linked Data movement with the aims of publishing and interconnecting machine readable data has originated in the last decade. Although the set of (open) data sources is rapidly growing, the visualization of information in this 'Web of Data' is still at a very early stage, which is primary due to the strong learning curve of semantic technologies. This paper describes an approach to visualize data 'ready-to-go' by configuration that enables Web developers and designers to build useful applications on top of the 'Web of Data'. We provide a visualization tool as a JavaScript Library, which makes it simple to aggregate Linked Data and design templates. The tool provides a way to accomplish this purely on the client using existing Web technologies, like JavaScript MVC Frameworks with data binding and JSON-LD. Based on a usability test, an evaluation is carried out by potential users, such as Web developers and semantic Web experts.

Keywords: Linked Data, visualization, client-side, json-ld, data-binding

1 Introduction

After efficiently encouraging the publication and linking of open datasets in a standardized way, the Linked Data (LD) research community is now facing the problem of creating meaningful applications on top of the Linked Open Data (LOD) Cloud. As Heath discussed in [2], the aim of these cloud interfaces is to give 'things', in the broadest sense, a central role and treat them as first-class citizens in the Web. The graph structure of the LOD Cloud, one of the big potentials regarding dynamics and distribution, makes this task particularly challenging. In many cases, complex graphs (that include various resources spread on different datasets and using several schemas and ontologies) are required in order to create a meaningful picture of a 'thing'. The gap between the understanding of the complex structures and the creation of human understandable representation makes the creation of LD User Interfaces (UI) even more difficult. In our demo, we present *visuaLOD*[3], a browser-based library that allows to split the

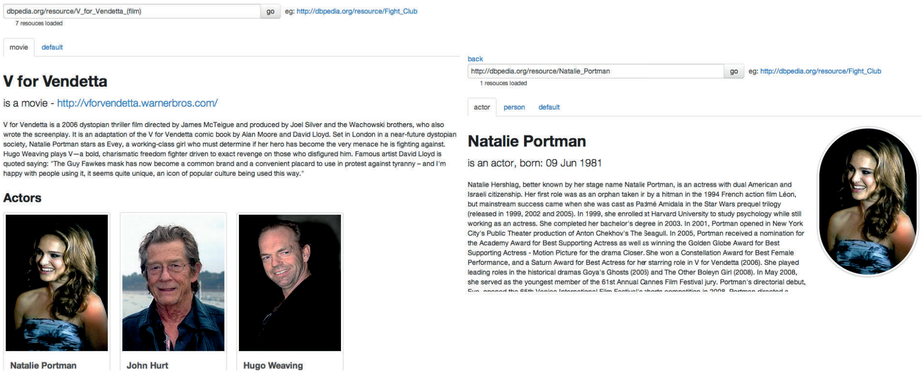
[3] https://bitbucket.org/visualod/visualod.bitbucket.org

work flow of the creation of LD applications in a Semantic Web (SW) and a UI part. This enables SW experts and Web designers to work closely together. *visuaLOD* thereby turns complex graph structures into configurable object models, using well-known technologies such as JavaScript (JS) and JSON. The interfaces themselves are built employing Google's AngularJS Framework[4] to accomplish data binding to the view. With *visuaLOD* we try to reach a linear dependency between comfort in usage and application complexity.

In this paper we will refer to two different types of Web developers: Semantic Web experts are developers having had experience with SW and advanced technologies like RDF or SPARQL. In contrast, non-expert developers do not have any relation to SW, but do know how to build Web applications with e.g. PHP, Java, JavaScript. There are already different ways to create LD visualizations. The approach presented in this paper, however, also enables non-semantic-experts to work with Linked Open Data.

2 visuaLOD - a Linked Data View Builder

Different resources need different UIs, e.g. a person could be displayed with a profile page, whereas a place should be displayed with a map. Depending on type, properties and source, presentation and actions may differ. In our demo[5], we present a simple movie mashup application visualizing movie details and starring actors. Information about the actors, e.g. their birth places (*geonames*), can be explored by following links to other datasets. To provide a flexible visualization tool, we elaborated simple configuration and abstraction of complex data structures as basic requirements.


dbpedia.org/resource/V_for_Vendetta_(film)
go
eg: http://dbpedia.org/resource/Fight_Club
7 resouces loaded
movie
default
back
http://dbpedia.org/resource/Natalie_Portman
go
eg: http://dbpedia.org/resource/Fight_Club
1 resouces loaded
actor
person
default

V for Vendetta

is a movie - http://vforvendetta.warnerbros.com/

V for Vendetta is a 2006 dystopian thriller film directed by James McTeigue and produced by Joel Silver and the Wachowski brothers, who also wrote the screenplay. It is an adaptation of the V for Vendetta comic book by Alan Moore and David Lloyd. Set in London in a near-future dystopian society, Natalie Portman stars as Evey, a working-class girl who must determine if her hero has become the very menace he is fighting against. Hugo Weaving plays V—a bold, charismatic freedom fighter driven to exact revenge on those who disfigured him. Famous artist David Lloyd is quoted saying: "The Guy Fawkes mask has now become a common brand and a convenient placard to use in protest against tyranny – and I'm happy with people using it, it seems quite unique, an icon of popular culture being used this way."

Actors

Natalie Portman

John Hurt

Hugo Weaving

Natalie Portman

is an actor, born: 09 Jun 1981

Natalie Hershlag, better known by her stage name Natalie Portman, is an actress with dual American and Israeli citizenship. Her first role was as an orphan taken in by a hitman in the 1994 French action film Léon, but mainstream success came when she was cast as Padmé Amidala in the Star Wars prequel trilogy (released in 1999, 2002 and 2005). In 1999, she enrolled at Harvard University to study psychology while still working as an actress. She completed her bachelor's degree in 2003. In 2001, Portman opened in New York City's Public Theater production of Anton Chekhov's The Seagull. In 2005, Portman received a nomination for the Academy Award for Best Supporting Actress as well as winning the Golden Globe Award for Best Supporting Actress - Motion Picture for the drama Closer. She won a Constellation Award for Best Female Performance, and a Saturn Award for Best Actress for her starring role in V for Vendetta (2006). She played leading roles in the historical dramas Goya's Ghosts (2006) and The Other Boleyn Girl (2008). In May 2008, she served as the youngest member of the 61st Annual Cannes Film Festival jury. Portman's directorial debut, Eve, opened the 65th Venice International Film Festival's shorts competition in 2008. Portman directed a

Fig. 1. Movie template is displayed because of resource's type. In addition, starring actors are fetched. Clicking on an actor's image changes the view to the right actor template. An actor's resource matches 3 different views: 'actor', 'person' and 'default'.

[4] http://angularjs.org

[5] http://visualod.bitbucket.org

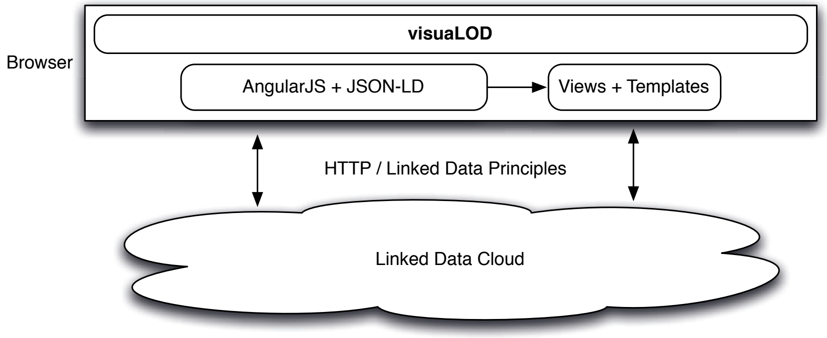


Fig. 2. Architecture: *visuaLOD* runs on the client, no need for server installation.

2.1 Resource Description Graph

Following the nature of Linked Data, the entry point of *visuaLOD* applications is a single LD resource. Taking this as a starting point, we follow dedicated links to fetch the part of the graph necessary for the information representation. We call the graph of resources and relations, needed to sufficiently visualize a 'thing' a 'Resource Description Graph' (RDG). RDGs are defined in so-called RDG views that include constraints (if a RDG is used for a specific resource), data mapping (how a graph is represented on client side) and a template (how a RDG is displayed).

The data mapping is managed with a JSON-LD Context. JSON-LD[6] is a lightweight LD serialization format based on JSON. Any RDF representation of a LD resource can be transferred into JSON-LD and vice versa. All defined properties in this **context** can be used in the template.

2.2 Work flow

The process, described in Figure 3, retrieves RDF data for the starting resource and maps it to the AngularJS model. It applies RDG views by validating their constraint part against the RDF data, fetches the additional resources and nests them into the model object. In AngularJS, the view is a projection of the model through the HTML template, so the templates are rendered in parallel.

3 Evaluation

A usability test was completed by six potential users: one SW expert and five Web developers with no prior knowledge of SW. Every user understood the given assignment and purpose of visuaLOD. The tool provides simple usage, but when

[6] http://json-ld.org/

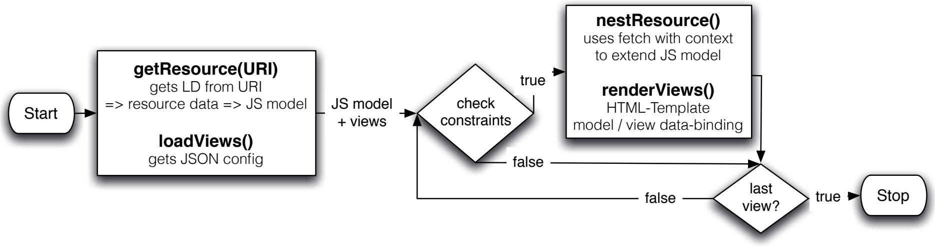


Fig. 3. Work flow

it comes to defining the JSON-LD Context the five non-semantic-Web-experts had difficulties. Feedback received from the SW expert suggests an easier usage to define the context. For example, visuaLOD could determine automatically if a property is a URI or a literal. The README[7] could be extended by screenshots and better examples on how to use visuaLOD.

4 Related Work

Fresnel[8] is a browser-independent presentation vocabulary for RDF. The main concept consists of two parts: *Lenses* and *Formats*. Lenses specify which properties of RDF resources are shown while formats indicate how to format content selected by lenses [4]. *Lenses and Formates* are defined in RDF, thus it is difficult to read and write a lens or format for non-experts.

LODSPeaKr[9] is a framework to create LD applications. It recommends to discover the data of the defined SPARQL endpoint, which means it is not suitable for all LD sources.

LESS [1] represents an approach for the visual presentation of LD resources and SPARQL query results. The process is based on the server side and uses a flexible, but proprietary templating language.

The KiWi project [3, 5] introduces perspective concepts allowing type-dependent visualization patterns. The weakness of the approach was mainly the strong coupling between back end and visualization. Nevertheless, the idea of KiWi perspectives lead to the visualization tool we presented in this demo.

All approaches introduced in the course of this chapter differ from the presented JS visualization tool: visuaLOD is purely browser-based. A server-side configuration is not required. Beyond JSON-LD works with any Linked Data server (e.g. RDF/XML can be converted to this serialization format).

[7] http://bitbucket.org/visualod/visualod.bitbucket.org

[8] http://www.w3.org/2005/04/fresnel-info/

[9] http://lodspeakr.org/

5 Conclusion and Further Work

This paper has sought to introduce a new approach to visualize Linked Data fully client-side. With *visuaLOD*, we presented a LD visualization tool that enables even non-semantic Web experts to build LD visualizations. To extend *visuaLOD* to a generic Linked Data Browser, we will provide a JS bookmarklet. It could be possible to allow the creation and storage of RDG views in an open accessible *view store*. In addition, future work could focus on the following aspects:

- Multi-language support: Detect the browsers language and show the end-users available texts in their language.
- Update: Not only read, but update LD resources using the advantages of data-binding and e.g. SPARQL update[10].
- View/Template builder: Create a UI for building views. View-Changes could automatically show how a visualization will look like.

References

1. S. Auer, R. Doehring, and S. Dietzold. LESS - template-based syndication and presentation of linked data. In *Proceedings of the 7th international conference on The Semantic Web: research and Applications - Volume Part II*, ESWC'10, pages 211–224, Berlin, Heidelberg, 2010. Springer-Verlag.
2. T. Heath. How Will We Interact with the Web of Data? *IEEE Internet Computing*, 12(5):88–91, Sept. 2008.
3. T. Kurz, S. Schaffert, T. Bürger, S. Stroka, and R. Sint. KiWi - A Platform for building Semantic Social Media Applications. In *Proceedings of the ISWC 2010 Posters and Demonstrations Track*, ISWC'10, pages 185–188, 2010.
4. E. Pietriga, C. Bizer, D. Karger, and R. Lee. Fresnel: a browser-independent presentation vocabulary for RDF. In *Proceedings of the 5th international conference on The Semantic Web*, ISWC'06, pages 158–171, Berlin, Heidelberg, 2006. Springer-Verlag.
5. S. Schaffert, J. Eder, S. Grünwald, T. Kurz, and M. Radulescu. Kiwi - a platform for semantic social software (demonstration). In *ESWC'09: The Semantic Web: Research and Applications, Proceedings of the 6th European Semantic Web Conference*, pages 888–892, Heraklion, Greece, June 2009.

[10] http://www.w3.org/TR/sparql11-update/

Types of Property Pairs and Alignment on Linked Datasets – A Preliminary Analysis

Kalpa Gunaratna, Krishnaprasad Thirunarayan, and Amit Sheth

Kno.e.sis Center, Wright State University, Dayton OH, USA
{kalpa,tkprasad,amit}@knoesis.org
http://knoesis.org

Abstract. Dataset publication on the Web has been greatly influenced by the Linked Open Data (LOD) project. Many interlinked datasets have become freely available on the Web creating a structured and distributed knowledge representation. Analysis and aligning of concepts and instances in these interconnected datasets have received a lot of attention in the recent past compared to properties. We identify three different categories of property pairs found in the alignment process and study their relative distribution among well known LOD datasets. We also provide comparative analysis of state-of-the-art techniques with regard to different categories, highlighting their capabilities. This could lead to more realistic and useful alignment of properties in LOD and similar datasets.

Keywords: Linked Data, Property Alignment, Property Pair Analysis

1 Introduction

LOD [2] has popularized the way individual datasets can be published on the Web by making inter-connections. This has resulted in the creation of a huge structured knowledge graph on the Web. Since dataset publishers are autonomous and design their datasets to meet their respective purposes for originally developing datasets, data interoperability and data integration tasks on these datasets are challenging. Property alignment is one such research problem where innovative solutions are required to handle complex data representations in these interconnected datasets that go well beyond simple string manipulations.

We introduced a novel way of computing property alignment (similarity) between interconnected datasets by exploring the available links between the datasets and using statistical measures [4]. Our solution can successfully handle complex data representations found at the property level in the matching process. We start with a breakdown of types of property pairs found on the LOD and discuss the performance of matching algorithms on the non-trivial task of property alignment between datasets. The analysis is based on manually identified and categorized property pairs of a sample of well known linked datasets in the LOD cloud. Moreover, the analysis presents how many of the manually identified property pairs in each category are identified by the different matching techniques (recall for each property type) highlighting their applicability.

2 Analysis

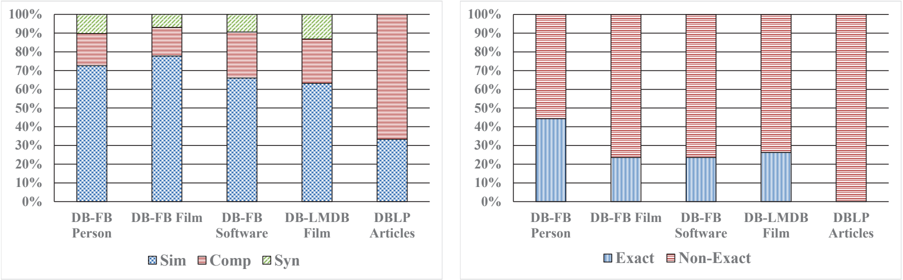
We analyze different types of property pairs found along with the experiments performed in [4]. Such analysis can provide a deeper understanding of types of property pairs that exist in linked datasets and how matching of such property pairs can be improved between two linked datasets using property extensions.

We can categorize the types of property pairs between linked datasets in two orthogonal ways: (1) on the basis of their semantics, and (2) on the basis of the techniques and tools required to determine the inter-relationships or alignment among property pairs. On the basis of semantics, the related property pairs can be classified as (1) equivalent properties or (2) those possessing a property-sub property relationship. On the basis of the techniques used to align properties, we can classify property pairs as follows:

1. *Simple property pairs*: These have high syntactic similarity in the property names and may have a common prefix, common suffix, adjectives, or different ordering of words, e.g., *birthPlace* vs *placeOfBirth*. Here the words “place” and “birth” are in a different order for the two properties.
2. *Opaque property pairs*: These have the same meaning but use different words. This can be further categorized into two parts.
   (a) *Synonymous property pairs*: Similarity of the two properties can be decided by analyzing the meaning of the property names and is intentional. This can be achieved by using an external dictionary or a lexical database like WordNet. If property name is a word phrase, similarity can be checked by removing common words from the property names, e.g., *occupation* vs *profession*, *city of birth* vs *place of birth*. In the second property pair, the common suffix can be eliminated from the comparison.
   (b) *Complex property pairs*: Similarity cannot be determined by considering property names alone, but requires additional information such as extension analysis, and domain and range. These are ambiguous or have multiple meanings but have a specific meaning in a dataset, e.g., *battle* vs *participated in conflict*, *resting place* vs *place of burial*. The two terms “conflict” and “resting place” have multiple meanings and are used in many contexts. Hence, the similarity is harder to identify.

In this analysis, we highlight the advantages of using property extensions compared to string based and external dictionary based methods that focus on analyzing property names in the matching process. We consider only object-type properties for this analysis in DBpedia, Freebase, LinkedMDB, and DBLP datasets[1], taking 5000 instances in each sample set [1]. We did not consider property chains or composite property alignment in this preliminary analysis, which belong to the complex property pair type. Composite property alignment is the process of aligning a property in one dataset with several properties (or property chains) in another dataset. There exist other efforts (within datasets), different from ours, that analyze sets of properties in RDF [6], combination of properties and classes in LOD [3], and time dynamics of LOD [5].

[1] person, film and software domains between DBpedia and Freebase, films between DBpedia and LinkedMDB, and articles in DBLP (L3S and RKB Explorer).



(a) Property pair type breakdown

(b) Exact matching of property pairs

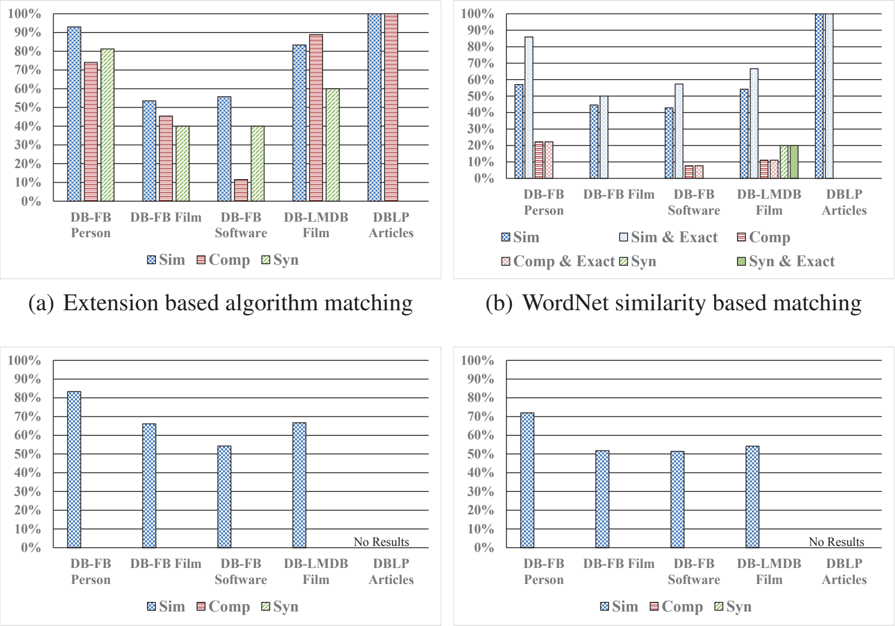
Fig. 1. Property pairs breakdown. Syn for Synonymous, Comp for Complex and Sim for Simple property pairs.

The correct matches in this analysis were manually identified and categorized by the authors and verified by an external reviewer. Figure 1 shows the breakdown of properties into the three types that we are interested in. According to Figure 1(a), the majority of the property pairs belong to simple property pairs followed by complex and synonymous property pairs. Moreover, some property pairs can be matched using exact property name matching as shown in Figure 1(b), but they account for less. Based on the facts presented in Figure 1, on average, the majority of the matching property pairs are simple, but cannot be matched using exact matching of property names.

There are different approaches for aligning property pairs between datasets including [4], which is based on property extension matching. In the extension based approach, alignment of two properties is decided by aggregating the number of matched subject-object pairs in the property extension over the number of co-appearances of the property pair in two linked datasets. We utilized Entity Co-Reference (ECR) links that exist between linked datasets in matching extensions. That is, two instances (in the property extension) are considered the same if they are connected by an *ECR* link. There can be incorrect matches for each property as extensions of properties overlap. For example, ”birthPlace” property may match to ”deathPlace” with some overlap in the extension, but when the whole result set is aggregated and analyzed, these coinsidental matches can be eliminated. For the WordNet based approach, we calculated the normalized WordNet similarity using eight similarity measures[2] found in the literature over terms appearing in the property names after removing stop words. For string similarity measurements, we added stemming in the preprocessing step before computing the similarity over property names. More details including threshold values and formulas used for matching are in [4][1].

Considering these matchers, Figure 2 shows the percentages of the correctly identified property pairs for the three types of property pairs. It also shows the superiority of the extension based approach over string based and dictionary (WordNet) based approaches. It is clear from Figures 2(a), 2(b), 2(c), and 2(d) that the extension based

[2] namely, LCH, RES, HSO, JCN, LESK, PATH, WUP and LIN



(a) Extension based algorithm matching

(b) WordNet similarity based matching

(c) String similarity based - Jaro Winkler

(d) String similarity based - Dice

Fig. 2. Matching % (recall) for each type of property pair using different approaches. Syn for Synonymous, Comp for Complex and Sim for Simple property pairs.

approach performed better and achieved the highest results in matching all three types of property pairs. We added exact matching of property names capability to WordNet based algorithm and improved its performance as shown in Figure 2(b). This is because some word phrases cannot be matched (searched) using WordNet but they have the same or common word phrases in their names. It is also interesting to note that the WordNet based approach failed to identify any of the synonymous property pairs in most of the experiments as shown in Figure 2(b). This kind of behavior is expected for string similarity or syntax based approaches, but not for a lexical database based approach like WordNet, which is specialized in synonym word categorization. Figures 2(c) and 2(d) present matching performances when the similarity of property names are considered using string matching algorithms. It is shown that string similarity based matching missed all synonymous and complex property pairs leaving them unsuitable for matching property pairs in general. Based on the facts (recall values) represented in Figure 2, extension based property alignment has the capability to identify many property pairs including complex and hidden property pairs compared to others. Furthermore, Table 1 outlines both precision and recall for each matcher for all property pair types, which also sheds lights on false positives (see [4] for more details). Note that it is not possible to provide a precision value breakdown for each property pair type, since we are not identifying each type in the alignment process but all.

	Measure Type	DBpedia-Freebase (Person)	DBpedia-Freebase (Film)	DBpedia-Freebase (Software)	DBpedia-LinkedMDB (Film)	DBLP_RKB-DBLP_L3S (Article)	Average
Extension Based Algorithm	Precision	**0.8758**	**0.9737**	0.6478	0.7560	**1.0000**	**0.8427**
	Recall	**0.8089***	**0.5138**	**0.4339**	**0.8157**	**1.0000**	**0.7145**
	F measure	**0.8410***	**0.6727**	**0.5197**	**0.7848**	**1.0000**	**0.7656**
Dice Similarity	Precision	0.8064	0.9666	0.7659	**1.0000**	0.0000	0.7078
	Recall	0.4777*	0.4027	0.3396	0.3421	0.0000	0.3124
	F measure	0.6000*	0.5686	0.4705	0.5098	0.0000	0.4298
Jaro Similarity	Precision	0.6774	0.8809	**0.7755**	0.9411	0.0000	0.6550
	Recall	0.5350*	**0.5138**	0.3584	0.4210	0.0000	0.3656
	F measure	0.5978*	0.6491	0.4903	0.5818	0.0000	0.4638
WordNet Similarity	Precision	0.5200	0.8620	0.7619	0.8823	**1.0000**	0.8052
	Recall	0.4140*	0.3472	0.3018	0.3947	0.3333	0.3582
	F measure	0.4609*	0.4950	0.4324	0.5454	0.5000	0.4867

Table 1. Alignment of object-type properties. Boldface and * mark highest and estimated values.

3 Conclusion

We provided a breakdown of types of property pairs that can be found on linked datasets in the alignment process. Even though the majority of the property pairs are simple, many cannot be identified using string manipulation techniques. In our sample datasets, 63%, 29%, and 8% of all property pairs are simple, complex, and synonymous, respectively. We have shown that in every category, extension based property pair alignment showed better results. For example, the extension based approach showed an average improvement in the range of 5% - 32% compared to simple syntactic and WordNet based approaches. Hence, we conclude that the extension (or instance) based approach can discover many property pairs that are semantically the same, which cannot be uncovered by purely syntactic means.

Acknowledgement - This work was supported by the NSF award 1143717 “III: EAGER Expressive Scalable Querying over Linked Open Data”. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. More at, `http://wiki.knoesis.org/index.php/Property_Alignment`
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. Gottron, T., Knauf, M., Scheglmann, S., Scherp, A.: A systematic investigation of explicit and implicit schema information on the linked open data cloud. In: The Semantic Web: Semantics and Big Data, pp. 228–242. Springer (2013)
4. Gunaratna, K., Thirunarayan, K., Jain, P., Sheth, A., Wijeratne, S.: A statistical and schema independent approach to identify equivalent properties on linked data. In: 9th International Conference on Semantic Systems. ACM (2013)
5. Käfer, T., Abdelrahman, A., Umbrich, J., OByrne, P., Hogan, A.: Observing linked data dynamics. In: The Semantic Web: Semantics and Big Data, pp. 213–227. Springer (2013)
6. Neumann, T., Moerkotte, G.: Characteristic sets: Accurate cardinality estimation for rdf queries with multiple joins. In: Data Engineering (ICDE), 2011 IEEE 27th International Conference on. pp. 984–994. IEEE (2011)

Automated Visualization Support for Linked Research Data

Belgin Mutlu[1], Patrick Hoefler[1], Vedran Sabol[1],
Gerwald Tschinkel[1], and Michael Granitzer[2]

[1] Know-Center, Graz, Austria
[2] University of Passau, Germany
{bmutlu,phoefler,vsabol,gtschinkel}@know-center.at
michael.granitzer@uni-passau.de

Abstract. Finding, organizing and analyzing research data (i.e. publications) published in various digital libraries are often tedious tasks. Each digital library deploys their own meta-model and technology to query and analyze the knowledge (in further text, scientific facts) contained in research publications. The goal of the EU-funded research project CODE is to provide methods for federated querying and analysis of such data. To achieve this, the CODE project offers a platform, that extracts scientific facts from research data and integrates them within the Linked Data Cloud using a common vocabulary (i.e. meta-model). To support users in analyzing scientific facts, the project provides means for easy-to-use visual analysis. In this paper, we present the web-based CODE Visualization Wizard, which aims to analyze research data visually with an emphasis on automating the visualization process. The main focus of the paper lies on a mapping strategy, which integrates various vocabularies to facilitate the automated visualization process.

Keywords: Linked Data; Visualization; Research Data; RDF Data Cube

1 Introduction

Digital libraries, which control the lifecycle of research publications (i.e. publishing and making them accessible for certain communities) mainly expose the research knowledge using domain-specific meta-models and technologies. Moreover, they only focus on some structural attributes and often don't consider the content of the publications. This domain-specificity and weakness in specifying querying attributes limit the ability to effectively find desired information, since the number of published content is continuously growing. The goal of the CODE[1] [4] [5] project is to offer a solution for this issue by providing a platform that structures (heterogeneous) research data using the RDF Data Cube Vocabulary[2] and releases them as Linked Data.

[1] CODE: http://code-research.eu/

[2] RDF Data Cube Vocabulary: http://www.w3.org/TR/vocab-data-cube/

The RDF Data Cube Vocabulary is a generic vocabulary used to describe quantitative data (e.g. research results from tables). To simplify the analysis of this data, the web-based CODE Visualization Wizard[3] has been developed, which integrates several visualizations. To achieve a Linked Data-based visualization, these visualizations (e.g. charts) should also be described semantically. For this purpose, we defined the Visual Analytics (VA) Vocabulary[4] in the form of an OWL ontology. This vocabulary is an interface between the RDF Data Cube and visualization-specific technologies, and together with the RDF Data Cube Vocabulary it forms the basis for automating the visualization process.

In this paper, we summarize the current status of the CODE Visualization Wizard and its ongoing research.

2 Related Work

Semantic description of visualizations using RDF is a new research topic and the literature, up to now, offers just a few related publications. The most significant research, the Statistical Graph Ontology [3], comes from the biomedical domain and presents a new approach to annotate visualizations semantically.

While the Statistical Graph Ontology provides a sophisticated ground for describing statistical graphs, some key issues (e.g. the description of size and color as visualization component or the datatype of a visualization component etc.) for our applications were missing. This is why we have extended this vocabulary for our Visualization Wizard.

At Stanford University, an interactive Web-based visualization system, the Vispedia [1], has been developed to visualize heterogeneous datasets. The visualization process of Vispedia is based on the integration of the selected data into an iterative and interactive data exploration and analysis process enabling non-experts to more effectively visualize the semi-structured data available. Vispedia was an inspiration for the Visualization Wizard, but being a Wikipedia plugin it only supports visualization of Wikipedia data. Also, it does not provide automatic binding of heterogeneous data onto visualizations.

3 Approach for Automated Visualization Support

In contrast to other available solutions for visualizing Linked Data [2], the CODE Visualization Wizard automatically suggests suitable visualizations based on (1) the content and structure of the provided research data and (2) semantic description of the visualizations. The following parts of our wizard contribute to these features:

Vocabularies: The RDF Data Cube is a W3C Standard and has been developed to represent statistical data as RDF. In the CODE project we use this standard to define the meta-model for the basic research data in order to capture the

[3] CODE Visualization Wizard: `http://code.know-center.tugraz.at/vis`

[4] VA Vocabulary: `http://code-research.eu/ontology/visual-analytics`

evaluation results from publications. The results are represented as a collection of observations consisting of a set of dimensions and measures, which represent the structure of the data. Dimensions identify the observation, measures are related to concrete values and attributes add semantics to them. For example: when we have a dataset representing the result of a scientific challenge (such as PAN[5]) for several teams, there will be a collection of observations with dimensions describing the teams with concrete values for the challenge result and with an attribute *percent* to identify the unit of the value it is measured in.

Our VA Vocabulary is used to represent the information about visualizations. It describes the visualization axes and other visual channels, such as color or size of visual symbols, used to visually represent the data. The vocabulary also describes suitable datatypes that can be represented by the axes and visual channels, including the allowed occurrence of the axes and visual channels. The definition of the occurrence is important to identify whether the axes or the visual channel can be instantiated only once (e.g. bar chart x-axis) or multiple times (e.g. parallel coordinates x-axis). In fact, this model is technology-independent and used by the Visualization Wizard to generate the specific visualization code. We use in our Wizard the D3[6] visualization library and Google Charts[7] to create our visualizations but as mentioned above, it is possible to use other technologies.

Currently, the Visualization Wizard supports nine different charts and a table. For the integration of each new visualization, a generator needs to be implemented, which has well-defined interfaces and can be plugged-in to the Visualization Wizard easily.

Mapping Vocabularies: The mapping between both mentioned vocabularies, the RDF Data Cube and the VA Vocabulary, is a relation from dimensions and measures of the RDF Data Cube (i.e. cube components) to the corresponding axes and visual channels of the visualization. The mapping combinations will be found based on the structural compatibility and on the datatype compatibility between a RDF Data Cube and visualizations.

The number of the dimension and measures in a RDF Data Cube is unbounded. The possible combinations (i.e. in the format dimension: measure) for each RDF Data Cube are: (1) `1:1`, (2) `1:n`, (3) `n:1` and (4) `n:n`. The structural definition of a visualization represents, how many axes/visual channels the visualization has. To find a valid mapping, the VA Vocabulary has to suggest visualizations with the same structural definition like the structural definition of the corresponding RDF Data Cube. To clarify this, let us analyze the bar chart from the Figure 1: The bar chart has two axis, `x-axis` and `y-axis`, and can only visualize RDF Data Cubes with one dimension and one measure (`1:1`). The structural compatibility is not sufficient for a valid mapping, but also the datatype compatibility. The datatype compatibility is based on the primitive datatypes[8] (string, integer, float etc.) supported by the both vocabularies

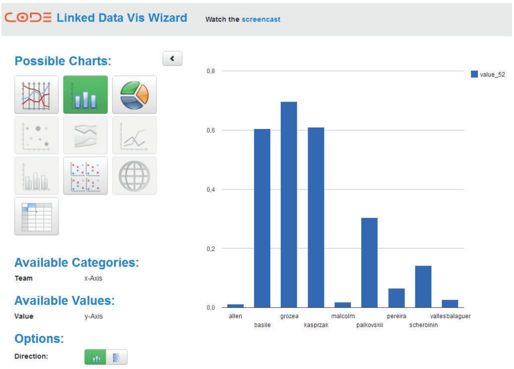
[5] PAN: `http://pan.webis.de/`

[6] D3: `http://d3js.org/`

[7] Google Charts: `https://developers.google.com/chart/`

[8] Datatypes: `http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/`

(see Visualization Process). Since the RDF Data Cube may expose composite datatypes, these must be mapped to supported primitive datatypes.



Available Categories:
Team x-Axis

Available Values:
Value y-Axis

Direction:

Fig. 1. The automatically generated visualization of PAN Data with *Team* as RDF Data Cube dimension and with a *Value* as RDF Data Cube measure.

Visualization Process: Based on the provided RDF Cube model, the Visualization Wizard proposes (1) visualizations and (2) possible variants of the mapping (see Fig. 2). The mapping is done by depicting dimensions and measures on the provided axes or on the visual channels of the visualizations. For instance, a bar chart consists of two axes: `x-axis` with a *string* and `y-axis` with a *decimal* datatype. Here, a dimension of datatype *string* will be mapped onto the `x-axis` and a measure of datatype *decimal* onto the `y-axis` (see Figure 1). However, if there are more dimensions or measures with the same datatype, we have various mapping variations for a visualization with axes which have the same datatype like these cube components. In this case (the option 2), the wizard creates a candidate table including all possible combinations between both models. The user can choose between different combinations, and for each combination, a specific visualization will be created and the provided data will be automatically visualized.

4 Conclusion and Future Work

The challenge of the first iteration in developing the CODE Visualization Wizard was to show that pitfalls of traditional visualization principles, such as the need for the manual work and high maintenance while visualizing datasets, can be effectively overcome by describing data and visualizations in dedicated vocabularies and by mapping these vocabularies. From the technical viewpoint, the main challenge was to automatically determine the right mapping between instances of the RDF Data Cube and the existing visualizations. Another, and

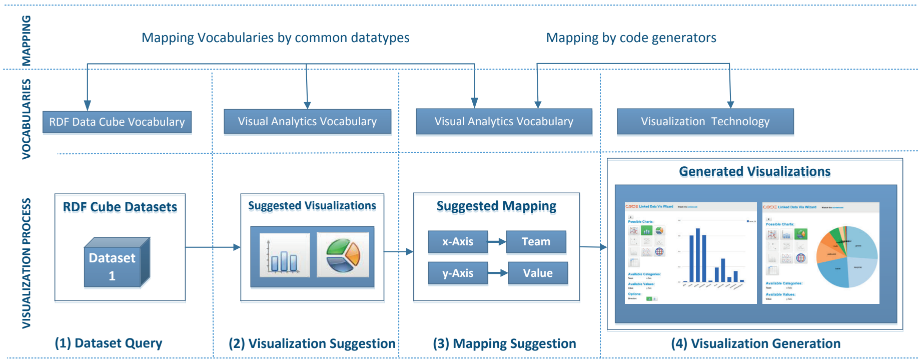


Fig. 2. Main parts of the Visualization Wizard: automated visualization process (bottom), vocabularies (middle) and mapping vocabularies (top). See live demo[3].

more serious challenge was to determine only valid suggestions among the provided visualizations.

The ongoing topics, which are parts of the project's next iterations, are (1) the investigation and the implementation of methods on how to use the previous user's knowledge (i.e. stored mappings) in order to effectively suggest mappings, (2) the extension of the automated visualization model for RDF Data Cubes with no explicit datatypes and (3) the implementation of refinement functionalities, like zooming, filtering etc.

The development of the prototype will continue throughout the rest of the year, leading to a final evaluation at the beginning of 2014.

Acknowledgement This work is being developed at the Know Center within the CODE project funded by the EU Seventh Framework Programme, grant agreement number 296150. The Know-Center is funded within the Austrian COMET Program-Competence Centers for Excellent Technologiesunder the auspices of the Austrian Federal Ministry of Transport, Innovation and Technology, the Austrian Federal Ministry of Economy, Family and Youth and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency (FFG).

References

1. Chan et al. Vispedia: Interactive Visual Exploration of Wikipedia Data via Search-Based Integration. IEEE Trans. Vis. Comput. Graphics, 14(6), 2008, 1213-1220.
2. Dadzie et al. Approaches to visualising linked data: A survey. Semant. web 2(2), 2011, 89-124.
3. Dumontier et al. Modeling and querying graphical representations of statistical data. Web Semant. 8(2-3), 2010, 241-254.
4. Seifert et al. Crowdsourcing Fact Extraction from Scientific Literature. Proc. of HCI-KDD 2013 Workshop, pp. 160-172, 2013.
5. Stegmaier et al. Unleashing Semantics of Research Data. 2nd Workshop on Big Data Benchmarking, 2012.

City Data Pipeline
A System for Making Open Data Useful for Cities

Stefan Bischof[1,2], Axel Polleres[1], and Simon Sperl[1]

[1] Siemens AG Österreich, Siemensstraße 90, 1211 Vienna, Austria
{bischof.stefan,axel.polleres,simon.sperl}@siemens.com
[2] Vienna University of Technology, Favoritenstraße 9, 1040 Vienna, Austria
stefan.bischof@tuwien.ac.at

Abstract. Some cities publish data in an open form. But even more cities can profit from the data that is already available as open or linked data. Unfortunately open data of different sources is usually given also in different heterogeneous data formats. With the City Data Pipeline we aim to integrate data about cities in a common data model by using Semantic Web technologies. Eventually we want to support city officials with their decisions by providing automated analytics support.

Keywords: open data, data cleaning, data integration

1 Introduction

Nowadays governments have a big arsenal of data available for decision support. But also city administrators need this kind of data to make better decisions and policies for leading cities to a greener, smarter, and more sustainable future. Having access to correct and current data is crucial to advance on these goals. Printed documents like the Green City Index [3] are helpful, but outdated soon after publication, thus making a regularly updated data store necessary.

Even though there is lots of data available as open data, it is still cumbersome to collect, clean, integrate, and analyze data from different sources, with different specifications, written in different languages, and stored in different formats. Sources of city data can be widely known linked open data sources like DBpedia, Geonames, or Eurostat via Linked Statistics. Urban Audit[3] for example, provides almost 300 indicators on several domains for 258 European cities. But there are also many smaller data sources which provide data in a narrow domain only, like oil prices or stock exchange rates. Furthermore several larger cities provide data from their own databases, e.g., London[4], Berlin[5], or Vienna[6]. Data is available in different formats following different data models. One can find data in RDF, XML, CSV, RTF, XLS, or HTML. The specification of the individual data fields

[3] http://www.urbanaudit.org/
[4] http://data.london.gov.uk/
[5] http://daten.berlin.de/
[6] http://data.wien.gv.at/

is often implicit only (in free text documents) and has to be processed manually for understanding. Small and medium sized cities often do not have the resources to handle these kinds of data heterogeneity and thus often miss relevant data.

With the *City Data Pipeline* we aim at providing an extensible platform to support citizens and city administrators by providing *city key performance indicators* (KPIs) based on diverse publicly available open data sources.

The project QuerioCity [5] uses partly similar techniques, but does not include an analytics component which is one of the main features of our system.

2 Architecture and Main Features

The City Data Pipeline collects data, organizes this data into indicators, and shows these indicators to the user. The system is organized in several layers which this section explains in more detail: crawler, wrapper components, semantic integration, data storage, analytics, and user interface (see Figure 1).

Crawler. The City Data Pipeline (semi-)automatically collects data from various registered open data sources in a periodic manner dependent on the specific source. The crawler currently collects data from 32 different sources, e.g., DBpedia, UN open data, Urban Audit, as well as datasets of several cities. Adding new data sources is a semi-automatic process where manual effort is necessary.

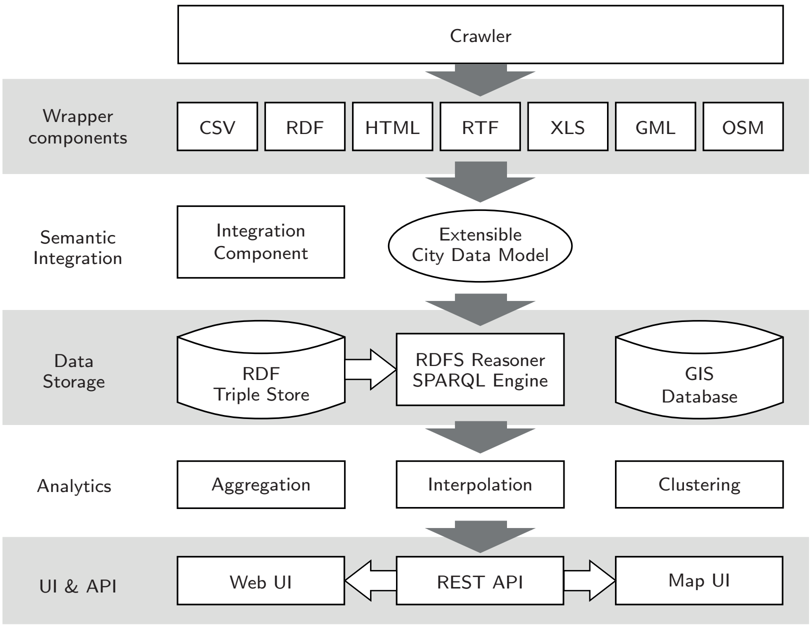


Fig. 1. City Data Pipeline architecture showing components for crawling wrapping, cleaning, integrating, and presenting information

Wrapper components. As a first step of data integration, a set of wrapper components parses the downloaded data and converts it to a source specific RDF.

The set of wrapper components include a CSV wrapper to parse and clean CSV data, a wrapper for extracting HTML tables, a wrapper for extracting tables of RTF documents, a wrapper for Excel sheets, and a wrapper for cleaning RDF data as well. All of these wrappers are customizable to cater for diverse source-specific issues. These wrapper components convert the data to RDF and preprocess the data before integrating the data with the existing triple store. Preprocessing contains the usual data cleansing tasks, unit conversions, number and data formatting, string encoding, and filtering invalid data.

Furthermore there is an OpenStreetMap (OSM) wrapper and a wrapper for GML [4] data, to feed the *geographic information system* (GIS) database.

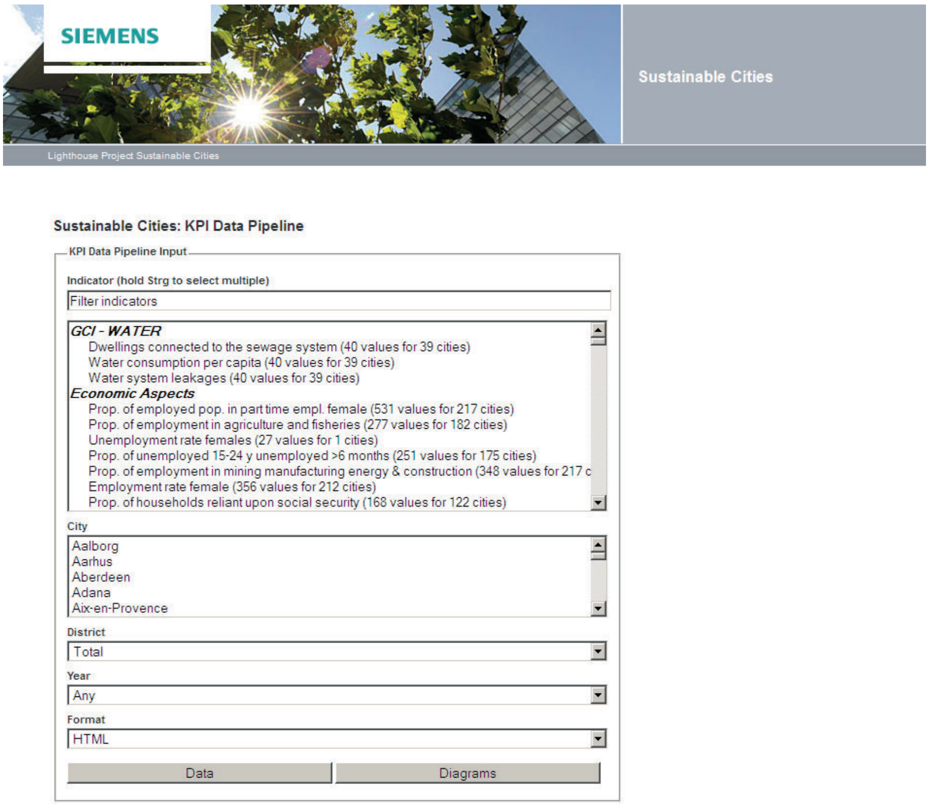
Semantic integration. To be able to access a single KPI such as the population number, which is provided by several data sources, the semantic integration component *unifies the vocabulary* used by the different data sources. The semantic integration component is partly implemented in the individual wrappers and partly by an RDFS [2] ontology (extended with capabilities for reasoning over numbers by using equations [1]) called *City Data Model.* The ontology covers several aspects: spatial context (country, region, city, district), temporal context (validity, date retrieved), provenance (data source), terms of usage (license), and an extensible list of indicators for cities. For each indicator the ontology contains descriptions and a reference to an indicator category, e.g., *Demography.* To integrate the source specific indicators the ontology maps data source specific RDF properties to City Data Model properties, e.g., it maps dbpedia:population to citydata:population by an RDFS subPropertyOf property.

Data storage. For storing the processed data we use Jena TDB[7] as *triple store* for RDF data, and PostGIS/PostgreSQL as a *GIS database* for geographic information. GIS databases allow us to compute missing information such as areas of cities or districts, or lengths of certain paths. Subsequent subsystems can access the RDF data via a SPARQL interface. The SPARQL engine provides RDFS reasoning support by query rewriting (including reasoning over numbers [1]).

Analytics. When integrated, open data contains incomplete data. Different tools in the analytics layer try to complete data by using statistical or simple algebraic methods. The analytics layer also includes tools for value aggregation as well as clustering of similar cities. We plan to extend the analytics part to allow in-depth analysis of city data to reveal hidden relationships.

User interface and API. Figure 2 shows the simple Java powered web interface. The interface also provides programmatic access via HTTP GET and HTTP POST to allow external tools such as data visualization frameworks, to query the database. The web application communicates with the Jena triple store via SPARQL 1.1 by using the Jena API directly.

[7] http://jena.apache.org/documentation/tdb/



Sustainable Cities: KPI Data Pipeline

KPI Data Pipeline Input

Indicator (hold Strg to select multiple)

Filter indicators

GCI - WATER
Dwellings connected to the sewage system (40 values for 39 cities)
Water consumption per capita (40 values for 39 cities)
Water system leakages (40 values for 39 cities)
Economic Aspects
Prop. of employed pop. in part time empl. female (531 values for 217 cities)
Prop. of employment in agriculture and fisheries (277 values for 182 cities)
Unemployment rate females (27 values for 1 cities)
Prop. of unemployed 15-24 y unemployed >6 months (251 values for 175 cities)
Prop. of employment in mining manufacturing energy & construction (348 values for 217 c
Employment rate female (356 values for 212 cities)
Prop. of households reliant upon social security (168 values for 122 cities)

City
Aalborg
Aarhus
Aberdeen
Adana
Aix-en-Provence

District: Total

Year: Any

Format: HTML

Fig. 2. Web interface for querying the City Data Pipeline, which also provides programmatic access via HTTP GET/POST

Users can select one or more of the 475 *indicators* from a list sorted by categories like *Demography*, *Geography*, *Social Aspects*, or *Environment*. The list also shows how many data points are available per indicator and for how many cities data points are available for this indicator. Next the user can select one or several of the 350 European *cities* for which we collected data. For a few cities we even have information on the individual districts available. In these cases the user can select one or several of the districts. Optionally the user can specify a *temporal context*, for which year the database should be queried. This feature allows to compare several cities with each other at a certain point of time instead of listing data of all available times. The user interface also allows the computation of *complex KPIs*. These KPIs are specified by a set of formulas in an Excel sheet and are computed on demand. Finally the system can *output* the query results as HTML report but also as XML document for further processing. With the XML export option, the web application can actually be used straightforwardly by external tools, providing for example more sophisticated visualization. One visualizer of this kind is already implemented, showing selected data points for different cities on an interactive world map.

Currently the City Data Pipeline stores an average of *285 data points per city*. Since bigger cities tend to have a wider coverage of domains, with finer granularity of time and space, the number of available data points per city is unequally distributed. While we are currently not able to provide data, ontology, or web interface for public access, we hope this changes in the future.

3 Conclusions and Outlook

The *City Data Pipeline* provides seamless access to indicators of over 30 open data providers. The system integrates data from different domains, in different formats with different data models. The City Data Pipeline allows querying and comparing indicators for many European cities thus making analytics easier.

Currently we are working on more methods for estimating missing values and predicting selected indicators based on multiple criteria. For this purpose and other kinds of data analytics we extend the data mining tool RapidMiner[8].

Furthermore we are in the process of improving the user interface to make the application more intuitive. For this purpose we use the Google Web Toolkit together with several libraries for more advanced information visualization like different kinds of interactive charts or world maps.

Acknowledgements. Stefan Bischof has been partially funded by the Vienna Science and Technology Fund (WWTF) through project ICT12-015.

References

1. Bischof, S., Polleres, A.: RDFS with Attribute Equations via SPARQL Rewriting. In: Cimiano, P., Corcho, O., Presutti, V., Hollink, L., Rudolph, S. (eds.) The Semantic Web: Semantics and Big Data, LNCS, vol. 7882, pp. 335–350. Springer Berlin Heidelberg (2013)
2. Brickley, D., Guha, R., (eds.): RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation (2004), `http://www.w3.org/TR/rdf-schema/`
3. Economist Intelligence Unit (ed.): The Green City Index. Siemens AG (2012), `http://www.siemens.com/press/pool/de/events/2012/corporate/2012-06-rio20/gci-report-e.pdf`
4. ISO: Geographic information – Geography Markup Language (GML). ISO standard 19136, International Organization for Standardization (2007)
5. Lopez, V., Kotoulas, S., Sbodio, M., Stephenson, M., Gkoulalas-Divanis, A., Aonghusa, P.: Queriocity: A linked data platform for urban information management. In: Cudré-Mauroux, P., Heflin, J., Sirin, E., Tudorache, T., Euzenat, J., Hauswirth, M., Parreira, J., Hendler, J., Schreiber, G., Bernstein, A., Blomqvist, E. (eds.) The Semantic Web – ISWC 2012, LNCS, vol. 7650, pp. 148–163. Springer Berlin Heidelberg (2012)

[8] `http://rapid-i.com/`

The Social Semantic Server

A Framework to Provide Services on Social Semantic Network Data

Dominik Kowald, Sebastian Dennerlein, Dieter Theiler, Simon Walk and
Christoph Trattner

Know-Center
Graz University of Technology
Inffeldgasse 13/5
A-8010 Graz
{dkowald, sdennerlein, dtheiler, swalk, ctrattner}@know-center.at

Abstract. This paper presents work-in-progress on the *Social Semantic Server*, an open framework providing applications and their users with a growing set of services of different granularity that utilize social and artifact network data. The *Social Semantic Server* forms a novel approach to store, query and update semantically enriched social data in order to exploit its relations within. The use of its services will be demonstrated in an exemplary use case in the health care domain.

Keywords: social semantics, semantic networks, social networks, artifact-actor networks

1 Introduction

During the last years, the popularity of social networks, such as Facebook[1] and Twitter[2], has drastically grown and much research has been performed in the field of social network analysis (an adapted combination of graph theory and network science [1] [2]). Moreover, structured and semantically enriched data have become increasingly important since, combined with social network data, they provide valuable supplementary information and make additional analysis possible [6].

The *Social Semantic Server* (*SSS*) presented in this paper is based on Artifact-Actor Networks (AANs) [5] that combine both, the classic social network and the artifact network approaches (e.g., Wikipedia[3]). It can establish more meaningful connections between artifacts and actors [5], which in turn can be further used in the respective systems (e.g., to determine topics that an author is interested in based on the articles he/she has read in order to recommend further references).

[1] `http://www.facebook.com/`
[2] `http://twitter.com/`
[3] `http://de.wikipedia.org/`

However, AANs neither explain exactly how these relations emerge nor how they can be exploited meaningfully. To address these issues, we introduce the *SSS* framework implementing services that utilize semantic technologies in dealing with social data from user-to-user interactions and/or user interactions with digital artifacts, such as texts or multimedia documents (e.g., pictures or videos).

To the best of our knowledge, the *SSS* is a novel approach to an open and extensible back-end framework equipping applications with services for exploiting and enriching social data using semantic relations.

2 Approach

The *SSS* is realized as a Java framework and can provide services of various degrees of complexity. It is accessible from within lightweight HTML applications via WebSockets or REST and from server-side applications capable of socket-based communication strategies. To increase interoperability, services that receive the application input and/or deliver the results to the requesting application use JSON for data encoding and transmission.

Together with the *SSS* framework, REST and JavaScript libraries based on WebSockets could be included in applications directly, which facilitates tying the framework to custom applications. As the set of services is supposed to grow and being extended, the core implementation of the *SSS* allows to easily register new services to its dedicated service registry. Generally, the server provides two types of services that are described below:

Low-Level Services are used to generate and enhance the semantic structures of an AAN. On the one hand, they allow to store and query semantically structured data in the form of RDF triples [3] [4] that are instantiated and further processed within the respective services. Low-level services are designed to access and update the semantic data structures in an RDF triple store (e.g., Virtuoso[4]). As users and digital artifacts are central concepts within an AAN, designated services make it possible to work with their respective representations (e.g., Java objects) that are directly mapped to the data base. This way, users and digital artifacts can be interlinked and used together with common features of social networks. On the other hand, low-level services provide functionalities, including:

- support sharing and subscribing processes of artifacts or groups of artifacts
- annotate/tag entities of the AAN with metadata and discuss digital artifacts
- handle collaborative work on digital artifacts with regard to read/write restrictions
- allow (re-)structuring of hierarchical and ordered collections of digital artifacts
- authenticate users and broadcast updates to connected applications

[4] `http://virtuoso.openlinksw.com`

- deal with digital artifacts, such as texts and multimedia documents (e.g., pictures and videos), uploaded via the respective services to an integrated Apache WebDAV[5] repository

High-Level Services use the given semantic structures formed by low-level services. They exploit explicit and implicit (social) relations to provide functions that support a personalized reflection of an AAN in order to:

- support (self-)reflection and increase the awareness of specific topics
- recommend various types of AAN entities, such as users (e.g., experts, novices), digital artifacts (e.g., discussions, collections of digital artifacts) and metadata (e.g., tags, ratings, descriptions)
- search within several entities and their relations via tags or content-based keywords (Apache Solr[6] is used for full-text indexing of all uploaded digital artifacts)

With regard to social semantic data, high-level services perform filtering, inferencing and modeling tasks utilizing metadata from digital artifacts/users, usage paths or even more complex data from artifact/user models based on semantic relations. Additional services are available, such as calculating certain indicators (e.g., the maturity of digital artifacts within an application) from the social data and the usage histories. Their results can be used by other high-level services (e.g., for searching within the AAN).

3 Applications

The *SSS* framework is currently being extended in the context of a big FP7 EU project, Learning Layers[7] (LL). The overall goal of the framework and its services is to help the participants from the health care and construction domains to build a large-scale social semantic knowledge repository that can easily be extended and used to scaffold learning episodes. In the following paragraphs we present a work-in-progress application scenario from the health care domain in the UK and elaborate on the related *SSS* service administration.

In the UK, the National Health Service establishes guidelines for General Practitioners (GPs), Diabetes Specialist Nurses, etc., for managing particular situations and diseases (e.g., diabetes) and deliver best care to locals with corresponding needs. However, the guidelines do not cover all issues encountered by GPs in practice. This triggers seeking support and related discussions, which can be facilitated via meaningful scaffolding with appropriate experts, documents, videos, pictures, etc.; e.g., up-to-date research results and worthwhile insights of GPs who face similar challenges, may help to reduce or eliminate the guidelines' ambiguities with regard to certain treatments or medications. Therefore, a tool is required that will enable GPs to state clearly defined questions for a chosen

[5] http://www.webdav.org

[6] http://lucene.apache.org/solr

[7] http://learning-layers.eu/

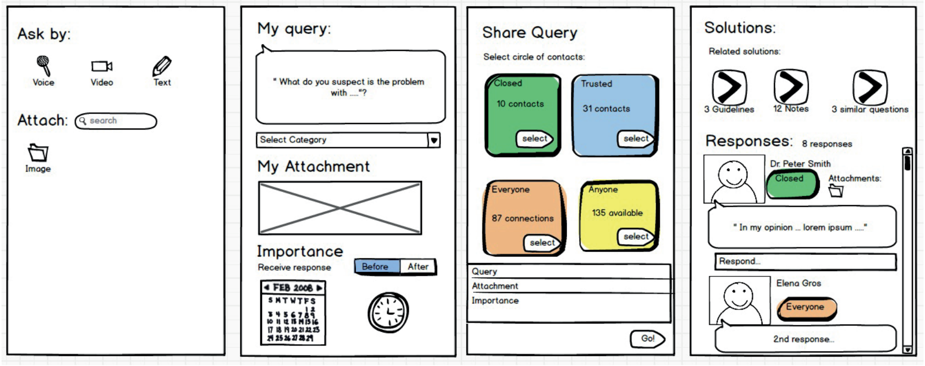


Fig. 1. Application scenario for supporting discussion and search tasks with recommendations based on AAN data (LL Design Team ’Pandora’).

group of people. By discussing and answering respective questions with the help of trusted networks and by taking advantage of the inherent knowledge from this specific kind of AAN, mentioned ambiguities can be addressed via a meaningful service instrumentation (see a dedicated application scenario sketch in Figure 1).

The *SSS* framework will support both, the basic and advanced features of the application using its low- and high-level services. The application will enable the user to enter problem statements into well-structured search queries. High-level services will deliver suitable recommendations (e.g., specialists or notes of related discussions) by exploiting the semantic structures of the knowledge base formed by low-level services. Thereby the following basic services will be used, as illustrated in the different screens of Figure 1:

The first screen shows various ways to enter problem statements (e.g., ask a question using voice, video or text) and the upload of documents as attachments to the post. The second screen contains the available annotation options (e.g., adding metadata) for classifying the question into a certain category and assigning it a certain importance based on an urgency value. Moreover, the question can be shared among different contact circles to start a discussion in the personal/trusted or extended/public social networks as demonstrated in screen three. Furthermore, the fourth screen displays the recommendation filtering in different categories and possible privacy levels (e.g., closed or everyone). The recommendation categories can be users, digital artifacts or metadata (e.g., tags), as stated in section 3.

The application scenario above describes 1 out of about 20 use cases generated by four design teams of the LL project. All of the use cases (or at least specific parts thereof) will be realized using applications based on the *SSS* services.

4 Conclusions and Future Work

This work presents the *SSS*, an open framework that enables applications to generate and use semantically enriched AAN data by integrating a growing set of dedicated services. One of the next steps will be to evaluate the framework's feasibility with regard to its services by testing the functional application prototypes in realistic field settings. For future work we plan to extend the available set of functionalities by services that provide meaningful assistance to various kinds of learners in performing their actual work tasks.

Additionally, services will be developed, which seamlessly integrate different types of vocabularies as ontologies to form a basis of semantic structures for the server. On the one hand, they could represent emerging vocabularies that are generated and utilized by users of applications via the *SSS* and on the other hand, they could represent vocabularies on a higher level of formality (e.g., not directly extendable disease classification vocabularities, such as the ICD[8]). As a result, applications could utilize metadata from different vocabularies upon various AAN entities that are needed in the respective contexts. Furthermore, we will attempt to distribute our ideas via SourceForge[9] as open-source software under the Apache License v2.

Acknowledgments: This work is supported by the Know-Center and the EU-funded project Learning Layers. The Learning Layers project is supported by the European Commission within the 7th Framework Programme. The Know-Center is funded by the Austrian COMET Program - Competence Centers for Excellent Technologies - under the auspices of the Austrian Ministry of Transport, Innovation and Technology, the Austrian Ministry of Economics and Labor and by the State of Styria.

References

1. Börner, K., Sanyal S., Vespignani A.: Network science. *Annual review of information science and technology. 41*(1) (2008): 537-607.
2. Doreian, P.: *Evolution of social networks.* Vol. 1. Routledge (1997).
3. McBride, B.: The resource description framework (RDF) and its vocabulary description language RDFS. *Handbook on ontologies* (2004): 51-66.
4. McGuinness, D., Van Harmelen, F.: OWL web ontology language overview. *W3C recommendation* 10.2004-03 (2004): 10.
5. Reinhardt, W., Moi, M., & Varlemann, T. (2009). Artefact-Actor-Networks as tie between social networks and artefact networks. *5th International Conference on Collaborative Computing Networking Applications and Worksharing*, 1-10. IEEE.
6. Van Atteveldt, W.: Semantic Network Analysis. *Techniques for Extracting, Representing, and Querying Media Content.* Charleston: BookSurge (2008).

[8] `http://www.who.int/classifications/icd/en/`

[9] `https://sourceforge.net/p/learning-layers/code/HEAD/tree/trunk/SocialSemanticServer/`

A comprehensive microbial knowledge base to support the development of *in-vitro* diagnostic solutions in infectious diseases

Magali Jaillard[1], Stéphane Schicklin[1], Audrey Larue-Triolet[1], and Jean-Baptiste Veyrieras[1]

Data & Knowledge Lab, Technology Research Department, bioMérieux SA, France
magali.dancette@biomerieux.com

1 Background

Research and development of innovative *in-vitro* diagnostic (IVD) solutions in infectious diseases require to federate up-to-date knowledge from several fields known for their complexity and their constant evolution: medical practices, microbiology and system and software engineering [6, 11, 12].
To tackle the inherent complexity of such multidisciplinary R&D projects, modern information technologies now offer powerful environments which can be leveraged to facilitate information sharing between corporate experts. This is key to ensure semantic alignments, information retrieval and then to foster decision making within the projects. The advent of almost mature semantic technologies together with international standards bring the possibility to create enterprise compliant knowledge bases. The major challenge is then to gather and link all the information from distinct and heterogeneous sources in a frequently updated and fully searchable resource. Ideally, for IVD projects, such a resource would allow for instance to map unmet needs onto current medical practices in infectious diseases, to facilitate comparison of results from different technologies, or to gather and maintain pathogen-related knowledge.
Towards this goal, we benefited from the recent efforts from the biomedical and bioinformatics communities which have been early adopters of the promising web 3.0 functionalities; multiple public data resources have developed and released domain specific ontology models or SPARQL endpoints [5, 8, 14]. Taking advantage of these semantic components we deployed on the company intranet BioPedia, a private collaborative semantic web platform carrying a cross domain knowledge base dedicated to human pathogens. The knowledge is stored on a triplestore while a wiki-based interface allows to create powerful faceted queries.

2 Methods

The current architecture of Biopedia is based on a central triplestore interfaced with sparql 1.1 compliant 4store [10] endpoint providing full sparql query and sparul update functionalities. The display and query of the triplestore content relie on several semantic wikis covering specific domains (Figure 1 A.). A benchmark of semantic solutions led our choice to MediaWiki (MW) [4] framework and

Semantic MediaWiki (SMW) [13] extensions which provide an always growing palette of querying tools. The global ontology describes four domains: bacteria and fungi strains (BioSource), taxonomy nomenclature and classification (BioTaxon), determinants and resistance mechanisms (BioGraM) and genomic data (BioSeq) (Figure 1 B.), laying on the following main classes:

- Strain: variant of a microorganism; distinct strains differ by their genomes
- Taxon: unit of close strain group, associated to a label (such as *Escherichia coli*) and a rank (for instance *species*)
- Genome: entire genetic information as chromosome and plasmid sequences
- Locus: sub-sequence of a genome annotated for its functionality
- Resistance determinant: a mutation, single nucleotide polymorphism, gene, or gene product that confers antibiotic resistance
- Antimicrobial: agent that kills microorganisms or inhibits their growth

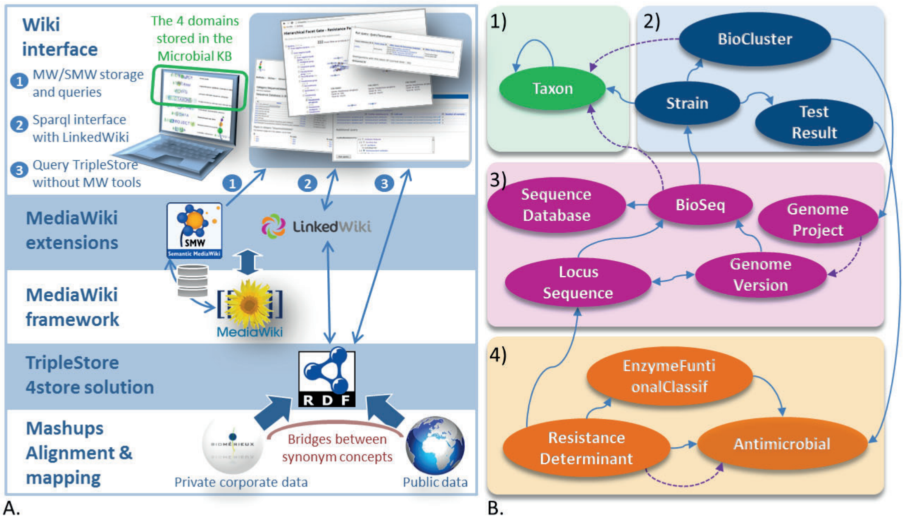


Fig. 1. Overview of the Microbial Knowledge Base structure.
A. Architecture and wiki interface of the solution, mainly based on a 4store endpoint and MediaWiki framework. **B.** Simplified view of the ontology showing the main classes and their relationships for the four domains 1) Biotaxon, 2) BioSource, 3) BioSeq and 4) BioGraM. Dashed lines represent inferred relationships.

The triplestore is populated with mashed up data mapped on the ontology. The mashup of data from heterogeneous sources includes ontology alignments, terms mapping or bridges between synonym concepts from the company and from public sources. BioTaxon domain contains bridges translating corporate identifiers to NCBI [3] taxon identifiers. These taxon identifiers are mapped using their associated taxon labels as there are the most standardized shared data. Indeed, the International Committee on Systematics of Prokaryotes [2] (ICSP) regularly publishes nomenclature rules for microbes used by the scientific community.

A crucial point to populate BioSource domain is to first identify and gather equivalent strains, *i.e.* strains issued from one unique sample and multiplied by creating subcultures. To do so we set up a clustering process using internal and external strain cross-references as edges to deduce connected components with the igraph R library [7]. We selected 75 strain reference collections, and collected strain identifiers belonging to them through StrainInfo [8], the PathoSystems Resource Integration Center (PATRIC) [9] and internal databases. Each BioCluster thus obtained was then connected to a Taxon instance. However a cross validation was necessary to highlight discrepancies: within one cluster, all strains should be tagged with the same taxon identifier. This is not the case when there are annotation or strain identification errors.

A mapping between Strain and Genome was built in order to federate public genome data from PATRIC and from our internal genome database and thus populate the BioSeq domain. Genome sequences can be processed to provide annotation that can be used as one source to populate the Locus class. Here, Loci that are registered as Resistance determinants can give a very valuable information about the strain ability to resist to antimicrobials. BioGraM is the alignment result between our corporate master data knowledge base and the Comprehensive Antibiotic Resistance Database (CARD) [1] (mainly Resistance Determinants and Antimicrobials classes).

3 Results

The current triplestore contains more than 14 million triples linking the four domains of BioPedia and allowing to infer new knowledge (Figure 2). The bridge between the taxonomy nomenclatures of NCBI and our corporate reference taxonomy was built using the taxon translator tools we developed. However 10% of the corporate labels could not be mapped on NCBI taxon labels (Figure 2 A. 3)). This is partially due to a shift of nomenclature versions between sources. Indeed, because labels are not standardized, NCBI can still uses *Fluoribacter bozemanae* when ICSP suggests *Legionella bozemanae*. For these labels, the bridge is manually completed by our expert taxonomy curator.

As shown on Figure 2 A. 1), public strains were much more reduced when gathered into clusters as there are very connected data while private strain have fewer strain cross-references to clusterize. Among the 114,383 strain clusters in which at least one strain belongs to our corporate collection, 8% were allowed to link much more metadata such as public genomes. The validation process also highlighted 2% of clusters whose strains did not share a common taxon identifier. A sparse matrix is then used to help the curator to identify the incriminated vertices.

The integration of this content in the semantic web portal BioPedia provides powerful querying tools such as a hierarchical browser to navigate within the taxonomy classification or faceted searches based on semantic properties. Together with the sparql facilitator provided by the LinkedWiki extension, this allows us bringing a solution for R&D project teams to (i) easily federate all the available data generated so far for any pathogen stored into the global strain collection or (ii) create reference strain panels based on various criteria depending on targeted diagnostic applications (Figure 2 B.).

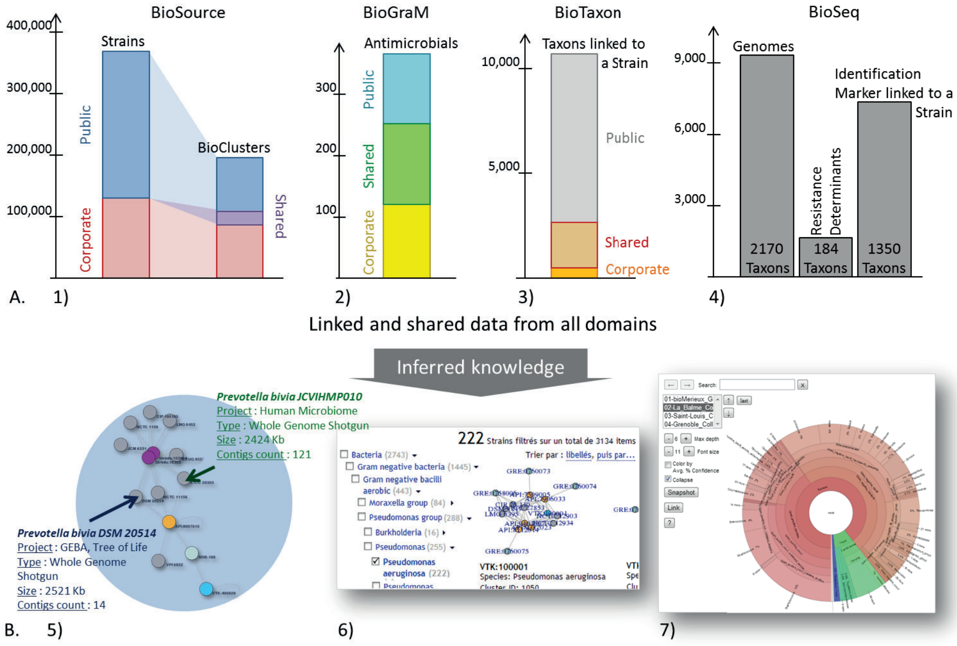


Fig. 2. Mashup results (A.) and examples of the querying capabilities (B.). **1)** Strain and clusters from public and corporate sources **2)** Alignments on antimicrobial terms. **3)** Mapping of taxon labels. **4)** Genome and locus sequence content. **5)** Inferring and chosing the best public genome related to a corporate strain. **6)** Hierarchical gate to explore clusters using the taxonomy. **7)** Global view of a strain panel, clearly showing the taxonomic classification.

4 Discussion

The microbial knowledge base provides global and uniform knowledge of the company strain collection and links it to many infectious diseases oriented public metadata, such as resistance to antimicrobials or genomes. This work gives an enriched overview of this strain collection and connects it to the achievements of the scientific community.

Then, as a side-benefit, linking data from several sources through a semantic store is of great help to improve data quality. Indeed in the mashups, sibling concepts from heterogeneous information streams are blended together and this new closeness drastically highlights the discrepancies. The data curation, even semi-automatic, is time-consuming but mandatory to build a trustworthy reference knowledge base on which powerful queries can be launched and reference datasets can be exported with confidence.

The resulting collaborative semantic web service makes possible to connect heterogeneous data in a corporate way. As the access to data is centralized, it avoids data silo and data tomb often caught out in excel spread-sheets without associated metadata. Moreover the collaborative aspect of this system encourages scientific experts to complete missing information that are then validated by a moderator, thus participating to the enrichment and quality increase of the knowledge base.

References

1. Comprehensive antibiotic resistance database, mcmaster university, canada. http://arpcard.mcmaster.ca.
2. International committee on systematics of prokaryotes). http://www.the-icsp.org/.
3. A. Acland, R. Agarwala, T. Barrett, J. Beck, D. A. Benson, C. Bollin, E. Bolton, S. H. Bryant, K. Canese, D. M. Church, et al. Database resources of the national center for biotechnology information. *Nucleic acids research*, 41(D1):D8–D20, 2013.
4. D. J. Barrett. *MediaWiki*. O'Reilly Media, Inc., 1 edition, 2008.
5. F. o. Belleau, M.-A. Nolin, N. Tourigny, P. Rigault, J. Morissette, et al. Bio2rdf: towards a mashup to build bioinformatics knowledge systems. *Journal of biomedical informatics*, 41(5):706–716, 2008.
6. L. Bissonnette and M. G. Bergeron. Next revolution in the molecular theranostics of infectious diseases: microfabricated systems for personalized medicine. *Expert review of molecular diagnostics*, 6(3):433–450, 2006.
7. G. Csardi and T. Nepusz. The igraph software package for complex network research. *InterJournal, Complex Systems*, 1695:38, 2006.
8. P. Dawyndt, M. Vancanneyt, H. De Meyer, and J. Swings. Knowledge accumulation and resolution of data inconsistencies during the integration of microbial information sources. *Knowledge and Data Engineering, IEEE Transactions on*, 17(8):1111–1126, 2005.
9. J. J. Gillespie, A. R. Wattam, S. A. Cammer, J. L. Gabbard, M. P. Shukla, O. Dalay, T. Driscoll, D. Hix, S. P. Mane, C. Mao, et al. Patric: the comprehensive bacterial bioinformatics resource with a focus on human pathogenic species. *Infection and immunity*, 79(11):4286–4298, 2011.
10. S. Harris, N. Lamb, and N. Shadbolt. 4store: The design and implementation of a clustered rdf store. In *5th International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS2009)*, pages 94–109, 2009.
11. M. Ieven, R. Finch, and A. van Belkum. European quality clearance of new microbiological diagnostics. *Clinical Microbiology and Infection*, 19(1):29–38, 2013.
12. T. A. Metcalfe. Development of novel ivd assays: a manufacturer's perspective. *Scandinavian Journal of Clinical & Laboratory Investigation*, 70(S242):23–26, 2010.
13. M. Völkel, M. Krötzsch, D. Vrandecic, H. Haller, and R. Studer. Semantic wikipedia. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 585–594, New York, NY, USA, 2006. ACM.
14. P. L. Whetzel, N. F. Noy, N. H. Shah, P. R. Alexander, C. Nyulas, T. Tudorache, and M. A. Musen. Bioportal: enhanced functionality via new web services from the national center for biomedical ontology to access and use ontologies in software applications. *Nucleic acids research*, 39(suppl 2):W541–W545, 2011.