Proceedings of the I-SEMANTICS 2012 Posters & Demonstrations Track, pp. 6-11, 2012.

Matching Linked Open Data Entities to Local Thesaurus Concepts

Peter Wetz[1], Hermann Stern[1], Jürgen Jakobitsch[2], and Viktoria Pammer[1]

[1] Know-Center GmbH
Inffeldgasse 21a, 8010 Graz, Austria
{pwetz,hstern,vpammer}@know-center.at
http://www.know-center.at
[2] Semantic Web Company GmbH
Neubaugasse 1, 1070 Vienna, Austria
j.jakobitsch@semantic-web.at
http://www.semantic-web.at

Abstract. We describe a solution for matching Linked Open Data (LOD) entities to concepts within a local thesaurus. The solution is currently integrated into a demonstrator of the PoolParty thesaurus management software. The underlying motivation is to support thesaurus users in linking locally relevant concepts in a thesaurus to descriptions available openly on the Web. Our concept matching algorithm ranks a list of potentially matching LOD entities with respect to a local thesaurus concept, based on their similarity. This similarity is calculated through string matching algorithms based not only on concept and entity labels, but also on the "context" of concepts, i.e. the values of properties of the local concept and the LOD concept. We evaluate over 41 different similarity algorithms on two test-ontologies with 17 and 50 concepts, respectively. The results of the first evaluation are validated on the second test-dataset of 50 concepts in order to ensure the generalisability of our chosen similarity matches. Finally, the overlap-, TFIDF- and SoftTFIDF-similarity algorithms emerge as winners of this selection and evaluation procedure.

Keywords: linked open data, dbpedia, thesaurus, similarity, evaluation, concept matching

1 Introduction

A solution for matching Linked Open Data (LOD) entities to concepts within a local thesaurus is specified in this paper. The solution is currently integrated into a demonstrator of the PoolParty thesaurus management software. The underlying motivation is to support thesaurus users in linking locally relevant

concepts in a thesaurus to descriptions available openly on the Web. This "linking" has a technical (realising an RDF triple) and a conceptual (realising that other, possibly complementary, descriptions of the same entity exist) component. The strategy of linking LOD entities to a local thesaurus uses the concepts of Linked Data to expand and enrich the information stored in the thesaurus ultimately leading to a more valuable knowledge base.

Since the maturing of Semantic Web technologies, and the massive emergence of LOD repositories in many domains[3], the LOD cloud presents a valuable source of knowledge. When managing a thesaurus, this source can be tapped into, either loosely in the sense of exploring additional, openly available information, or by creating an RDF triple that technically links the local concept to a LOD entity.

Naturally, others have explored the challenges and possibilites around concept matching (e.g., in the field of schema matching and ontology matching). Specifically for interlinking LOD entities, Raimond et al. [2] for instance describe two naïve approaches using literal lookups to interlink music datasets as well as an explorative graph matching algorithm based on literal similarity and graph mappings. Waitelonis and Sack [3] use matching algorithms to map labels of their yovisto video search engine to DBpedia entities. Mendes et al. [1] describe with DBpedia Spotlight a service that interlinks text documents with LOD from DBpedia. Similar to these works, we experiment with a mixture of string similarity and exploiting the graph nature of both the local thesaurus and LOD entities.

In the live demo, participants will be able to create a new thesaurus with PoolParty, or use an existing thesaurus, and enrich it with the presented matching algorithm with LOD entities from DBpedia. Participants will thus be able to gauge the usefulness of such a semi-automatic data linking themselves.

2 Problem Statement

The problem which we describe the solution for in this paper is the following: Given a specific concept in a local thesaurus, and a list of potentially matching LOD entities, which LOD entity is most similar to the local thesaurus concept?

We assume that typically both the local concept and the given list of LOD entities have a context, i.e. will have additional properties that describe them, such as a verbal description, a categorisation etc. We delegate the task of finding "potentially matching LOD entities" to a LOD lookup service, that queries the LOD cloud with a request that stems from the local concept's label.

This approach can be called *interlinking of entities*, *alignment of entities*, *semantical enrichment of data*, *augmenting data with LOD* or *entity reconciliation*.

[3] media - `http://data.nytimes.com/`, geography - `http://www.geonames.org/`, encylcopedic knowledge - `http://dbpedia.org`

3 Solution

A *lookup service* is responsible for finding potentially matching LOD entities by matching concept labels to labels of LOD entities. This lookup service can be used with any LOD SPARQL[4] endpoint. We also investigated on how much context information should be taken into account when querying for potentially matching LOD concepts.

Contextual information is integrated by adding the literal string values of connected properties of the query's entity into the similarity comparison process. In the SKOS[5] syntax, which all thesauri of this system are based on, these properties are represented as *broader* (describing hierarchically more general entities), *narrower* (describing hierarchically more specific entities) and *related* (describing similar entities) links to other entities in the same thesaurus. This additional information describes the entity in more detail, furthermore helping to deal with ambiguous terms and getting more precise results. In our current implementation we take into account broader, narrower and all related concepts of the local thesaurus concept which lead to satisfying results. In theory it is also possible to only use a subset of these contextual properties.

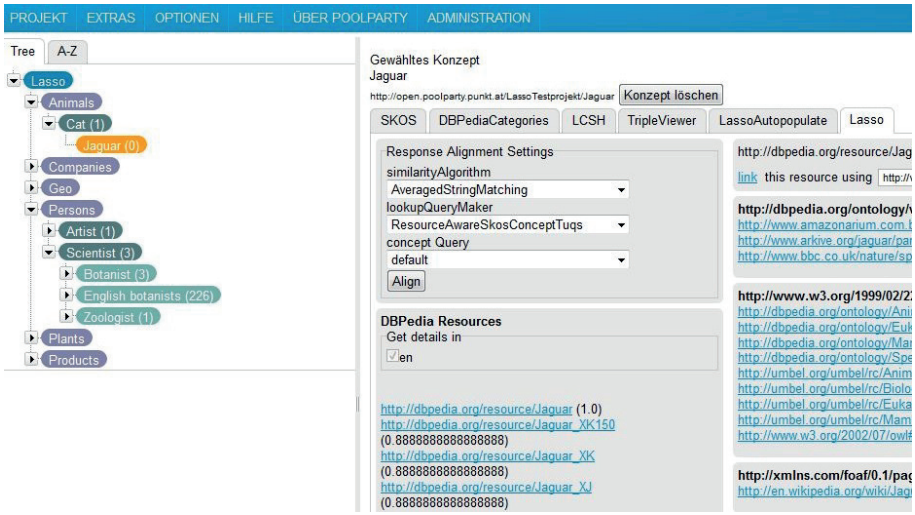

Fig. 1. DBpedia Lookup for concept "Jaguar".

As can be seen in Figure 1 the entity *Jaguar* has *Cat* as a broader concept. This relation will pour into the query as contextual information, which will yield the animal called Jaguar as a result. If the query would be triggered by

[4] http://www.w3.org/TR/rdf-sparql-query/
[5] http://www.w3.org/2004/02/skos/

choosing the *Jaguar* from the *Products* branch of the thesaurus, and therefore using other contextual information, the famous car would be on top of the results.

3.1 Implementation

The *concept matching algorithm* is responsible for comparing the local concept and the potentially matching LOD entities based on a similarity algorithm. Depending on the similarity algorithm, strings of labels and of different properties are compared to each other. Coefficients are calculated and the resulting similarity values determine the ranking of the LOD entities (the highest ranking is the LOD entity most similar to the local concept). The ranked list of LOD entities is visible for users in the PoolParty demonstrator as described in the next paragraph.

In Fig. 1 we show a *classical Semantic Web disambiguation example* in terms of the PoolParty user interface of our concept matching algorithm: the user wants to connect a concept with the preferred label *Jaguar* to the appropriate counterpart in DBpedia. The concept matching algorithm grabs the labels of concepts that are connected to Jaguar, which are *cat* and *animal*. These labels are compared to the labels of candidate resources from DBpedia using the active similarity algorithm. We see that the resource *Jaguar* referring to the cat is on the top of this list, followed by several resources referring to cars of that name (see bottom middle part of Figure 1). The DBpedia facts of the selected Jaguar resource are displayed on the right hand side. If the top-ranked LOD entity indeed describes the same real-world entity as the local thesaurus concept, then these concepts can be linked through the graphical user interface, which technically corresponds to creating an RDF triple relating both concepts.

3.2 Method and Result of Selecting a Similarity Algorithm

We selected the default similarity algorithm which the final presented ranking is based on by comparing the performance of 41 similarity algorithms on two test-datasets of 17 and 50 concepts, respectively. The first dataset includes general ambiguous terms to enable testing of the algorithm's efficiency regarding disambiguation. To get further insight into the datasets please register a demo account[6] to be able to browse directly using the PoolParty System.

In order to ensure generalisability, we compared the performance of the best algorithms on the first dataset with their performance on the Reegle[7] thesaurus - which consists of concepts dealing with clean energy - by extracting and using 50 concepts. In both cases the algorithms *overlap*, *TFIDF* and *SoftTFIDF* performed very well (see Table 1).

[6] `http://poolparty.punkt.at/de/try-it/`
[7] `http://www.reegle.info/`

4 Discussion and Outlook

In our selection and evaluation procedure of similarity algorithms, the overlap algorithm worked very well for both test ontologies. It simply checks how many of the terms in the query are also found in each result and then calculates a coefficient. On the second and third rank there are similar algorithms, which only differ in parameters dealing with tokenisation (TFIDF and SoftTFIDF). The TFIDF algorithms calculate a so-called corpus of all words including the query and all results. Based on this corpus the relevancy of each result compared to the query is computed. Overall, an accuracy of about 80% can be achieved resulting in a meaningful and efficient linkage of local thesaurus entities with entities from remote LOD repositories. Additionally, our results indicate that the winning similarity algorithms will perform well also on ontologies of other domains.

In an implementation where all complexity should be hidden from the user, one of these algorithms would be selected as the default (and probably only) similarity algorithm. Alternatively, a "voting" mechanism that always involves all three algorithms is conceivable.

To sum up, the integration and usage of SKOS principles helping us to gain contextual information for the queries, the high accuracy of top ranked algorithms and the confirmation that the overlap and TFIDF algorithms work best are a major contribution to findings which have already been made in related work.

#	Algorithm	Points	#	Algorithm	Points
1	overlap	0,823	37	qGramsDistance (qg2)	0,507
2	overlap (ws)	0,823	38	MatchingCoefficient (qg3)	0,477
3	overlap (qg3)	0,765	39	levenshtein	0,470
4	TFIDF	0,749	40	NeedlemanWunsch	0,320
5	SmithWaterman	0,725	41	stringTFIDF	0,318

Table 1. A list of the top and bottom five ranked algorithms after both evaluations. *ws* means whitespace-tokenisation; *qg2* and *qg3* mean qgram2- and qgram3-tokenisation, respectively. *Points* is a relative number to 1. 1 meaning all results would have been ranked correctly. Please find the complete table at `http://bit.ly/O7ufgk`.

Acknowledgements. The Know-Center is funded within the Austrian COMET Program - Competence Centers for Excellent Technologies - under 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. This work has been co-funded by the FFG project LASSO (Fit-IT).

References

1. Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. DBpedia Spotlight: Shedding Light on the Web of Documents. In *Proceedings of the 7th International Conference on Semantic Systems (I-Semantics)*, 2011.
2. Yves Raimond, Christopher Sutton, and Mark Sandler. Automatic Interlinking of Music Datasets on the Semantic Web. 2008.
3. Jörg Waitelonis and Harald Sack. Augmenting Video Search with Linked Open Data. In *Proc. of Int. Conf. on Semantic Systems 2009, i-Semantics 2009*, 2009.