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

Christian Hachenberg and Thomas Gottron

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

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

1 Introduction

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

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

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

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

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

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

2 Task Definition

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

2.1 Formal Definition of the Task

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

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

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

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

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

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

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

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

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

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

2.2 Example

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

3 Approaches

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

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

3.1 Keyword Based Web Search

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

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

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

3.2 Search Making Use of the Link Structure

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

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

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

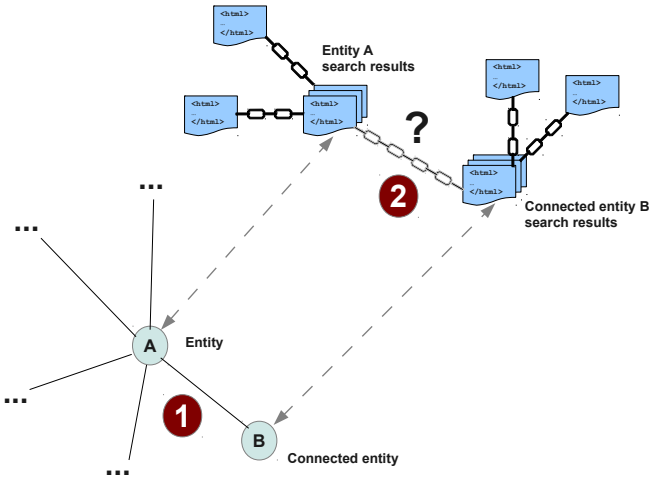

Fig. 1. Sketch of a typical link network originating from entity A with corresponding entities being connected in the knowledge base (e.g. entity B). **(1)** Entity A is connected to another entity B in the knowledge base. Their labels are used to find a result set of URLs via a web search engine **(2)**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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

3.3 Data Source Filtering Using Semantic Typing

The last category of approaches makes use of the semantic typing of entities in the knowledge base. The hypothesis for this approach is that entities of the same type are typically found together at the same location on the Web. Therefore, by querying the web for several entities of the same type we can observe web sources ranking repeatedly

high for this type. Such knowledge can be used to filter results sets by removing web documents from the result list which did not appear repeatedly.

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

4 Experiments and Evaluation

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

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

4.1 Selecting Entities for Evaluation

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

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

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

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

4.2 Construction and Evaluation of the Web Document Collection

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

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

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

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

4.3 Retrieval Performance of the Algorithms

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

5 Related Work

Our approach makes use of Linked Data [2, 4] as a source of structured data whereas the purpose is finding good (or appropriate) URLs on the document web aiming for a preferably comprehensive representation of the given entity. To the best of our knowledge there have not been any efforts to address this problem to date. Though, our work relates to several topics in varying degrees. The probably most related area is on generating structured queries and applying it to unstructured data like the document web in one way or another. Similarly to us, some works use a search engine and corresponding keywords to transform queries on structured data to comprehensible syntax for web search engines [12]. The results (documents) are often ranked, as well. In order to raise precision and as a follow-up, n-tuples [19] or simply facts [6] are extracted using information extraction methods [13]. However, some works rather focus on the generation or

extraction of entities or objects from unstructured data starting with structured queries [20]. In fact, keyword search also plays a crucial role in semantic search [5] itself where it is also used for entity/object retrieval [9, 8]. More elaborated work comes up with an entity relevance model (ERM) based on keywords from entities which in their context is used to generalize SPARQL queries on different RDF datasets [10] or to improve RDF ranking [3]. The results both of some of the works mentioned as well as our approach can be used to enrich datasets of Linked Data which has already been described in e.g. [17] using information extraction. Other works also trying to achieve this or in parts are e.g [18, 22] in the so-called Small Web of organizations etc. They learn relations for taxonomies from websites by utilizing the hierarchical links between organizational webpages not only within a single page.

6 Conclusions

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

7 Acknowledgements

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

References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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