Identifying Relevant Sources for Data Linking using a Semantic Web Index

Andriy Nikolov
a.nikolov@open.ac.uk
Knowledge Media Institute
Open University
Milton Keynes, UK

Mathieu d'Aquin
m.daquin@open.ac.uk
Knowledge Media Institute
Open University
Milton Keynes, UK

ABSTRACT

With more data repositories constantly being published on the Web, choosing appropriate data sources to interlink with newly published datasets becomes a non-trivial problem. While catalogs of data repositories and meta-level descriptors such as VoiD provide valuable information to take these decisions, more detailed information about the instances included into repositories is often required to assess the relevance of datasets and the part of the dataset to link to. However, retrieving and processing such information for a potentially large number of datasets is practically unfeasible. In this paper, we examine how using an existing semantic web index can help identifying candidate datasets for linking. We further apply ontology schema matching techniques to rank these candidate datasets and extract the sub-dataset to use for linking, in the form of classes with instances more likely to match the ones of the local dataset.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Storage and RetrievalInformation Search and Retrieval

Keywords

Data fusion, data linking, linked data

1. INTRODUCTION

The fourth principle of Linked Data[1] recommends to include links to other URIs so that more information can be obtained by following the links. In order to do that, data publishers must be aware of other repositories containing relevant data and be able to find existing resources which can be reused or linked to. With the growing number of repositories published within the Linked Data initiative, identifying such datasets and resources can become problematic. As a result, data publishers usually only link their datasets to the popular repositories (such as DBPedia[2] and Geonames[3]). This may not always be the optimal solution in some cases, for example:

- If the data domain is highly specialised and not covered by popular repositories in sufficient details.
- If different parts of the dataset are covered by several external repositories: e.g., when a repository contains references to scientific publications both on computer science (described by DBLP[4]) and medicine (described by PubMed[5]).

To support identifying different sources, catalogs of Linked Data repositories are maintained (e.g., CKAN[6]), and meta-level descriptors of repositories are provided using the VoiD vocabulary[7]. However, these sources can still be insufficient as they do not take into account the distribution of instances in repositories. For example, several repositories contain information about academic researchers, however, they use different criteria to include individuals: e.g., DBPedia only mentions the most famous ones, DBLP only includes Computer Science researchers, and RAE[8] deals with researchers working in UK institutions. In order to be able to choose the most appropriate repositories to link to, one must have access to complete instance-level data stored in them. Obtaining these data directly from the data sources and analysing them is often not feasible due to the size of datasets which need to be downloaded.

This instance-level information, however, is collected by semantic indexes such as Sindice [7] or Openlinksw[9] and can be accessed using keyword-based search. In this paper we describe an approach which utilises keyword-based search to find initial candidate sources for data linking, and ontology matching techniques as a way to assess the relevance of these candidates. The approach involves two main steps:

- Using a subset of labels in the newly published data as keywords to search for potentially relevant entities in external data sources.
- Using ontology matching techniques to filter out irrelevant sources by measuring semantic similarities between classes used to structure data.

The rest of the paper is organized as follows. Section 2 briefly outlines the use case which provided the main motivation for this work. Section 3 describes our approach in

[1] http://www.w3.org/DesignIssues/LinkedData
[2] http://dbpedia.org
[3] http://www.geonames.org/
[4] http://dblp.l3s.de/
[5] http://www.ncbi.nlm.nih.gov/pubmed/
[6] http://ckan.net/
[7] http://semanticweb.org/wiki/VoiD
[8] http://rae2001.rkbexplorer.com/
[9] http://lod.openlinksw.com/

LDOW2011, March 29, 2011, Hyderabad, India.

more detail. Section 4 discusses the results of the initial experiments we performed to test our algorithm. Finally, section 5 concludes the paper.

2. MOTIVATION

The problem of determining a set of relevant repositories is a generic one and can occur in different contexts. One of the tasks within the SmartProducts project[10] involves reusing the data from external semantic repositories to build knowledge bases for smart consumer devices: e.g., to extend the core domain knowledge base of food recipes for a smart kitchen with nutritional data, alternative recipes, health profiles of food products, etc. In order to extend the core domain knowledge base, the developer has to be able to find relevant repositories on the Web of Data and interlink them with this core knowledge base.

In another scenario, the *data.open.ac.uk* repository[11] aims at publishing various data related to the activities of The Open University (OU)[12] according to Linked Data principles. These datasets include, among others, the publications originated by OU researchers, courses provided by the university, etc. Many entities referenced in these datasets are also mentioned in other public repositories. Thus, in order to facilitate data integration, it makes sense to create links from instances used in the *data.open.ac.uk* datasets to external semantic data stores. Given the range of categories to which data instances belong, it is difficult to select a single external source to link to: e.g., publication venues can be linked to different subsets of RKBExplorer, DBLP, PubMed, DBPedia, or Freebase. Moreover, the repository is constantly extended with more instance data for existing topics (e.g., as more research output is published with time) as well as with more topics (as more internal datasets are released online). Selecting relevant sources for linking and selecting specific individuals to link to within these sources becomes a time-consuming procedure, which needs to be automated as much as possible.

There are several factors which can guide the selection of the repository for linking, in particular:

- *Degree of overlap.* In order to maximise the possibility to reuse external descriptions, the sources which contains more references to the entities stored in the newly published repository are preferable.

- *Additional information provided by the source.* When selecting a source to link to, it is important to take into account how much additional information about entities is provided by each external source: i.e., what properties and relations are used to describe these entities.

- *Popularity of the source.* Linking to URIs defined in a popular data source or reusing them makes it easier for external developers to find the published data and use them.

Among these factors, only the degree of overlap heavily relies on instance-level data stored in external repositories. The level of detail of instance descriptions can be obtained from the domain ontology used by the external dataset and, possibly, a few example instances, while the popularity of the source can be estimated based on VoiD linkset descriptors. Therefore, when designing our algorithm, we primarily focused on estimating the degree of overlap between the internal dataset prepared for publishing and potentially relevant external datasets.

3. ALGORITHM

The task of finding relevant repositories assumes that there is a dataset to be published $D_p = \{O_p, I_p\}$ containing a set of individuals I_p structured using the ontology O_p. Each individual belongs to at least one class c_λ defined in O_p: $I = \{i_j | c_\lambda(i_j), c_\lambda \in O_p\}$. On the Web there is a set of Linked Data repositories $\{D_1, \ldots, D_n\}$ such that $D_j = \{O_j, I_j\}$. There is a subset of these repositories $\{D_1, \ldots, D_m\}$ which overlap with D_p, i.e., $\forall (j \leq m) \exists (I_j^O \subseteq I_j)$: $I_j^O = \{i_k | equiv(i_k, i_p), i_j \in I_j, i_p \in I_p\}$, where *equiv* denotes the relation of equivalence between individuals. The meaning of the equivalence relation here depends on the intentions of the data publisher and the type of links (s)he wants to generate: e.g., *owl:sameAs* links or direct reuse of URIs assume that URIs must be strictly interchangeable while *rdfs:seeAlso* may only assume some kind of similarity (see [2] for the analysis of different types of identity). The goal is to identify the subset of relevant repositories $\{D_1, \ldots, D_m\}$ and to rank them according to the degree of overlap $|I_j^O|/|I_p|$. Given that the publisher may want to select different repositories to link for different categories of instances in D_p, then for each class $c_\lambda \in O_p$ a separate ranking should be produced based on the degree of overlap for instances of this class $|I_{j\lambda}^O|$, where $I_{j\lambda}^O = \{i_k | equiv(i_k, i_p), i_p \in I_p, c_\lambda(i_p)\} \subseteq I_j^O$.

Since the actual discovery of links is usually performed by an automated tool (such as Silk [8] or KnoFuss [5]), another important task is to restrict the search space for this tool by identifying in each dataset D_j a set of relevant classes c_{jk} which contain potentially overlapping individuals with c_λ. Then the tool can be configured to select only individuals of these classes as candidates for linking.

The main obstacle with these tasks is the need to identify the overlapping subset of instances $|I_j^O|$ from each external dataset. Downloading whole datasets or applying data linking tools to their complete sets of instances is often unfeasible due to their size and required computational time, network load, and local disk space. Thus, the degree of overlap has to be estimated, and keyword search services can be utilised to perform this task.

3.1 Using keyword search to find potentially relevant sources

We assume that a semantic keyword search service takes as its input a set of keywords $K = \{k_1, \ldots, k_i\}$. As output, it returns a set of potentially relevant individuals which may belong to different repositories: $I^{res} = I_1^{res} \cup I_2^{res} \cup \ldots \cup I_m^{res}\}$, where $I_j^{res} \subseteq I_j$. For returned individuals $i_{jk} \in I_j^{res}$, their types $\{c_{jk\lambda} | c_{jk\lambda}(i_{jk})\}$ are also available in the search results. An example of the search service which satisfies this assumption is Sig.ma [6], which uses Sindice as its search index.

In order to find potentially relevant individuals for individuals from the newly published dataset D_p, we query the search service using the labels of individuals (values of

[10] http://www.smartproducts-project.eu

[11] http://data.open.ac.uk

[12] http://www.open.ac.uk

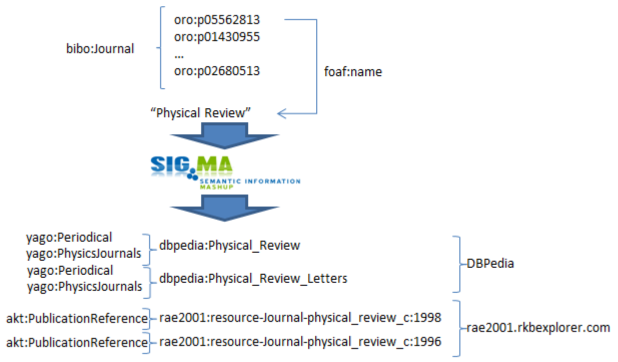

Figure 1: Keyword-based search for relevant individuals.

rdfs:label, *foaf:name*, *dc:title*, etc.) as keywords. Then, these query results are aggregated to estimate the degree of overlap of different data sources (Fig. 1). The procedure consists of the following steps:

1. Randomly selecting a subset of individuals from D_p belonging to a class c_p. This is done in order to reduce the number of queries to the search service in case where the complete extension set of individuals is too large. On the other hand, the subset must be large enough to produce reliable ranking of sources.

2. Querying the search service (Sig.ma) for labels of each individual in the selected subset. The results of each search are returned as an RDF document, which includes the references to individuals, their sources, and the classes they belong to.

3. Aggregation of the search results. RDF documents returned by Sig.ma are loaded into a common repository, and the individuals i_{jk} are grouped according to their sources D_j.

4. Data sources are ranked according to the number of their individuals returned by the search service $|\{i_{jk}|i_{jk} \in D_j\}|$.

In our approach we assume that the relevance function used by the search service to select query answers serves as an approximation of the identity function *equiv*(). In the general case, this is in not true due to ambiguity of labels and the fact that search services may not always achieve 100% precision. Taking a sufficiently large subset of individuals to search makes it possible to reduce the impact of "false positives" returned by the search engine.

After applying these steps to our test scenarios (see section 4), we found that the rankings obtained using this procedure are still likely to be imprecise for two main reasons:

- Inclusion of irrelevant sources. For individuals belonging to classes with highly ambiguous labels, many "false positives" in the set of answers can result in irrelevant repositories achieving high ranking positions. For instance, when searching for specific subcategories of people, any source mentioning sufficiently large number of people would be considered relevant: e.g., Twitter and DBLP were highly ranked when searching for music contributors.

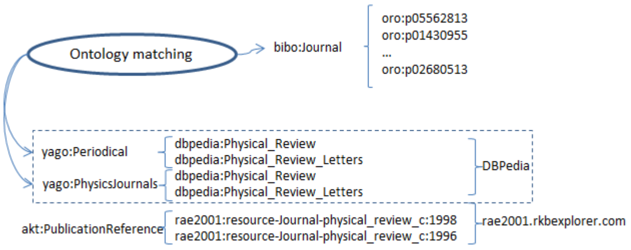

Figure 2: Using ontology matching to refine search results.

- Inclusion of irrelevant classes. Resulting sets often contained classes which would not allow selecting appropriate candidate individuals by a matching tool. Sometimes a generic superclass was ranked higher than the correct class: e.g., *dbpedia:Person* was ranked higher than a more relevant *dbpedia:MusicalArtist*. In other cases, completely irrelevant classes were included: e.g., for scientific journals the class *akt:Publication-Reference* describing specific volumes of journals was ranked higher than *akt:Journal*.

In order to overcome these issues, our approach includes the second stage: filtering of search results using ontology matching techniques.

3.2 Using ontology matching techniques to filter out irrelevant results

In order to filter out irrelevant search results, our approach can utilise mappings between classes provided by existing schema matching tools (Fig. 2). In our experiments we utilised ontology mappings produced by two algorithms:

- CIDER [1] which takes as input two ontologies in RDF format and two URIs defining ontological terms from these ontologies and produces as output the similarity score between these terms. CIDER utilises evidence defined at the level of ontological schema: string similarity between class labels, semantic relations defined in WordNet and positions of classes in class hierarchies.

- Instance-based matching algorithm described in [4], which generated schema mappings between classes on the Web of Data based on their overlapping sets of instances. Overlapping sets of instances were inferred based on existing *owl:sameAs* relations between them published in the Billion Triple Challenge 2009 (BTC) dataset[13]. Resulting mappings represent subsumption relations of the form $c_A \sqsubseteq c_B$, where c_A and c_B belong to different ontologies.

As the first step of the filtering procedure, CIDER is applied to measure similarity between the class c_p in D_p, for which overlapping sources have to be found, and each of the classes $c_{jk\lambda}$ appearing in the aggregated search results. Then, a threshold is applied to filter out classes with low similarity scores. Remaining classes from the search results constitute the set of "confirmed" classes $C_{confirmed}$. At the next stage, this set of "confirmed" classes is enriched using the mappings obtained using instance-based matching. For

[13] http://vmlion25.deri.ie/

each class $c_i \in C_{confirmed}$, all mappings from the BTC-based set where $c_A \sqsubseteq c_i$ are selected, and all c_A are added into $C_{confirmed}$. Then, the resulting set of search results is filtered so that only individuals belonging to "confirmed" classes remain. In our tests described in section 4, the filtering stage led to improved precision in the resulting ranking.

4. EXPERIMENTS

In our initial tests, we have applied the approach described in section 3 to the following datasets:

- ORO journals. A set of 3110 journals mentioned in the ORO repository constituting a part of *data.open.ac.uk*. Each individual belongs to the class *bibo:Journal*[14].
- LinkedMDB films. A subset of 400 randomly selected instances of the class *movie:film*[15] representing movies in the LinkedMDB repository.
- LinkedMDB music contributors. A subset of 400 randomly selected instances of the class *movie:music_contributor* representing music contributors for films in the LinkedMDB repository.

For each individual in these sets, we queried Sig.ma using their labels as keywords. First, we produced the ranking of sources using the whole set of search results returned by Sig.ma as described in section 3.1 and counted the number of actually relevant data sources among the top-10 ranked ones. Then, we applied the filtering mechanism using ontology schema matching results and checked the relevance of remaining sources. The results we obtained are presented in Table 1: for each dataset it shows the list of top ranked sources as well as our judgement whether these sources were actually relevant (column "+/-"). In the table, "(RKB)" denotes the datasets from RKBExplorer and "open EAN" corresponds to *openean.kaufkauf.net*. The "+/-" value denotes that the source could only be considered relevant in a specific context. In particular, the repositories listing film DVDs as trade commodities are relevant in the context of e-commerce, but not, e.g., as reference sources for students. For both LinkedMDB datasets, we did not consider the LinkedMDB repository itself when it was returned in the search results. As we can see from the results, the initial search-based ranking managed to discover relevant datasets for the sets of individuals in question. Top-ranked sources in the *Journals* and *Films* categories contained relevant individuals which could be linked to the individuals in D_p, and their sets of individuals are to a large degree overlapping. For music contributors, the proportion of irrelevant sources was substantially larger due to higher ambiguity of human names. The filtering stage in all cases resulted in improving the ranking precision: only relevant sources were confirmed. However, if we look at the ranking of ontological classes (Table 2), we can see that correctly identifying classes presents a number of issues. The table shows the highest ranking classes returned after each stage of the algorithm (only one highest-ranking class from each ontology is shown). Top-ranked classes produced from the search results usually represent high-level concepts and correspond to superclasses of the original class: e.g.,

Table 1: Test results: ranking of data sources

Dataset	Before filtering		After filtering	
	Top-ranked	+/-	Top-ranked	+/-
Journals	rae2001(RKB)	+	rae2001(RKB)	+
	dotac(RKB)	+	DBPedia	+
	DBPedia	+	dblp.l3s.de	+
	oai(RKB)	+	Freebase	+
	dblp.l3s.de	+	DBLP(RKB)	+
	wordnet(RKB)	-	eprints(RKB)	+
	www.bibsonomy.org	-		
	eprints(RKB)	+		
	Freebase	+		
	www.examiner.com	-		
Films	DBPedia	+	DBPedia	+
	open EAN	+/-	Freebase	+
	bestbuy.com	+/-		
	Freebase	+		
	www.answers.com	-		
	bitmunk.com	-		
	wordnet	-		
	www.examiner.com	-		
	it.bestshopping.com	+/-		
	www.songkick.com	-		
Musicians	DBPedia	+	Freebase	+
	www.realpageslive.com	-	DBPedia	+
	twitter.com	-		
	BBC	+		
	www.songkick.com	+/-		
	Freebase	-		
	Open EAN	+/-		
	LinkedIn	-		
	dblp.l3s.de	-		
	Yahoo!Movies	+		

Table 2: Test results: ranking of ontological classes.

Dataset	Before filtering	After filtering
	Top-ranked	Top-ranked
Journals	*akt:Publication-Reference*	*akt:Journal*
	dc:BibliographicResource	*yago:Periodical*
	foaf:Document	*swrc:Journal*
	swrc:Publication	*dbpedia:Work*
	vcard:VCard	*freebase:book.periodical*
	yago:Periodical	
	geo:SpatialThing	
	wn:Word	
	rss:item	
	swap:SocialEntity	
Films	*dbpedia:Work*	*dbpedia:Film*
	goodrelations: ProductOrServiceModel	*yago:Movie*
	yago:Movie	*freebase:film.film*
	icalendar:Vevent	
	foaf:Person	
	vcard:VCard	
	searchmonkey:Product	
	skos:Concept	
	geo:SpatialThing	
	freebase:common.topic	
Musicians	*vcard:VCard*	*freebase:film. music_contributor*
	geo:SpatialThing	*yago:American TelevisionComposers*
	swap:Person	
	foaf:Person	
	dc:Agent	
	mo:MusicArtist	
	icalendar:vcalendar	
	dbpedia:Person	
	goodrelations:ProductOrService	
	frbr:ResponsibleEntity	

[14] http://purl.org/ontology/bibo/Journal

[15] http://data.linkedmdb.org/movie/film

foaf:Document or *dc:BibliographicResource* for journals, *dbpedia:Work* for movies, and *foaf:Person* for musicians. Considering all instances of these classes as candidates for a data linking tool can lead to several problems. Matching algorithms usually implement pairwise comparison of individuals, so choosing all instances of a generic class as candidates for matching is likely to increase the computational time substantially. Also, less fine-grained feature selection is possible because important discriminating properties are often subclass-specific, and only properties common for all subclasses are defined for top-level concepts. This, in turn, can lead to lower quality of produced links, in particular, greater number of "false positives" [5]. Moreover, incorrect types were sometimes identified within relevant sources. For example, instances of *akt:Publication-Reference* cannot be linked to instances of *bibo:Journal* because they represent separate published volumes of a journal rather than the journal itself.

The filtering stage largely removed these problems so that only classes with a stronger degree of semantic similarity were confirmed. However, it also reduced the recall in cases where a directly corresponding class was not present in the external ontology: e.g., individuals from *dotac.rkbexplorer.com* and *oai.rkbexplorer.com*, which only used the generic class *dc:BibliographicResource* were not considered as relevant sources for linking journals. Similarly, many relevant classes were filtered out because they were not considered as exact matches or subclasses of the class *movie:music_contributor* (e.g., *mo:MusicArtist* and *dbpedia:MusicalArtist*).

5. DISCUSSION

Identifying relevant sources for interlinking already can present a non-trivial problem, and in future this issue is likely to become more important. The Linked Data cloud is constantly growing, and in order to make its use widespread, data owners must be able to publish their datasets without extensive knowledge about the state of the Web of Data or assistance from the research community. Interlinking is an important part of the publishing process and the one which can require substantial exploratory work with external data. Thus, this process has to become straightforward for data publishers and, preferably, require minimal human involvement. While the problem of link discovery has been addressed by several approaches (see, e.g., SILK [8] and *sameas.org*[16]), the problem of identifying relevant sources so far did not require such attention: published datasets were often interlinked with the help of researchers interested in the Linked Data initiative. A specific feature of this problem is the fact that the amount of necessary information about the Web of Data which is immediately available on the client (data publisher) side is limited, and gathering this information is a time-consuming process for the user. The proposed solution provides the data publisher with a ranked set of potentially relevant data sources and, in addition, a partial configuration of the data linking tool (classes containing relevant sets of instances). In this way, it can substantially reduce the need to perform exploratory search. Current version of the algorithm represents an initial solution, and we plan several directions for future work, among them:

- Integration into the generic data publishing workflow in order to provide a structured approach for data publishing within the organisation.
- Improvement of the search quality, in particular, the filtering stage. One particular route involves the analysis of possible choices of relevant sources and classes by estimating potential loss of precision and recall (e.g., see [3]).

Another potentially interesting research direction is related to the development of semantic indexes. Search for relevant data repositories can become a novel interesting use case in addition to the more common search for entities and documents. In order to support it, new types of search services can be valuable: for example, batch search for a large array of resource labels instead of multiple queries for small sets of keywords, which increase number of server requests and overall processing time.

6. ACKNOWLEDGEMENTS

This research has been partially funded under the EC 7th Framework Programme, in the context of the SmartProducts project (231204).

7. REFERENCES

[1] J. Gracia and E. Mena. Matching with CIDER: Evaluation report for the OAEI 2008. In *3rd Ontology Matching Workshop (OM'08) at the 7th International Semantic Web Conference (ISWC'08)*, Karlsruhe, Germany, 2008.

[2] H. Halpin, P. J. Hayes, J. P. McCusker, D. L. McGuinness, and H. S. Thompson. When owl:sameas isn't the same: An analysis of identity in linked data. In *9th International Semantic Web Conference (ISWC 2010)*, pages 305–320, Shanghai, China, 2010.

[3] E. Mena, A. Illarramendi, V. Kashyap, and A. P. Sheth. OBSERVER: An approach for query processing in global information systems based on interoperation across pre-existing ontologies. *Distributed and Parallel Databases*, 8(2):223–271, 2000.

[4] A. Nikolov and E. Motta. Capturing emerging relations between schema ontologies on the web of data. In *Workshop on Consuming Linked Data (COLD 2010), ISWC 2010*, Shanghai, China, 2010.

[5] A. Nikolov, V. Uren, E. Motta, and A. de Roeck. Integration of semantically annotated data by the KnoFuss architecture. In *16th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2008)*, pages 265–274, Acitrezza, Italy, 2008.

[6] G. Tummarello, R. Cyganiak, M. Catasta, S. Danielczyk, R. Delbru, and S. Decker. Sig.ma: Live views on the web of data. *Journal of Web Semantics*, 8(4):355–364, 2010.

[7] G. Tummarello, R. Delbru, and E. Oren. Sindice.com: Weaving the open linked data. In *6th International Semantic Web Conference (ISWC/ASWC 2007)*, pages 552–565, Busan, Korea, 2007.

[8] J. Volz, C. Bizer, M. Gaedke, and G. Kobilarov. Discovering and maintaining links on the web of data. In *8th International Semantic Web Conference (ISWC 2009)*, pages 650–665, Washington, DC, USA, 2009.

[16] http://www.sameas.org