Capturing Emerging Relations between Schema Ontologies on the Web of Data

Andriy Nikolov, Enrico Motta

Knowledge Media Institute, The Open University, Milton Keynes, UK
{a.nikolov, e.motta}@open.ac.uk

Abstract. Semantic heterogeneity caused by the use of different ontologies to describe the same topics represents an obstacle for many data integration tasks on the Web of Data, in particular, discovering relevant repositories for interlinking and comparing repositories with respect to the coverage of specific domains. To facilitate these tasks, mappings between schema terms are needed alongside the links between instances. Currently, explicitly specified schema-level mappings are scarce in comparison with instance-level links. However, by analysing existing instance-level links it is possible to capture correspondences between classes to which these instances belong. In our experiments, we applied this approach on a large scale to generate schema-level mappings between several Linked Data repositories. The results of these experiments provide some interesting insights about the use of ontologies on the Web of Data and schema-level relations which emerge from existing data-level interlinks.

1 Introduction

One of the main motivations behind large-scale data publishing using the Linked Data approach [1] is the possibility to integrate relevant information originally published by different providers. This is achieved, in particular, by establishing links between instances in different repositories. However, linking a new repository to other datasets in the cloud remains a non-trivial task for a data publisher. In order to tackle this task, several questions have to be answered, in particular:

– *Which other repositories contain relevant data?*
– *Which of these repositories should a new repository be connected to?* (or, alternatively, *URIs from which repositories should be reused in a new repository?*)

In order to answer the first question, one needs to know what types of individuals are stored in the datasets. With respect to the latter question, the choice of a candidate third-party repository for establishing links depends on several factors, in particular, their coverage (are all instances from the new repository mentioned in a candidate repository?), popularity (which one is the commonly accepted reference source for a specific type of data?), and level of detail (which repository describes the most properties for instances of a particular class?).

These questions can partially be answered with the help of meta-level descriptions using the voiD ontology[1]. However, voiD descriptors may be insufficient to compare some of the characteristics (e.g., whether the domain of food and diets is better covered in Freebase or DBPedia). Moreover, voiD descriptors not always describe all relevant properties of datasets (e.g., *dcterms:subject* is not always provided) and for some datasets may be not available.

One of the major obstacles which complicate this kind of analysis is schema heterogeneity. It can be difficult to establish automatically that two repositories describe the same kind of data, retrieve relevant data subsets from them, and make a comparison, if these repositories use different terminology to describe the same or semantically similar types of instances. For example, a hypothetical repository describing a TV program may need to refer to descriptions of movies, music pieces, and their performers. There are several repositories available on the Web: e.g., specific sources describing the music topic (MusicBrainz, Jamendo, etc.), the movie topic (LinkedMDB), as well as generic sources covering both (DBPedia, Freebase). In order to compare how well these repositories are suitable as reference sources, it is useful to know which classes in the respective ontologies contain overlapping data: e.g., *music:MusicArtist* and *dbpedia:MusicalArtist*, *linkedmdb:film* and *dbpedia:Movie*, etc. Having a high-level overview of schema-level correspondences, which would show the coverage of topics by available ontologies would help the data publisher to make appropriate choices.

In this paper, we described our work on constructing such a network of class level mappings for a subset of the Linked Data cloud. So far, several ontologies used by popular Linked Data repositories were enriched with mappings connecting them to other ontologies (most notably, in the context of the UMBEL project[2]). However, these mappings, constructed in a top-down way, only cover a limited subset of the Web of Data and do not fully reflect the structure of the repository network formed by instance-level links (e.g., such important repositories as Freebase, RKBExplorer, and LinkedMDB are not covered). Given the abundance of existing instance-level links, a bottom-up process where the correspondences between classes are captured based on the links between sets of their instances becomes a promising approach. We applied light-weight instance-based ontology matching techniques to a snapshot of the Web of Data which was proposed for the Billion Triple Challenge 2009 competition[3] and extracted a large-scale network of ontology mappings. This network provides interesting insights into the use of ontologies on the Web of Data and can be employed to facilitate data integration.

The rest of the paper is organised as follows: in section 2 we briefly outline the ontology matching process we used to extract the mappings and discuss our observations about its applicability and limitations. Then, in section 3 we describe the resulting network of schema mappings we obtained. In section 4 we

[1] http://semanticweb.org/wiki/VoiD
[2] http://www.umbel.org
[3] http://vmlion25.deri.ie/

overview relevant existing work. Finally, section 5 discusses the limitations of our work and directions for the future work.

2 Constructing the schema network

The snapshot of the Web of Data which we used in our work was proposed for the Billion Triple Challenge 2009 competition[4]. This is a large-scale dataset containing about 1.14 billion statements. It contains the core portion of the repositories published within the Linking Open Data (LOD) initiative, as well as many smaller datasets retrieved using Semantic Web search engines, such as Watson and Falcon-S. The LOD datasets included into the BTC repository such as DBPedia, Freebase, Bio2RDF, RKBExplorer, Geonames, and others still constitute the core of the Web of Data cloud and are commonly used to connect other datasets. Thus, their schema ontologies are particularly interesting for potential data integration scenarios.

To derive the sets of mappings between these ontologies, we applied a light-weight matching technique which computes the similarity between a pair of classes based on the degree of overlap between their instance sets. Originally, we used this approach to produce schema-level mappings in order to facilitate further instance coreference resolution and discover previously missing links [2]. An advantage of using instance-based ontology matching techniques in the Linked Data environment lies in their ability to capture interconnections between ontologies which emerged from the way they are used by actual repositories rather than the way they were originally designed.

When two classes share at least one individual, we say that there is an *overlap* relation between these classes. There are two common cases where an individual becomes assigned to several classes defined in different ontologies:

- *Declared coreference association.* In this case, two individuals belonging to different repositories are declared to be identical and linked via the *owl:sameAs* property. This creates an overlap relation between the classes to which the instances belong.
- *Co-typing.* In this case the publishers of a repository structure the data using terms of several ontologies. In this way, one individual can be explicitly assigned to several classes from different ontologies. One example is DBPedia, which uses Yago and UMBEL ontologies in addition to its native DBPedia ontology.

These two types of overlaps illustrate different aspects of the data structure. Declared association-based overlap relations characterise the distribution of data in different repositories and correspondences between sets of their individuals. Co-typing-based mappings mostly highlight the choices of data publishers to use specific vocabularies to annotate their data. To keep this distinction, in

[4] Dataset statistics can be found on http://vmlion25.deri.ie/ and http://gromgull.net/blog/category/semantic-web/billion-triple-challenge/.

this paper we analyse the *declared association-based* and *co-typing-based* overlap mappings separately.

In order to generate all overlap relations present in the dataset, we used the following procedure:

1. Extract all *rdf:type* relations present in the dataset: $A(I)$, where A is a class and I is an instance of this class.
2. For each class A, generate the set of its instances (extension): $e(A) = \{I|A(I)\}$.
3. For each pair of classes A and B, generate the co-typing-based overlap set: $e_c A \cap B = \{I|A(I), B(I)\}$. In total, this constituted about 3.6M co-typing-based overlap mappings (we only considered intersections between classes which did not share the same URI namespace)
4. Extract all *owl:sameAs* relations present in the dataset ($sameAs(I_1, I_2)$) and generate their transitive closure.
5. Generate association-based overlap sets: $e_a(A \cap B) = \{I_1|A(I_1), B(I_2), sameAs(I_1, I_2)\}$ (one *sameAs* relation corresponds to one element in the set). In total, about 1M (992482) association-based overlap mappings were produced.

For association-based overlap sets we distinguished between a direct class link (when their individuals were explicitly stated in the dataset as identical) and an indirect link (when *owl:sameAs* relations were inferred using transitivity). Indirect mappings occurred, in particular, when two repositories were connected via a third one (e.g., MusicBrainz and Freebase via DBPedia). Both sets of mappings were filtered to remove general-purpose concepts (such as OWL and RDFS terms) and blank nodes. These two sets of mappings constitute the "raw data" which were later analysed to retrieve valid semantic mappings.

In our original work [2], we used a set similarity-based metrics to discover relations between "strongly overlapping" classes in the ontologies. We used a fuzzy notion of "strong overlap" instead of strict subsumption or equivalence for two main reasons. First, in the Linked Data environment such mappings in many cases are impossible to derive: sometimes even strong semantic similarity between concepts does not imply strict equivalence. For instance, the concept *dbpedia:Actor* denotes professional actors (both cinema and stage), while the concept *movie:actor* in LinkedMDB refers to any person who played a role in a movie, including participants in documentaries, but excluding stage actors. Second, such "strong overlap" relations are valuable because they often point to semantically similar categories which to a large extent share the same instances. While not always strictly logically correct, these relations are still valuable for the goals we discussed in section 1: determining and comparing suitable sources for linking.

In order to capture the optimal parameters for distinguishing valid semantic mappings, in the experiments described in this paper we employed a machine learning approach. To construct a gold standard set, we have randomly selected a set of 6000 mappings (3000 association-based and 3000 co-typing-based ones) and annotated them manually ("strong overlap" relations were assigned based on subjective judgement). In these initial experiments, annotation was done by one person. After that, we used this gold standard set to train a classification model which would assign the relation type to any pair of overlapping classes. Our goal was to find a suitable classifier to distinguish between valid subsumption and equivalence mappings (*owl:equivalentClass* and *rdfs:subClassOf*) and other mappings.

For the classifier, we included the following features:

- *ns1*, *ns2*: namespaces of two class URIs A and B respectively.
- $|e(A \cap B)|$: the size of the set of instances belonging to both classes A and B.
- $|e(A)|$, $|e(B)|$: sizes of instance sets for classes A and B respectively.
- $\lambda(A, B)$, $\lambda(B, A)$, where $\lambda(X, Y) = \frac{|e(X \cap Y)|}{|e(X)|}$
- *direct* (only for declared association-based links): a boolean value equal to *true* for direct declared association-based mappings and *false* otherwise.

To test the resulting model, we used the standard 10-fold cross-validation mechanism. After testing, we found that the J48 decision tree algorithm was able to achieve the best performance (Table 1), so this learned classifier was then applied to the whole dataset.

Table 1. Test results: class matching

Mapping set	Test	Algorithm	Precision	Recall	F1
Association-based	1	J48	0.939	0.689	0.795
co-typing-based	2	J48	0.952	0.944	0.948

The resulting set of mappings was compared against the set of already existing schema-level relations declared in the dataset. We discovered that the majority of overlap mappings were not covered by explicitly defined axioms. Only 3119 mappings (2162 and 957 for the declared association-based and co-typing-based subsets respectively) were found to be defined as *rdfs:subClassOf* and *owl:equivalentClass* (or could be inferred), which constituted less than 2.6% and 1.4% of the number of mappings selected by the learned classifier in each case.

3 Analysing the resulting mappings

We applied the learned decision tree models (J48) to our two sets of mappings containing declared association-based and co-typing-based overlap mappings. At the next step, we filtered out redundant mappings: when a class A is found to be a subclass of two classes B and B_{super} where $B \sqsubseteq B_{super}$ and the distance metrics are equal ($\lambda(A, B) = \lambda(A, B_{super})$), then only the mapping $A \sqsubseteq B$ remains, and the mapping $A \sqsubseteq B_{super}$ is removed. Two resulting sets of mappings were then used to construct networks connecting classes from different ontologies. The characteristics of this network are discussed in section 3.1. Then, in order to study the relations between whole vocabularies, we used the original mappings between classes to generate a set of mapping-based links between ontologies. This stage is described in section 3.2.

3.1 Links between classes

We obtained two graphs where classes played the role of nodes and mappings represented edges. The properties of these resulting networks of classes are given in Table 2. To give an overview of the most important "hub" nodes in the network, Table 3 lists the top 10 classes ranked by the number of connections they are involved in.

Table 2. Networks of classes

Property	Declared association-based	Co-typing-based
Number of nodes	20365	35578
Number of edges	82422	67620
Maximum number of connections per node	5301	18137
Node with the maximum number of connections	geonames:Feature	foaf:Person
Average number of connections per node	8.09	3.80

We can see that the "hub" nodes represent classes representing popular concepts and defined at the high level in the class hierarchy. Large number of mappings per class is mostly caused by many *rdfs:subClassOf* relations. After analysing the distribution of mappings per class, we found that in both cases it follows the *power law* and most classes had only one mapping to another class.

The declared association-based network derived from *owl:sameAs* links between instances is more connected: average number of mappings per class is 8.09 compared to 3.8 in the co-typing-based case despite the fact that it contains less nodes. This is possibly caused by the "data-level focus" of the LOD initiative: the priority for a data repository owner is to generate instance-level links to other repositories rather than reuse several different vocabularies for data description. In this case, class-level mappings automatically derived from *owl:sameAs* links can be particularly helpful for data integration tasks, because they add new information which was not explicitly stated in any one repository. On the other hand, the co-typing-based network illustrates the impact of ontology popularity: although the graph has more nodes, it is less connected, and a single class *foaf:Person* contributes to more than 25% of all mappings. From the results we obtained, we can see the strong influence of DBPedia on the resulting mappings. In the association-based set, 7 out of the top 10 nodes relate to

Table 3. Top 10 classes (by number of edges)

Rank	Declared association-based		Co-typing-based	
Rank	Name	Edges	Name	Edges
1	geonames:Feature	5301	foaf:Person	18137
2	freebase:people.person	2318	umbel:Person	4533
3	yago:PhysicalEntity100001930	2230	dbpedia:Person	2478
4	yago:Object100002684	2076	foaf:OnlineAccount	1983
5	yago:Abstraction100002137	1759	dbpedia:FootballPlayer	1300
6	yago:Whole100003553	1511	wordnet:Person	1237
7	linkedmdb:film	1085	dbpedia:Album	996
8	yago:LivingThing100004258	975	dbpedia:Species	920
9	yago:Organism100004475	974	dbpedia:Artist	900
10	yago:CausalAgent100007347	956	dbpedia:MusicalArtist	853

top-level entities from the YAGO ontology. High positions of *geonames:Feature* and *freebase:people.person* are also largely due to the number of DBPedia and YAGO classes modelling the respective topics. In the co-typing-based network, we can see the strong presence of the FOAF and WordNet ontologies (largely due to their high reuse in small-scale datasets even before the start of the LOD initiative). Beyond that, all top nodes in the network were produced based on DBPedia instances annotated using different schemas. It is interesting to see the high position of the class *dbpedia:FootballPlayer*. The main reason for it is the large number of YAGO classes (Wikipedia categories) describing this topic.

When we merged two mapping sets into one, we found that only a small subset of mappings (3591) was shared between two networks. Two types of evidence we used produced complementary sets of mappings rather than duplicated each other.

3.2 Mapping-based links between ontologies

In order to capture the relations between different vocabularies used on the Web of Data, we generated a set of mapping-based links between ontologies. In accordance with [3], we say that there is a mapping-based link between two ontologies O_1 and O_2 if there exists a mapping between classes A and B such that $A \in O_1$ and $B \in O_2$. The classes were assigned to ontologies based on their URI prefixes, and mappings between classes from the same pair of ontologies were grouped together. Table 4 contains the details of the resulting graphs, and Table 5 lists for each case top 10 nodes sorted by the number of edges they are connected to.

The graphs constructed using declared association-based and co-typing-based evidence are shown in Fig. 1 and Fig. 2. In the declared association-based graph (Fig. 1), the main factor which influences the position of an ontology in the graph is *topic coverage*. The top 5 "hub" ontologies with wide coverage do not have a large difference in the number of connections: YAGO (29), Freebase (28),

Table 4. Networks of ontologies

Property	Declared association-based	Co-typing-based
Number of nodes	52	743
Number of edges	172	1352
Maximum number of connections per node	29	504
Node with the maximum number of connections	YAGO	FOAF
Average number of connections per node	3.96	1.85
Connected components	5	35
Average path length	2.92	2.48

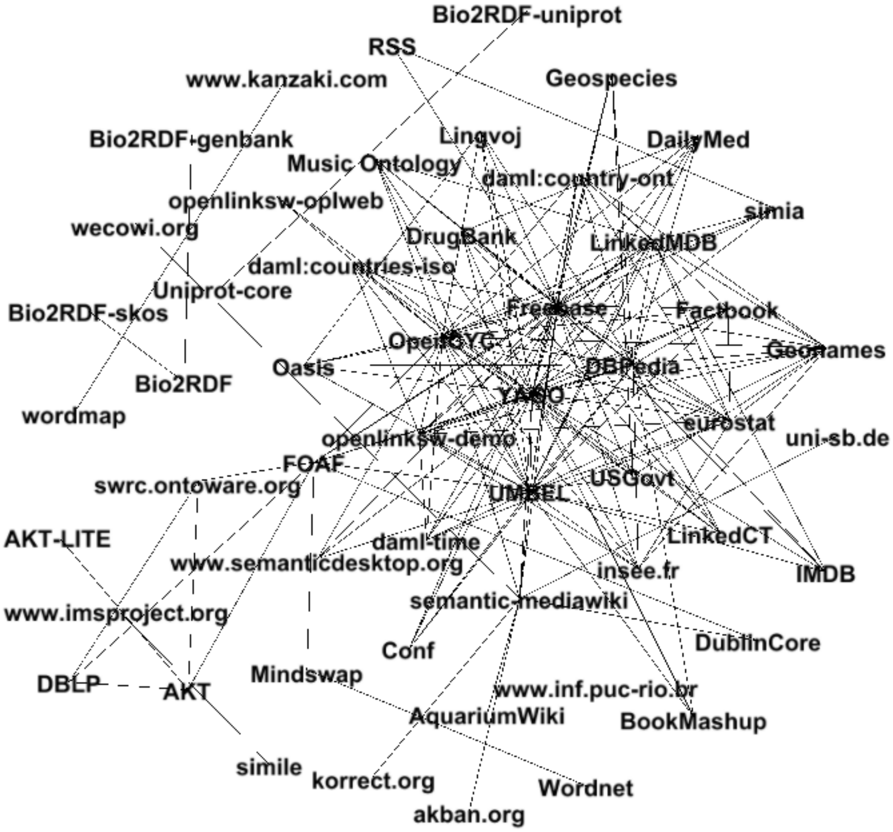

Fig. 1. The network of ontologies derived from instance coreference links. Ontologies with wide coverage used by popular repositories serve as "hubs": YAGO, DBPedia, OpenCYC, Freebase, and UMBEL.

UMBEL(27), OpenCYC (26), and DBPedia (23). The 6th and the 7th ranking nodes (LinkedMDB and eurostat), which cover specific domains, have only 13 connections each. It is interesting to note that although Freebase is connected to less repositories than DBPedia in the LOD cloud[5], this does not have an impact at the schema level. This is the effect of indirect *owl:sameAs* mappings inferred by transitivity. Connections of domain-specific ontologies (such as Music ontology or Geonames) point to other ontologies covering the same domain, and, indirectly, to the underlying repositories which contain relevant data. This makes them good starting points when the task is to find several datasets relevant to a specific topic. Both networks contain several disjoint subgraphs (5 and 35 respectively), and in both cases the same pattern occurs: there exists one large central cluster including the majority of nodes and several small ones usually including a pair of ontologies (e.g., a cluster {http://purl.uniprot.org/core/, http://bio2rdf.org/ns/uniprot#}). In Fig. 1, similarly to the data-level LOD cloud, we can also observe the existence of two "communities" centered around DBPedia and RKBExplorer. At the schema level these are centered around YAGO and AKT ontologies. Both communities are connected via the FOAF ontology (*rdfs:subClassOf* relations with the *foaf:Person* class). At the data level, RKBExplorer and DBPedia are connected via two other repositories: DBLP Hannover and DBLP Berlin. The reason for missing schema-level links between AKT and the ontologies used in DBPedia was the omission of intermediate *owl:sameAs* links on this route, which did not allow indirect declared association-based class mappings to be produced.

Table 5. Top 10 ontologies (by number of edges)

Rank	Declared association-based		Co-typing-based	
	Name	Edges	Name	Edges
1	YAGO	29	FOAF	504
2	Freebase	28	Wordnet	296
3	UMBEL	26	AKT	66
4	OpenCYC	25	Music ontology	52
5	DBPedia	23	semantic-mediawiki	37
6	eurostat (VU Berlin)	13	RSS	30
7	LinkedMDB	13	eurostat	30
8	Geonames	12	DAML-OIL	29
9	openlinksw-demo	12	geneontology	26
10	FOAF	11	Mindswap	25

The co-typing-based network (Fig. 2) is substantially larger (746 nodes vs 53) and mainly connects ontologies used outside the LOD cloud (including even legacy schemas like DAML-OIL). In this graph, the distribution of nodes primar-

[5] http://richard.cyganiak.de/2007/10/lod/

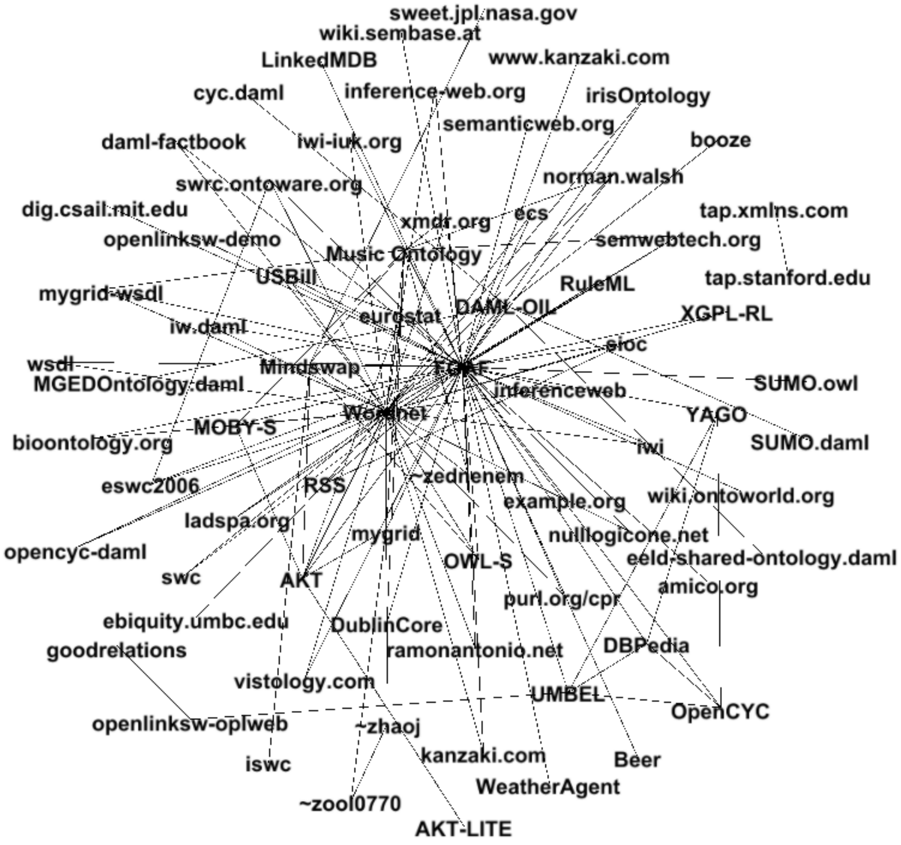

Fig. 2. The network of ontologies derived from ontology reuse (only ontologies with at least 10 populated classes are shown). FOAF and Wordnet, reused by many datasets, have the most connections.

ily illustrates *ontology popularity*: FOAF (504 connections) and Wordnet (296) get the most connections because they are reused in many datasets.

4 Related Work

Originally, schema matching approaches in the database and Semantic Web domains primarily focused on the task of matching two input schemas in isolation from others [4], [5]. With the availability of public ontologies, schema matching methods started to utilise external sources as background knowledge. One approach proposed in [6] matches two ontologies by linking them to an external third one. Then, semantic relations defined in this external ontology are used to infer mappings between entities of two original ones. The SCARLET tool [7] further elaborates this approach and employs a set of external ontologies, which it searches and selects using the Watson ontology search server[6].

Recently, with the growing number of public repositories storing data about overlapping domains, it became important to analyse the emerging network of interconnections as a whole. The *idMesh* system[8] analysed the network of instance-level *owl:sameAs* coreference links between semantic repositories with the aim to identify spurious links and remove them. In [3] the authors used light-weight matching techniques to create a large set of schema-level mappings between ontologies from the BioPortal repository describing the medical domain. Then, the authors analysed the resulting network to gain insights about ontological coverage of the domain. We take a similar approach, however, our primary interest is in schema mappings which emerge from existing data-level links between repositories.

5 Conclusion and future work

As mentioned in section 1, schema-level mappings can become a valuable asset for the data publisher who wants to integrate a new repository into the Linked Data environment: for example, having a new repository about music described using the Music ontology, the pool of potential data sources to connect to would include other datasets using the the same ontology, but also repositories which use ontologies mapped to the it (DBPedia, Freebase, LinkedMDB, etc.). From this pool the publisher can select the most comprehensive data source for her needs.

We consider the work described in this paper as our starting point in studying the emerging relations between ontologies on the Web of Data. There are several interesting future directions of research. First, our approach focused on establishing mappings between classes while ignoring mappings between properties, which are equally important in data integration scenarios. Mappings between properties are needed to represent data from different ontologies in a uniform

[6] http://watson.kmi.open.ac.uk/WatsonWUI/

way, which is necessary for applying coreference resolution tools or, in a more general scenario, to present query results to the user.

Second, in the context of our intended scenario (assisting the publisher in the choice of appropriate points of linkage) the quality of mappings had relatively low importance: a mapping is still useful if it connects two classes with a strong degree of overlap, but no strict logical relation holds. This allowed us to use very simple matching techniques to generate schema-level mappings. However, this assumption does not hold for many actual data integration scenarios: in general, a precise SPARQL query is not expected to return irrelevant results. Thus, applying state-of-the-art ontology matching tools to discover high-quality schema mappings in the Linked Data environment constitutes the second direction for future work.

6 Acknowledgements

Part of this research has been funded under the EC 7th Framework Programme, in the context of the SmartProducts project (231204). The authors would like to thank Paul Groth and Cristophe Gueret for providing the 4store Amazon EC2 community server hosting the BTC dataset.

References

1. 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
2. Nikolov, A., Uren, V., Motta, E., de Roeck, A.: Overcoming schema heterogeneity between linked semantic repositories to improve coreference resolution. In: 4th Asian Semantic Web Conference (ASWC 2009), Shanghai, China (2009) 332–346
3. Ghazvinian, A., Noy, N.F., Jonquet, C., Shah, N., Musen, M.A.: What four million mappings can tell you about two hundred ontologies. In: 8th International Semantic Web Conference (ISWC 2009), Washington DC, USA (2009) 229–242
4. Rahm, E., Do, H.H.: Data cleaning: Problems and current approaches. IEEE Bulletin of the Technical Committee on Data Engineering **23**(4) (2000)
5. Euzenat, J., Shvaiko, P.: Ontology matching. Springer-Verlag, Heidelberg (2007)
6. Aleksovski, Z., Klein, M.C.A., ten Kate, W., van Harmelen, F.: Matching unstructured vocabularies using a background ontology. In: 15th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2006). (2006) 182–197
7. Sabou, M., d'Aquin, M., Motta, E.: Exploring the Semantic Web as background knowledge for ontology matching. Journal of Data Semantics **XI** (2008) 156–190
8. Cudré-Mauroux, P., Haghani, P., Jost, M., Aberer, K., de Meer, H.: idMesh: Graph-based disambiguation of linked data. In: 18th International World Wide Web Conference (WWW 2009), Madrid, Spain, ACM (2009) 591–600