Ontology Alignment for Linked Open Data

Prateek Jain[1], Pascal Hitzler[1], Amit P. Sheth[1], Kunal Verma[2], and Peter Z. Yeh[2]

[1] Kno.e.sis Center, Wright State University, Dayton, OH
[2] Accenture Technology Labs, San Jose, CA

Abstract. The Web of Data currently coming into existence through the Linked Open Data (LOD) effort is a major milestone in realizing the Semantic Web vision. However, the development of applications based on LOD faces difficulties due to the fact that the different LOD datasets are rather loosely connected pieces of information. In particular, links between LOD datasets are almost exclusively on the level of instances, and schema-level information is being ignored. In this paper, we therefore present a system for finding schema-level links between LOD datasets in the sense of ontology alignment. Our system, called BLOOMS, is based on the idea of bootstrapping information already present on the LOD cloud. We also present a comprehensive evaluation which shows that BLOOMS outperforms state-of-the-art ontology alignment systems on LOD datasets. At the same time, BLOOMS is also competitive compared with these other systems on the Ontology Evaluation Alignment Initiative Benchmark datasets.

1 Introduction

The Linked Open Data (LOD) community effort is a cornerstone in the realization of the Semantic Web vision [1]. So far it has resulted in an openly available ”Web of Data” comprising several billion RDF triples. LOD captures knowledge from diverse domains and is constantly growing. Some of the domains include: information from Wikipedia, governmental and geospatial data, entertainment, bio-informatics and publications. However, in terms of practical usability, LOD is still in its infancy. Several central issues remain to be investigated and solved, and discussions of these are ongoing among researchers (see, e.g., [2,3,4]). Our own preliminary investigations into LOD querying [2] in particular exposed a need for schema-level integration of LOD datasets, an issue which has also been pointed out in [1], and elsewhere, as a core challenge.

While LOD datasets are well interlinked on the instance level, they are very loosely connected on the schema level (see also Table 3). Since our work involves schema alignment for our work, we investigated the most competitive state-of-the-art ontology alignment systems available in order to use them for the integration task. However, it turned out that the performance of these systems on LOD schema datasets was rather poor, even though they performed fine on established benchmarks. We were thus left with finding our own solution to LOD schema alignment, on which we report here. Our resulting system, BLOOMS, in fact outperforms state-of-the-art ontology alignment systems in LOD schema alignment, while is roughly on par with these systems on established ontology alignment benchmarks (see Section 4).

P.F. Patel-Schneider et al. (Eds.): ISWC 2010, Part I, LNCS 6496, pp. 402–417, 2010.

Conceptually a key strength of BLOOMS is that we utilize a bootstrapping approach (see Section 3). The system computes alignments with the help of noisy community-generated data available on the Web. Currently, BLOOMS uses Wikipedia and the Wikipedia category hierarchy for this purpose. However there is no conceptual reason why one would not be able to use other inputs (or even existing upper-level ontologies or upper-level domain-specific ontologies) instead. This would simply result in a different bias for the alignment, which could potentially be exploited, e.g., for alignment tasks on narrower thematic domains (see also our discussion of future work, Section 6). Furthermore BLOOMS utilizes the Alignment API [5] as a base system by exploiting its capabilities which complement the native BLOOMS bootstrapping approach.

The structure of the paper is as follows. In Section 2, we clarify some notions and explain the background. In Section 3, we give details about our bootstrapping approach. In Section 4, we give a detailed quantitative evaluation of BLOOMS by comparing it with state-of-the-art ontology alignment systems, for LOD schema alignment and for the Ontology Alignment Evaluation Initiative benchmark. In Section 5, we discuss related work. In Section 6, we conclude with a summary and ideas for future work.

Further details on the evaluation, and the BLOOMS system for download, can be found at `http://wiki.knoesis.org/index.php/BLOOMS`.

2 Preliminaries

An overview of LOD appears in [1]. Although different LOD datasets are interlinked, it should be noted that interlinks are still relatively scarce. Interlinks are mainly on the instance level (using owl:sameAs), and are clustered within three major thematic domains which are hardly connected by links—see [6]. Schema-level information, by which we mean taxonomies built using rdfs:subClassOf (possibly enriched with further RDF Schema or OWL axioms not involving instance data), is also relatively scarce. In particular there is a lack of interlinks between the different schemas.

DBpedia [7] is an LOD dataset which is based on Wikipedia infoboxes. Our bootstrapping approach employs noisy community-generated data, and we have chosen to use Wikipedia and DBPedia. A central role is played by the Wikipedia category hierarchy, which is a user-generated class hierarchy for Wikipedia pages. It is important to notice that this category hierarchy is not a taxonomy in any reasonable sense. In particular, many of the ”sub-category” relations are semantically not rdfs:subClassOf relations [8]. We will discuss our reasons for choosing Wikipedia/DBpedia later on in Section 3.

BLOOMS is a system for schema alignment. For the purpose of this paper, we mean by schema alignment the generation of links between class hierarchies (taxonomies), which are rdfs:subClassOf relations. For an example, if ”Human” occurs in some dataset and ”Woman” occurs in some other dataset, then we would expect BLOOMS (or any other ontology alignment system) to create a relation between these two classes in the form of an RDF triple ”Woman rdfs:subClassOf Human”. Note that two classes A and B will always be related by one out of four relationships: A rdfs:subClassOf B, B rdfs:subClassOf A, A owl:equivalentClass[1] B, or none of the previous three.

[1] This is semantically equivalent to stating *both* A rdfs:subClassOf B and B rdfs:subClassOf A, and we abstract from the (syntactic) difference.

3 The BLOOMS Approach

At the core of the BLOOMS bootstrapping approach is the utilization of the Wikipedia category hierarchy. In essence, BLOOMS constructs a forest (i.e., a set of trees) T_C (which we call the *BLOOMS forest* for C) for each matching candidate class name C, which roughly corresponds to a selection of supercategories of the class name. Comparison of the forests T_C and T_B for matching candidate classes C and B then yields a decision whether or not (and with which of the candidate relations) C and B should be aligned. We next spell this out in detail.

BLOOMS accepts as input two ontologies which are assumed to contain schema information. It then proceeds with the following steps.

1. **Pre-processing of the input ontologies** in order to (i) remove property restrictions, individuals, and properties, and to (ii) tokenize composite class names to obtain a list of all simple words contained within them, with stop words removed.
2. **Construction of the BLOOMS forest** T_C for each class name C, using information from Wikipedia.
3. **Comparison of constructed BLOOMS forests**, which yields decisions which class names are to be aligned.
4. **Post-processing** of the results with the help of the Alignment API and a reasoner.

We now give more details and examples on the key steps just described. As a running example, we use the class names Event and JazzFestival taken from the LOD datasets DBpedia and Music Ontology, respectively.

Pre-processing of the input ontologies. This involves a straightforward algorithm which normalizes each input class name C into a string C' obtained by replacing underscores and hyphens[2] by spaces, splitting at capital letters, and the like.[3] For stop word removal we used the 319 stop words defined by the Information Retrieval Research Group of Glasgow University.[4]

For our running example, JazzFestival is transformed to ”Jazz Festival”, whereas Event is not modified at all.

Construction of the BLOOMS forest T_C **from** C. The first step in constructing T_C is to invoke a call to the Wikipedia Web service using C' as input. This Web service returns a set of Wikipedia pages[5] W_C as results of a search on Wikipedia for the words in the string. If a returned result is a Wikipedia disambiguation page, it is then removed from W_C and replaced by all Wikipedia pages mentioned in the disambiguation page. We call the elements of the resulting set W_C *senses* for C.

Concerning our running example, for Event, the Web service returns Event, Eventing, Sport, NFL Draft, News, Festival, Event-driven programming, Rodeo, Athletics at the Summer Olympics, and Extinction event.

[2] We actually did the hyphens manually, because they occurred only in one of our test ontologies, namely the AKT Portal Ontology (see Section 4).

[3] There was no need to make use of a dictionary, mainly because the resulting strings are used as input to Wikipedia search, which works well without stemming etc.

[4] `http://ir.dcs.gla.ac.uk/resources/linguistic_utils/stop_words`

[5] More precisely, their URLs.

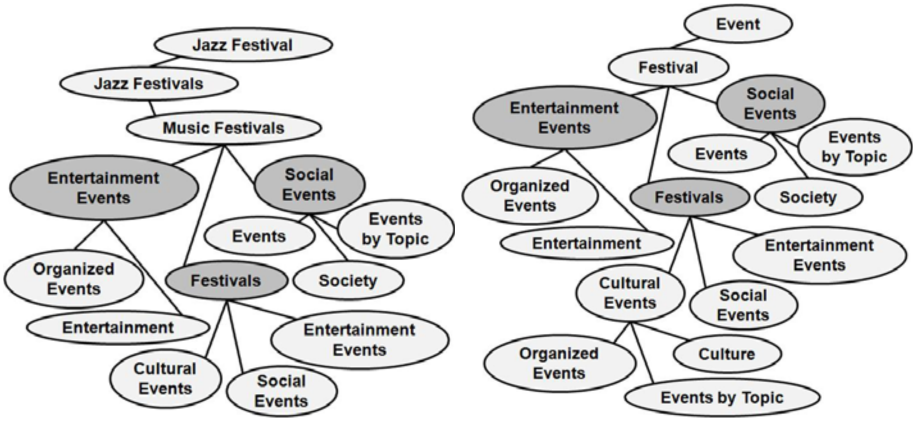

Fig. 1. BLOOMS trees for Jazz Festival with sense Jazz Festival and for Event with sense Event. To save space, some categories are not expanded to level 4.

In the next step, for each sense $s \in W_C$, a tree $T_s \in T_C$, called the *BLOOMS tree for C with sense* s, is constructed, as follows.

- The root of the tree is s.
- Children of s are exactly all the Wikipedia categories into which the Wikipedia page s is categorized.
- Subsequently, for each category c which is a node in the tree, its children are exactly all Wikipedia categories of which c is a subcategory.
- T_s is the resulting tree, which is cut at level 4 (i.e., branches of T_s have maximally 5 nodes, including the root).

We decided to cut the tree at level 4 because we found that deeper levels involve Wikipedia categories which are very general, like ”Humanities”, and thus would be ineffective for our purposes.

Figure 1 shows the BLOOMS tree for Event with sense Event and for Jazz Festival with sense Jazz Festival.

Comparison of constructed BLOOMS forests. Any concept name C in the one input ontology is now matched against any concept name D in the other input ontology. This is done by comparing each $T_s \in T_C$ with each $T_t \in T_D$. For this, we define a function o, which assigns a real number in the unit interval to each (ordered) pair of BLOOMS trees. The value $o(T_s, T_t)$, called the *overlap of* T_t *with* T_s, is defined as follows.

1. Remove from T_s all nodes for which there is a parent node which occurs in T_t. All leaves of the resulting tree T_s' are either of level 4 or occur in T_t. Note that due to the way BLOOMS trees are constructed, we removed only nodes from T_s which actually occur in T_t—we remove them because they do not give us any essential additional information for comparing T_s with T_t.
2. $o(T_s, T_t) = \frac{n}{k-1}$, where n is the number of nodes in T_s' which occur also in T_t, and k is the total number of nodes in T_s' (we do not count the root).

In our running example, the BLOOMS trees in Figure 1 are pruned beneath the dark gray nodes. We obtain $o(T_{\text{Event}}, T_{\text{Jazz Festival}}) = \frac{3}{4}$ and $o(T_{\text{Jazz Festival}}, T_{\text{Event}}) = \frac{3}{5}$.

The decision on an alignment is then made as follows.

- If, for any choice of $T_s \in T_C$ and $T_t \in T_D$, we have that $T_s = T_t$, then we set C owl:equivalentClass D.
- If $\min\{o(T_s, T_t), o(T_t, T_s)\} \geq x$ for any choice of $T_s \in T_C$ and $T_t \in T_D$, and for some pre-defined threshold x,[6] then set C rdfs:subClassOf D if $o(T_s, T_t) \leq o(T_t, T_s)$, and set D rdfs:subClassOf C if $o(T_s, T_t) \geq o(T_t, T_s)$.

For our running example, we have $o(T_{\text{Event}}, T_{\text{Jazz Festival}}) > o(T_{\text{Jazz Festival}}, T_{\text{Event}})$, and therefore obtain Jazz Festival rdfs:subClassOf Event.

Post-processing. For post-processing, we first invoke the Alignment API for finding alignments between the original input ontologies. Those alignments returned with a confidence value of at least 0.95 are kept, and added to the results previously obtained.[7] We then invoke a reasoner (in fact, Jena) which finds inferred alignments. E.g., if A is a subclass of B in one of the input ontologies, and an alignment B rdfs:subClassOf C has already been found, then the alignment A rdfs:subClassOf C is also added, and finding these alignments is done using a reasoner. We finally output the alignment results, in the Alignment API format.

The BLOOMS approach as just described makes heavy use of Wikipedia/DBPedia for bootstrapping. It is natural to ask, if Wikipedia could be replaced with something else. In general, we think so. In fact, any upper level ontology or thesaurus could be used, and perhaps there are even more options we did not think of. BLOOMS currently uses Wikipedia because it seemed an intuitive choice due to a number of reasons.

- Wikipedia provides wide thematic coverage.
- The Wikipedia category hierarchy is community-built and thus seemed a natural choice for an alignment system for community-built LOD datasets.
- Wikipedia provides a search feature which we could exploit. This search feature makes it possible to naturally include trees in BLOOMS forests which would be difficult to associate with the input concept name in a more controled setting, e.g., when using an upper level ontology.

We have not systematically investigated any alternatives yet. The evaluation in Section 4 shows that the current approach using Wikipedia is already rather strong. It is left for future work to investigate to what extent alternatives would bring an increase in performance. We hypothesize that alternatives should indeed be very helpful for alignment in more specialized thematic domains, e.g., for life science data in the LOD Cloud. Potential alternatives include the following: Ontologies such as Cyc or SUMO, as used, e.g., in [9]; Thesauri such as WordNet;[8] Taxonomies created from Wikipedia, such as the one reported in [8]; or efforts like the Open Directory Project[9] or YAGO [10].

[6] This threshold was typically 0.8 or 0.6 in our experiments in Section 4, where we will discuss how to set suitable thresholds.

[7] 0.95 seems to be the lowest threshold generally giving indisputable results.

[8] WordNet is used by the Alignment API [5], and thus is indirectly utilized by our approach.

[9] `http://www.dmoz.org/`

4 Evaluation

We have implemented our approach in the BLOOMS[10] system in Java on top of the Alignment API framework [11]. We utilize the Jena Framework[11] for parsing the ontologies, extracting the concepts and for the mentioned reasoning step. The input for BLOOMS is two different ontologies serialized using RDF/XML or OWL.

We performed a comprehensive evaluation of BLOOMS using third party datasets and other state-of-the-art systems in ontology matching. More specifically, we evaluated BLOOMS in two different ways. Firstly, we examined the ability of BLOOMS to serve as a general purpose ontology matching system, by comparing it with other systems on the Ontology Alignment Evaluation Initiative (OAEI) benchmarks.[12] Secondly, we evaluated BLOOMS for the purpose of LOD schema integration and compared it with other systems for ontology matching on LOD schema alignment.

Established in 2004 by leading researchers in the area of ontology matching, the OAEI aims at forging consensus on methods available for schema matching/ontology integration. As a part of this initiative various datasets and reference alignments between these datasets have been made available for evaluating the performance of the participating systems. The systems are evaluated on various parameters such as precision, recall, endurance to lack of structure in the ontologies and absence of properly named concepts.

The initiative consists of various tracks such as a benchmark track, instance matching and oriented matching. The datasets mainly belong to the very narrow domain of bibliographic information with a number of alternative ontologies of the same domain for which alignments are provided. We decided to evaluate BLOOMS on both the benchmark track and the oriented matching track. In the former the task is to identify (only) equivalence relationships. In the latter the task is to identify subclass relationships. The objective of the BLOOMS system is naturally aligned with these two tracks. Furthermore, the OAEI provides us with baselines, and results from the previous version of the oriented matching track are available on the web.[13]

In the 2009 initiative, there were five major systems in the oriented matching track: ASMOV [12], CSR [13], RiMOM [14], AROMA [15] and TaxoMAP [16]. We picked RiMOM and AROMA, for the following reasons: (1) RiMOM was the top system in the oriented track in terms of f-measure and available for download. It was one of the consistent performers in the past two years. (2) AROMA ranked second in the 2008 event. (3) Another important factor was the availability of systems for download in order to run experiments on LOD datasets using them.[14] (4) RiMOM and AROMA

[10] BLOOMS is available from `http://wiki.knoesis.org/index.php/BLOOMS`

[11] `http://www.openjena.org/`

[12] `http://oaei.ontologymatching.org/`

[13] `http://oaei.ontologymatching.org/2009/results/oriented/`

[14] In the OAEI 2009 initiative there were other systems which performed better than RiMOM, namely ASMOV, Lily and CSR. However, ASMOV is a commercial system and the free version runs only on OAEI 2009 datasets and therefore we cannot use it on LOD datasets. CSR is not available for download and our requests for an evaluation copy remained unanswered. TaxoMAP and Lily we could not get working due to platform incompatibility issues, and our support requests were not answered in time.

utilize different techniques and hence this gives good variety in the techniques utilized for the purpose of matching. RiMOM, in fact, automatically determines which ontology alignment methods to use for a particular matching task, and what kinds of information to use in the similarity calculation and how to combine multiple methods as necessary. AROMA is an ontology matcher which utilizes association rule mining.

In order to achieve more breadth in our evaluation, we also included recent systems which have not participated in the OAEI. OMViaUO [9] utilizes upper level ontologies such as SUMO and DOLCE as semantic bridges in the ontology matching process. S-Match [17] is another novel approach in which semantic correspondences are discovered by computing and returning, (as a result) the semantic information implicitly or explicitly codified in the labels of nodes and arcs.

Some of the systems had tunable parameters. As mentioned in Section 3, we used BLOOMS with a threshold value of 0.8 for the ontologies belonging to the same domain, and used a value of 0.6 where one of the ontologies was an abstract ontology such as DBpedia or SUMO. This was done for the following reasons: (1) We expect BLOOMS trees for concepts belonging to the same domain to have higher overlap. (2) Relations between an abstract and a domain specific ontology can be found using a lower overlap. This is because BLOOMS trees constructed for concepts in the domain specific ontology will usually require more nodes to become generic enough in order to match a concept of the more generic ontology.

For RiMOM, while evaluating on LOD datasets, based on our understanding we specified a number of thresholds in the ”MatchThreshold” parameter, which range from 0.3 to 0.8. However, the execution with the different parameters always resulted in the same output. On inspection of the results, we found that there were entries with threshold values as low as 0.01 in the output file.

For AROMA, we utilized a threshold of 0.6 for ”lexicalThreshold”. While parameters below 0.5 were too low and resulted in very poor precision, higher thresholds such as 0.8 resulted in identification of very few results. If guidelines were available for deciding the thresholds, we might have been able to tune the system in a better way.

We could not tune S-Match since the S-Match GUI does not provide for this.

We consulted the OMViaUO literature to get information related to setting suitable thresholds. However, we found no discussion related to this. Further, with respect to the Alignment API and OMViaUO, altering the threshold values (even to 0) did not result in any significant improvement of results on LOD datasets. For the Alignment API and OMViaUO we kept the threshold at 0.5 to achieve an optimum balance between precision and recall.

4.1 Evaluation: Ontology Alignment Evaluation Initiative Oriented Track

In order to test the quality of mappings generated using BLOOMS, we ran our system on the oriented datasets using the reference alignment and compared its performance with the other systems mentioned above. Table 1 presents our results on the oriented matching track of the OAEI. The different tests 1XX, 2XX, and 3XX comprise of matching a single source ontology (101) to other ontologies beginning with the prefix digit of the test. Thus, test 1XX comprises of matching ontology 101 to ontologies 101, 103, and so forth. Similarly 2XX comprises of matching ontology 101 to ontologies 201, 202, and

Table 1. Results on the oriented matching track. Results for RiMOM and AROMA have been taken from the OAEI 2009 website. Legends: Prec=Precision, Rec=Recall, A-API=Alignment API, OMV=OMViaUO, NaN=division by zero, likely due to empty alignment.

Ontology Alignment Initiative—Oriented Matching Track												
	A-API		OMV		S-Match		AROMA		RiMoM		BLOOMS	
Test	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec
1XX	0	0	0.02	0.06	0.01	0.71	NaN	0	1	1	1	1
2XX	0	0	0.01	0.03	0.05	0.30	0.84	0.08	0.67	0.85	0.52	0.51
3XX	0.01	0.03	0.02	0.047	0.01	0.14	0.72	0.11	0.59	0.81	1	0.84
Avg.	0.00	0.01	0.02	0.04	0.03	0.38	0.63	0.07	0.75	0.88	0.84	0.78

so forth. Unlike the ontologies used in the tests 1XX and 2XX which are created by the organizers, the test 3XX comprises of ontologies which have been created by other organizations and are used in the real world. We computed the precision and recall figures using the baselines and results made available on the OAEI website.

In the oriented matching track, BLOOMS along with RiMOM provided superior results in the test 1XX. For the test 2XX, all systems including BLOOMS show a drop in the performance. We fathom that the reasons for this drop are the following. (1) Some ontologies in test 2XX contain concepts from French. Thus systems which rely on lexico-syntactic tools obviously have difficulties with these ontologies.[15] (2) Some of these ontologies consist of concepts with random names where the matching has to be done on the basis of structure alone.

For the test 3XX, BLOOMS outperforms the other systems in its recall without comprising on its precision. The reasons for the superior performance of BLOOMS could be the following: (1) Wikipedia has a large number of articles with a rich category hierarchy in which the articles and categories summarize the concepts mentioned in the real world ontologies. (2) The ontologies in these tests are of related domains (e.g. Scientific Publishing) and therefore, require a higher overlap between the BLOOMS trees for two concepts to be related. A higher overlap threshold enforces that the concepts and their corresponding BLOOMS trees have to be very similar. This reduces the number of false positives. (3) The mentioned invocation of a reasoner allows us to identify some of the concepts which otherwise have to be found using the structure of the ontology.

The other systems (besides RiMOM) suffer from poor precision and recall due to a variety of reasons. (1) A number of systems such as OMViaUO generate only equivalence mappings. In the oriented matching track, the provided reference alignments consist mainly of subsumption relationships. (2) While S-Match provides good results for the recall, its precision is affected by a plethora of results which are generated for the ontologies. S-Match produces two different output files. We utilized the ”default results” file, since it gives a larger number of results. The other file ”minimal results” produces a very small set of results, which one could expect to have a higher precision but lower recall, but this is not necessarily the case. For example, for matching

[15] In future investigations, one could attempt to exploit the fact that Wikipedia is available in many languages, and that the different-language versions are in fact interlinked.

Table 2. Comparison of various systems on the benchmark track. Results for RiMOM and AROMA have been reused from the OAEI 2009 website. Legends: Prec=Precision, Rec=Recall.

Ontology Alignment Initiative—Benchmark Track												
	S-Match		OMViaUO		Alignment API		BLOOMS		AROMA		RiMoM	
Test	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Recall	Prec	Rec
1XX	0.11	1	0.26	0.37	0.59	0.96	0.71	1	1	1	1	1
2XX	0.1	0.2	0.21	0.31	0.3	0.54	0.38	0.49	0.88	0.65	0.93	0.81
3XX	0.1	0.2	0.28	0.28	0.45	0.77	0.62	0.84	0.80	0.76	0.81	0.82
Avg.	0.1	0.46	0.25	0.33	0.45	0.76	0.57	0.78	0.88	0.81	0.91	0.88

ontologies 101 and 103, S-Match produced 267 results in the default file (precision: 0.46; recall: 0.50), and 57 in the minimal file (precision: 0; recall: 0). (3) OMViaUO could not produce satisfactory results due to poor matching performance. We believe the reason for this is the absence of required ontological concepts in WordNet and in the upper level ontologies utilized by OMViaUO. (4) The Alignment API also suffered from poor precision and recall due to reasons similar to those for OMViaUO. (5) We think AROMA suffers from poor results due to difficulties in identifying association rules related to the ontologies.

4.2 Evaluation: Ontology Alignment Evaluation Initiative Benchmark Track

To test the quality of mappings generated using BLOOMS, we ran it on the benchmark datasets using the reference alignment and compared its performance with the other systems mentioned above. Table 2 presents our results on the benchmark track of the ontology alignment initiative. As in the oriented matching track, the different tests 1XX, 2XX and 3XX comprise of matching a source ontology to other ontologies beginning with the prefix digit of the test. This test utilizes a larger number of ontologies than the oriented matching track. However, to a large extent the ontologies involved are identical.

In the benchmark track, BLOOMS is able to retrieve all results in 1XX, however, it results in loss of precision. In the 1XX track, the other systems gave varying performances. RiMOM and AROMA are impressive with their excellent precision and recall, whereas S-Match and OMViaUO suffer from retrieval of few and incorrect results.

BLOOMS does a better job in 3XX than 2XX due to the involvement of real world ontologies. It ranks right behind RiMOM and AROMA in its recall and does a decent job with respect to precision. The Alignment API does a significantly better job in retrieving the results and matching the ontologies, probably due to the fact, that this track involves finding equivalence relations between ontological concepts. The reasons for poor performance of the other systems are identical to those in the oriented track.

For the 3XX test, BLOOMS outperforms RiMOM and the other systems in finding the correct results. However, the increase in recall goes with a dip in precision. AROMA performs the best in terms of precision.

4.3 Evaluation: LOD Schema Alignment

For a comparative evaluation of BLOOMS on LOD schema alignment, it was necessary to provide a baseline for the alignment task. Since there are no established benchmarks or available baselines for measuring precision and recall for LOD schema alignment, we asked human experts familiar with the domains to create reference alignments.[16] The experts were asked to identify if the concepts belonging to the to-be-matched pairs of schemas are related to each other via a *subclass* or an *equivalence* relationship. In case the experts were not familiar with the terms they utilized descriptions of the concepts (if available in the ontology) or other appropriate references for identifying the relationships. The experts identified all possible subclass and equivalence mappings between the concepts of different ontologies. This process may result in some redundancy if equivalence has already been established at a top level concept C1 and C2. As a result of this mapping, subclass relationships between can be inferred between C1 and subclasses of C2 using this equivalence relationship automatically. This process is obviously subjective to some extent, but in the absence of a community agreed reference alignment, there is no other way to identify the accuracy of any of the systems. However, this phenomenon A similar methodology has been utilized previously in [9].

Table 3. LOD datasets=LOD datasets utilizing this schema, D=taxonomic depth, # C=number of classes, Linked datasets=LOD datasets they are linked to at the instance level

Schema	LOD datasets	D	# C	Linked datasets
DBpedia[17]	DBpedia	4	204	Geonames, US Census, Freebase
Geonames[18]	Geonames, Geospecies	2	11	DBpedia, Jamendo, FOAF Profiles
Music Ontology[19]	Jamendo, Music Brainz, DBTunes	4	136	GovTrack, DBpedia, Geonames
BBC Program[20]	BBC Programs, BBC Music	4	100	BBC Music, BBC Playcount Data
FOAF Profiles[21]	FOAF, Music Brainz	3	16	Crunch Base, QDOS, SIOC Sites
SIOC[22]	DBpedia, LinkedMDB	2	14	Virtuoso Sponger, FOAF Profiles, SemanticWeb.org
AKT Reference Ontology[23]	ACM, DBLP	5	17	Pisa, IEEE, eprints
Semantic Web Conference Ontology[24]	SW Conference Corpus	5	177	SemanticWeb.org, Revyu

[16] The reference alignments and the other material related to this work are available for download at http://wiki.knoesis.org/index.php/BLOOMS

[17] http://wiki.dbpedia.org/Downloads351/#dbpediaontology

[18] http://www.geonames.org/ontology/

[19] http://musicontology.com/

[20] http://purl.org/ontology/po/

[21] http://xmlns.com/foaf/spec/

[22] http://rdfs.org/sioc/ns#

[23] http://www.aktors.org/ontology/support

[24] http://data.semanticweb.org/ns/swc/ontology

Table 4. Results of various systems for LOD Schema Alignment. Legends: Prec=Precision, Rec=Recall, M=Music Ontology, B=BBC Program Ontology, F=FOAF Ontology, D=DBpedia Ontology, G=Geonames Ontology, S=SIOC Ontology, W=Semantic Web Conference Ontology, A=AKT Portal Ontology, err=System Error, NA=Not Available.

Linked Open Data Schema Ontology Alignment												
	Alignment API		OMViaUO		RiMoM		S-Match		AROMA		BLOOMS	
Test	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec
M,B	0.4	0	1	0	err	err	0.04	0.28	0	0	0.63	0.78
M,D	0	0	0	0	err	err	0.08	0.30	0.45	0.01	0.39	0.62
F,D	0	0	0	0	err	err	0.11	0.40	0.33	0.04	0.67	0.73
G,D	0	0	0	0	err	err	0.23	1	0	0	0	0
S,F	0	0	0	0	0.3	0.2	0.52	0.11	0.30	0.20	0.55	0.64
W,A	0.12	0.05	0.16	0.03	err	err	0.06	0.4	0.38	0.03	0.42	0.59
W,D	0	0	0	0	err	err	0.15	0.50	0.27	0.01	0.70	0.40
Avg.	0.07	0.01	0.17	0	NA	NA	0.17	0.43	0.25	0.04	0.48	0.54

Since the LOD cloud consists of more than 100 datasets, we had to make a selection for the purpose of evaluation. Table 3 gives the brief core data about the various LOD datasets which we used. We decided to use the schemas mentioned above due to the following reasons: (1) As specified in the second column of table 3, instance data on LOD are often created using a few well known schemas. For example, Jamendo, Music Brainz and AudioScrobbler primarily utilize concepts from the Music Ontology. The chosen datasets give significant coverage of the LOD cloud. (2) These datasets cover different domains such as Music, Publication and the Web. Thus they allow us to identify connections between ontologies belonging to different thematic domains. (3) Some of the dataset providers such as LinkedMDB have not made their schema publicly available. To eliminate any unfair advantage we might obtain as a result of ”self created schemas”, we decided not to include the datasets where the schema was not explicitly provided. Please note that the choice of the datasets was made a priori. The selection is not tailored to favor any specific system.

Discussion of Results: LOD Schema Alignment. The precision and recall values of the various systems performed on combinations of the various LOD schemas are listed in Table 4. The reasons for picking these combinations are the following: (1) The combinations correspond to datasets which are of related domains. For example, the Music Ontology and BBC Program both belong to the entertainment domain. Similarly, the Semantic Web Conference Ontology and the AKT Reference Ontology are for the scientific publication domain. (2) DBpedia schema is generic enough to encapsulate various kinds of domains and it can be matched to a large number of schemas. In a sense DBpedia can be understood as having an ”umbrella” function.[25]

On LOD schema matching, BLOOMS outperforms the other state of the art systems, as seen in Table 4. In the following paragraph, we examined each of the individual pairs and discuss possible reasons for the performance of the various systems next.

[25] This mirrors the central position which DBPedia currently has in the LOD Cloud.

(1) **Music Ontology and BBC Program:** These two datasets are very closely related to each other due to the reuse of concepts and similarity in domain. Unfortunately, RiMOM failed to work on these two ontologies, possibly because of their size.[26] AROMA did not find any relevant relations. OMViaUO finds only a few correct answers. OMViaUO was only able to match identical concepts being used across the two ontologies. The Alignment API also found same concepts, but there were also other concepts which were wrongly matched, which lead to a lower precision than that of OMViaUO. S-Match retrieves some of the results, but again due to the vast number of results computed by S-Match, its precision is low. BLOOMS performed significantly better, we think for the following reasons: (1) Concepts used in these ontologies are not commonly used English language terms (e.g., ”DAT”). They belong to the domain of signal recording and cannot be found in common thesauri. However, since Wikipedia contains many domains, they are properly categorized as well. (2) It is hard to match domains just on the basis of linguistic and structural matching.

(2) **Music Ontology and DBpedia:** For the matching of the Music Ontology with DBpedia, other than S-Match and BLOOMS all systems fail to deliver any noteworthy results. While S-Match identifies plenty of relevant relations, again due to the large number of erroneous results its precision is low. The Alignment API and OMViaUO fail to find any matching concepts besides alignments to owl:Thing. For the purpose of this evaluation, this particular correspondence (of concepts to owl:Thing) was excluded due to its highly obvious nature and lack of usefulness. Again we could not get RiMOM to work. AROMA identifies a few correspondences but few of them are correct.

(3) **FOAF Ontology and DBpedia:** The results in this case are similar to that of Music Ontology and DBpedia. S-Match registers a slight increase in both precision and recall. Similarly for AROMA there is an increase in the recall, but its precision drops. BLOOMS registers an increase in both precision and recall due to (a) the fact that concepts in FOAF are very specific and limited, such as ”PersonalProfileDocument”, and (b) the knowledge about these concepts is very well categorized in Wikipedia.

(4) **Geonames and DBpedia:** This is an interesting category, since apart from S-Match, all systems fail to deliver any results. This is partly due to the modeling of the Geonames ontology which is (a) very limited in nature, and (b) consists of concepts which are hard to understand and identify using only their names—examples are ”Code” and ”Feature”. It is hard to relate them to concepts in DBpedia due to their ambiguous meaning and absence of corresponding concepts in DBpedia.

(5) **FOAF and SIOC:** The FOAF and SIOC Ontologies consist of data related to on-line communities and social networks. RiMOM, which outperforms all the other systems on the ontology alignment initiative benchmarks, suffers from poor precision and recall on these ontologies. The performance of AROMA is identical to that of RiMOM. While they identify about a fifth of the relevant relations, they do so at the cost of low precision. S-Match, on the other hand, does a significantly good job at picking a few correct relations. The other two systems, OMViaUO and the Alignment API, fail to find any relevant relations.

[26] We contacted the authors for assistance in resolving the issue, but at the time of writing of this paper, they had not replied to our request with a solution for the problem.

(6) **The Semantic Web Conference Ontology and the AKT Reference Ontology:** These two datasets belong to the domain of scientific publications and hence consist of terms which are somewhat related to some of the terms in the OAEI. Correspondingly, BLOOMS performs well by retrieving more than half of the relations while S-Match retrieves about 40% of the relations. Again due to the vast number of correspondences retrieved by S-Match, its precision is low. AROMA retrieves largely correct but few relations. Due to its excellent performance in the OAEI track, we expected RiMOM to do well in this category. However, because of similar problems as above, RiMOM did not execute on this pair of ontologies. The Alignment API delivers better results than AROMA, whereas OMViaUO retrieves few results with a large number of wrong ones.

(7) **The Semantic Web Conference Ontology and DBpedia:** Using this pair we created alignment between the domain of scientific publication and general information. The overlapping concepts in the two ontologies consist of terms describing professions of people, events, and places. BLOOMS retrieves close to half of the correspondences with about 70% precision. S-Match retrieves half of the relations with a low precision. The other systems either do not retrieve any results or their results are insignificant.

Summary of the Results on LOD Schema Alignment. The results illustrate that on an average BLOOMS performs significantly better (40% better recall with at least twice the precision of the other systems) than the other state-of-the-art systems when it comes to ontology matching on the LOD cloud. Even individually, BLOOMS gives one of the highest recalls in 5 out of 7 pairs utilized for the purpose of evaluation. BLOOMS is a close second in one of the pairs. Of the other systems S-Match is impressive with its consistency in retrieving correct relations, however it comes at the cost of low precision. With regards to precision, BLOOMS leads in 6 out of the 7 pairs. Hence, by providing high recall, with high precision, BLOOMS makes it easy to curate the results for high quality mappings, thus making the results useful for practical purposes. [27]

Although S-Match gives decent recall, its low precision makes it difficult to work with the output. This is due to the vast number of mappings retrieved by S-Match, containing only relatively few correctly found ones. For example, S-Match found 3120 relations between concepts of BBC Program and the Music Ontology, of which only 4% were correct.

Of the other systems, AROMA gives decent precision but suffers from poor recall on LOD datasets. OMViaUO and the Alignment API suffer from both poor precision and recall on LOD datasets.

To summarize, these results indicate that state-of-the-art systems fail to provide the support required for a practically useful alignment of ontologies on the LOD cloud. On the other hand, BLOOMS provides significantly better precision and recall. The reasons for this significantly better performance lie in the fact that the BLOOMS approach is much better suited to handling the diverse domains of the LOD cloud datasets.

[27] A reviewer pointed out the advantage which BLOOMS might be obtaining due to the use of reasoner. Possibly other systems could also be improved by adding suitable post-processing by a reasoner. However, our experiments demonstrated that even without the reasoner BLOOMS is superior than other system on LOD schema alignment. For example, for Music Ontology and BBC Program schema alignment without using a reasoner results in precision and recall figures of 0.63 and 0.60 which are still significantly better than those obtained by other systems.

Thus, it significantly utilizes its advantage of using Wikipedia, a community-created data source, in dealing with community-created LOD datasets.

5 Related Work

To the best of our knowledge, this is the first work which exploits a generic and noisy categorization system such as Wikipedia in the context of ontology matching. In [18,19] the authors present a survey in the area of ontology matching.[28] Previously, Wikipedia categorization has been utilized for creating and restructuring taxonomies [8,20]. A taxonomy that covers popular approaches in database schema matching was presented in [21]. A generic algorithm for the same was presented in [22]. In [23] ontology schema matching was used to improve instance co-reference resolution. This helps in cleaning up the data and improving the quality of links at the instance level, but the issue of identifying appropriate relationships at the schema level has not been addressed. The voiD Framework [24] along with the SILK Framework [25] automate the process of link discovery between LOD datasets at the instance level. At the schema level, a notable effort for creating a unified reference point for LOD schemas is UMBEL [26], which is a coherent framework for ontology development which can serve as a reference framework.

6 Conclusion and Future Work

We have presented our approach—BLOOMS—for bootstrapping ontology alignment using the LOD cloud. Our results demonstrate that BLOOMS does not only significantly outperform state-of-the-art ontology alignment systems in LOD schema alignment; it also outperforms most other systems on the Ontology Alignment Initiative benchmark, and is roughly on par with the other best performing other system, RiMoM. We believe that BLOOMS draws its strength from (1) bootstrapping noisy data, and (2) the richness of Wikipedia which is used for the bootstrapping.

The fact that BLOOMS does so well on the benchmark is a rather pleasing result, in particular since it was developed solely with LOD schema alignment in mind. Indeed the initial motivation for BLOOMS came from a bottleneck in the LOD querying approach we are currently following—as outlined in [2]—which requires significant tool support for LOD schema matching in order to be scalable, in order to keep up with the growth of the LOD cloud. With BLOOMS we have made a major step towards solving this bottleneck; progress on our LOD querying approach will be reported elsewhere.

We can only hypothesize that BLOOMS will be able to keep up in the future, with the expanding LOD cloud. Our optimism is based on two core observations:

- The high precision and recall values which BLOOMS achieves on the LOD cloud, have in fact been achieved without significant fine-tuning. This gives ample optimism that there is room for improvement and future development, e.g., through adjusting the system to large thematic subdomains of the LOD cloud (e.g., life sciences), or by substituting the current use of DBPedia in our system by some other dataset which may only come into existence in the future.

[28] The ontology matching portal at http://www.ontologymatching.org/ gives a good review of the state-of-the-art research in this area.

- Due to the central use of DBPedia, i.e., of information exported from Wikipedia, it can be expected that the performance of BLOOMS will increase with the expansion of DBPedia/Wikipedia, since we will then have more data to bootstrap. So in a sense, assuming that DBPedia/Wikipedia will keep growing as the LOD cloud keeps growing, one might be tempted to say that BLOOMS has a bit of what some people call the "Google property"—namely that a system is getting better the more data it gets.

As future work we intend to identify other kinds of relationships such as partonomical relationships or disjointness on the LOD cloud. We therefore focused on the performance evaluation. We also intend to publicly release an upper level ontology for LOD based on existing upper level ontologies such as SUMO and DOLCE, created with significant but curated input of BLOOMS. We would also like to evaluate BLOOMS using other platforms such as OWL-API and other reasoners besides Jena. This will be the next few steps in our quest to LOD querying as outlined in [2]. Further, we would like to evaluate BLOOMS w.r.t scalability on ontologies larger than used for evaluation presented in Table 3.

Acknowledgement. This work is funded primarily by NSF Award:IIS-0842129, titled "III-SGER: Spatio-Temporal-Thematic Queries of Semantic Web Data: a Study of Expressivity and Efficiency". Pascal Hitzler acknowledges support by the Wright State University Research Council. We would like to sincerely thank Jérôme Euzenat for his insightful comments about the work which were extremely helpful in refining our manuscript.

References

1. Bizer, C., Heath, T., Berners-Lee, T.: Linked data – the story so far. International Journal On Semantic Web and Information Systems 5(3), 1–22 (2009)
2. Jain, P., Hitzler, P., Yeh, P.Z., Verma, K., Sheth, A.P.: Linked Data is Merely More Data. In: Brickley, D., Chaudhri, V.K., Halpin, H., McGuinness, D. (eds.) Linked Data Meets Artificial Intelligence, pp. 82–86. AAAI Press, Menlo Park (2010)
3. Polleres, A., Hogan, A., Harth, A., Decker, S.: Can we ever catch up with the Web? Semantic Web—Interoperability, Usability, Applicability (to appear), `http://www.semantic-web-journal.net/`
4. Hitzler, P., van Harmelen, F.: A reasonable Semantic Web. Semantic Web—Interoperability, Usability, Applicability (to appear), `http://www.semantic-web-journal.net/`
5. Euzenat, J.: An API for ontology alignment. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 698–712. Springer, Heidelberg (2004)
6. Guéret, C., Wang, S., Schlobach, S.: The Web of Data is a complex system—first insight into its multi-scale network properties. In: Proceedings of the ECCS 2010 European Conference on Complex Systems, Lisbon, Portugal (September 2010)
7. Bizer, C., et al.: DBpedia—A crystallization point for the Web of Data. Journal of Web Semantics 7(3), 154–165 (2009)
8. Ponzetto, S.P., Strube, M.: Deriving a large scale taxonomy from Wikipedia. In: Proceedings of the 22nd National Conference on Artificial Intelligence, pp. 1440–1445. AAAI Press, Menlo Park (2007)

9. Mascardi, V., Locoro, A., Rosso, P.: Automatic Ontology Matching via Upper Ontologies: A Systematic Evaluation. IEEE Trans. on Knowledge and Data Engr. 22(5), 609–623 (2010)
10. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: A Core of Semantic Knowledge. In: Williamson, C.L., et al. (eds.) Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12. ACM Press, New York (2007)
11. David, J., Euzenat, J., Scharffe, F., dos Santos, C.T.: The Alignment API 4.0 Semantic Web—Interoperability, Usability, Applicability (to appear), http://www.semantic-web-journal.net/
12. Jean-Mary, Y.R., Shironoshita, E.P., Kabuka, M.R.: Ontology matching with semantic verification. Journal of Web Semantics 7(3), 235–251 (2009)
13. Spiliopoulos, V., Valarakos, A.G., Vouros, G.A.: CSR: Discovering Subsumption Relations for the Alignment of Ontologies. In: Bechhofer, S., et al. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 418–431. Springer, Heidelberg (2008)
14. Li, J., Tang, J., Li, Y., Luo, Q.: RiMOM: A dynamic multistrategy ontology alignment framework. IEEE Transactions on Knowledge and Data Engineering 21, 1218–1232 (2009)
15. David, J., Guillet, F., Briand, H.: Matching directories and OWL ontologies with AROMA. In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management, CIKM 2006, pp. 830–831. ACM, New York (2006)
16. Hamdi, F., Safar, B., Niraula, N.B., Reynaud, C.: Taxomap in the OAEI 2009 Alignment Contest. In: Shvaiko, P., et al. (eds.) Proceedings of the 4th International Workshop on Ontology Matching (OM 2009) at the 8th International Semantic Web Conference (ISWC 2009) Chantilly, USA, October 25 (2009)
17. Giunchiglia, F., Shvaiko, P., Yatskevich, M.: S-Match: an algorithm and an implementation of semantic matching. In: Kalfoglou, Y., et al. (eds.) Semantic Interoperability and Integration. Number 04391 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany (2005)
18. Euzenat, J., Shvaiko, P.: Ontology matching (DE). Springer, Heidelberg (2007)
19. Choi, N., Song, I.Y., Han, H.: A survey on ontology mapping. SIGMOD Rec 35(3), 34–41 (2006)
20. Ponzetto, S.P., Navigli, R.: Large-scale taxonomy mapping for restructuring and integrating wikipedia. In: Boutilier, C. (ed.) Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, pp. 2083–2088 (2009)
21. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. The VLDB Journal 10(4), 334–350 (2001)
22. Madhavan, J., Bernstein, P.A., Rahm, E.: Generic schema matching with Cupid. In: Proceedings of the 27th International Conference on Very Large Data Bases, VLDB 2001, pp. 49–58. Morgan Kaufmann Publishers Inc., San Francisco (2001)
23. Nikolov, A., Uren, V.S., Motta, E., Roeck, A.N.D.: Overcoming schema heterogeneity between linked semantic repositories to improve coreference resolution. In: Gómez-Pérez, A., Yu, Y., Ding, Y. (eds.) ASWC 2009. LNCS, vol. 5926, pp. 332–346. Springer, Heidelberg (2009)
24. Alexander, K., Cyganiak, R., Hausenblas, M., Zhao, J.: Describing Linked Datasets – On the Design and Usage of voiD, the 'Vocabulary of Interlinked Datasets'. In: WWW 2009 Workshop on Linked Data on the Web (LDOW 2009), Madrid, Spain (2009)
25. Volz, J., Bizer, C., Gaedke, M., Kobilarov, G.: Discovering and maintaining links on the web of data. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 650–665. Springer, Heidelberg (2009)
26. Bergman, M.K., Giasson, F.: UMBEL ontology, volume 1, technical documentation, http://umbel.org/doc/UMBELOntology_vA1.pdf