Towards "On the Go" Matching of Linked Open Data Ontologies*

Isabel F. Cruz
ADVIS Lab
Department of Computer Science
University of Illinois at Chicago
ifc@cs.uic.edu

Matteo Palmonari†
DISCo
University of Milan-Bicocca
palmonari@disco.unimib.it

Federico Caimi
ADVIS Lab
Department of Computer Science
University of Illinois at Chicago
fcaimi@cs.uic.edu

Cosmin Stroe
ADVIS Lab
Department of Computer Science
University of Illinois at Chicago
cstroe1@cs.uic.edu

Abstract

Creating links between the schemas of published datasets is a key part of the Linked Open Data (LOD) paradigm. The ability to discover these links "on the go" requires for ontology matching techniques to achieve sufficient precision and recall within acceptable execution times. In this paper we add two methods to the AgreementMaker ontology matching system (but that could be used together with other systems). A first method consists of discovering a set of high-quality equivalence mappings based on concept similarity and of inferring subclass mappings from that set. A second method is based on the adoption of background terminology that serves as a mediator. Sets of synonyms and a hierarchy of concepts are taken into account, as well as subclass relationships involving external concepts that are imported into the ontologies being matched. Experimental results show that our approach has good performance and improves precision on most matching tasks when compared with a leading LOD approach.

1 Introduction

Linked Open Data (LOD) extends the linked data paradigm, which identifies a set of best practices to publish and share data on the web [Bizer *et al.*, 2009], to open licensed data. In order to integrate information from different datasets, the capability of establishing "correct" links among data is crucial. Linked data together with their schemas are usually represented by web ontologies that are defined using semantic web languages such as RDFS and OWL. The problem of establishing links between datasets [Volz *et al.*, 2009; Bizer *et al.*, 2009] is therefore closely related to the problem of ontology matching that has been investigated in the semantic web and in the database communities [Rahm and Bernstein, 2001; Euzenat and Shvaiko, 2007].

Recently, the need to perform matching "on the go" has been addressed for dynamic applications [Besana and Robertson, 2005; Togia *et al.*, 2010]. For example, the need to match terms issued by an agent (sender) to terms in another agent (receiver) where such communication may require only a transitory agreement between parts of the agents' ontologies is one such scenario [Besana and Robertson, 2005]. To address this scenario, possible research directions that are mentioned include the modification of "static" ontology matching approaches to include filters that emphasize some mappings over others, thus increasing efficiency. However, none of these dynamic applications are in the LOD domain.

Matching a set of input data and several LOD ontologies "on the go" requires new techniques: (1) for ontology matching that achieve a good trade-off between quality of the mappings and efficiency, (2) for classifying sentences against the knowledge base built from the LOD cloud integration, and (3) for evolving individual ontologies or the knowledge base to adapt to a stream of incoming statements. In this paper we focus on the first aspect.

Ontology matching in the linked data context faces new challenges for it has been shown that several ontology matching systems perform poorly when it comes to matching LOD ontologies [Jain *et al.*, 2010]. One of the reasons is that many ontology matching systems are better tailored to discovering equivalence relations. This is clearly a drawback in matching LOD ontologies because only a few equivalence relations can be found among concepts in different ontologies. Therefore the capability to discover subclass relations becomes crucial when the number of links among LOD sources increases.

Prior work in matching ontologies in LOD has been performed by the BLOOMS system [Jain *et al.*, 2010]. This work has introduced a new matching approach based on searching Wikipedia pages related to ontology terms: the categories extracted from these pages are then organized into graphs and used to match the terms in the ontology. BLOOMS performs better than other systems that were not designed with the goal of matching LOD ontologies, but were instead designed to work in "classic" ontology matching settings based on equivalence mappings, such as those in the

*Research partially supported by NSF IIS Awards 0513553 and 0812258 and by the Intelligence Advanced Research Projects Activity (IARPA) via Air Force Research Laboratory (AFRL) contract number FA8650-10-C-7061. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, AFRL, or the U.S. Government.

†Work partially performed while visiting the ADVIS Lab at UIC.

Ontology Alignment Evaluation Initiative (OAEI) competition [Euzenat *et al.*, 2010]. However, both the accuracy and the efficiency obtained by BLOOMS in LOD settings are far lower than those obtained by "classic" systems when performing tasks for which they were designed. BLOOMS is also not a top performer in "classic" ontology matching.

In this paper we extend AgreementMaker[1] [Cruz *et al.*, 2009a; 2009b], an ontology matching system for ontologies expressed in a wide variety of languages (including XML, RDF, and OWL), which has obtained some of the best results in the OAEI competition [Cruz *et al.*, 2010], with the objective of testing its viability in the LOD domain. Therefore, in this paper we address the following two questions. How can a system like AgreementMaker be extended to handle mappings other than equivalence mappings? Can AgreementMaker achieve good accuracy and efficiency in the LOD domain?

To address the first question, we present two methods. The first method evaluates the lexical similarity between two ontology terms to discover equivalence relations. Afterwards we exploit the equivalence relations to infer a first set of subclass relations between the ontology terms. The second method discovers a second set of subclass relations by comparing the two ontologies with a third-party ontology representing background knowledge.

As for the second question, we show that our approach achieves better results in matching LOD ontologies than any other ontology matching system in terms of average precision (over a set of tasks). In terms of average recall our approach is the second best after the BLOOMS system. In addition, our approach is more efficient than BLOOMS and has the advantage of consisting of methods that can be integrated with an existing ontology matching system. Because of these results, AgreementMaker is currently the only system that achieves top performance both in the "classic" and LOD domains.

The paper is organized as follows. Related work is discussed in Section 2. The proposed methods to improve ontology matching in the LOD domain are described in Section 3. The experimental evaluation of the proposed approach, based on previously proposed reference alignments [Jain *et al.*, 2010] is discussed in Section 4. Concluding remarks end the paper in Section 5.

2 Related Work

We discuss related work whose main focus is on schema-level mappings (as opposed to instance-level mappings [Volz *et al.*, 2009]). We also mention an approach that makes use of background information and finally we describe three approaches that use the "on the go" paradigm.

The data fusion tool KnowFuss uses schema-level mappings to improve instance co-reference [Nikolov *et al.*, 2009]. It does not, however, address the discovery of schema-level mappings. An approach for ontology matching that uses schema-level (as well as instance-level) mappings has been proposed in the context of geospatial linked datasets [Parundekar *et al.*, 2010]. This approach infers mappings between ontology classes by analyzing qualitative spatial relations between instances in two datasets. It is therefore specific to the geospatial domain.

The BLOOMS system features a new approach that performs schema-level matching for LOD. It consists of searching Wikipedia pages related to ontology concepts: the categories extracted from these pages (using a Web service) are organized into trees and are compared to support matching between ontology concepts [Jain *et al.*, 2010]. To evaluate ontology matching for LOD, BLOOMS uses seven matching tasks and defines the *gold standard* or *reference alignment* for those tasks. Their tasks consider pairs of popular datasets (e.g., DBpedia, FOAF, GeoNames). They compare BLOOMS with well-known ontology matching systems such as RiMOM [Li *et al.*, 2009], S-Match [Giunchiglia *et al.*, 2007], and AROMA [David *et al.*, 2006] that have participated in the Ontology Alignment Evaluation Initiative (OAEI) [Euzenat *et al.*, 2010]. They show that BLOOMS easily outperforms those systems in the LOD domain. However, in the OAEI tasks, when compared with those systems, BLOOMS produces worse results when discovering equivalence mappings even if achieving better results when discovering subclass mappings [Euzenat *et al.*, 2010].

The SCARLET system introduces the idea of looking for clues in background ontologies to determine the meaning of concepts [Sabou *et al.*, 2008]. It uses logic-based rules to infer mappings whereas in this paper we determine the overlap between sets of concept descriptors. SCARLET has not been evaluated in the LOD domain.

There are a few ontology matching approaches that specifically address the "on the go" matching paradigm (but have not been tested in the LOD domain). In one of them, an interesting matching process takes place where mappings between terms are dynamically discovered during the interaction between autonomous agents [Besana and Robertson, 2005]. Because only relevant portions of the ontologies are mapped at a time, this approach is quite relevant to the problem of matching sentences against a set of available ontologies. Another approach introduces an "on-the-fly" method to match RDF triples to support semantic interoperability in smart spaces [Smirnov *et al.*, 2010]. A third approach proposes a framework where folksonomies are used as mediators in the ontology matching process [Togia *et al.*, 2010].

3 Matching LOD ontologies

Given a *source ontology* S and a *target ontology* T, a *mapping* is a triple $\langle c_S, c_T, r\rangle$ where c_S and c_T are concepts in S and T, respectively, and r is a semantic relation that holds between c_S and c_T.

A set of mappings is called an *alignment*. A *reference alignment* is an alignment found by experts, against which the accuracy of other alignments, as measured in terms of precision and recall, can be determined. In *ontology matching* one attempts to find as many accurate mappings as possible using *matching algorithms*, which we call *matchers*.

We consider three types of semantic relations: *subclass of*, *superclass of*, and *equivalence*, or, for short, *sub*, *sup*, and *equiv*, respectively. Given these relations we can define three types of mappings: $\langle c_S, c_T, sub\rangle$, meaning that c_S is a *subclass* of c_T, $\langle c_S, c_T, sup\rangle$ meaning that c_S is a *superclass*

[1] http://www.agreementmaker.org.

of c_T, and $\langle c_S, c_T, equiv\rangle$ if and only if $\langle c_S, c_T, sub\rangle$ and $\langle c_S, c_T, sup\rangle$. In this case, c_S and c_T are *equivalent* classes.

Our approach to matching LOD ontologies is based on two methods:

Similarity-based mapping discovery. This method uses a *similarity metric* to compare the source and target ontologies: (1) to discover a set EQ_{sim} of equivalence mappings, and (2) to infer two sets of subclass and superclass mappings, respectively SUB_{sim} and SUP_{sim} from the equivalence mappings.

Mediator-based mapping discovery. This method compares the source and target ontologies to a third-party ontology, the *mediator ontology*, to discover two sets of subclass and superclass mappings, respectively SUB_{med} and SUP_{med}.

The alignment between S and T is defined as the union of the sets of mappings determined by the two discovery methods:

$$EQ_{sim} \cup SUB_{sim} \cup SUP_{sim} \cup SUB_{med} \cup SUP_{med}$$

Next, we first describe the algorithms that are at the core of each method and then we explain how these algorithms have been modified to improve the accuracy of the alignment.

3.1 Similarity-based Mapping Discovery

Equivalence mappings are discovered by evaluating a similarity value in the interval [0,1] between every pair $\langle c_S, c_T\rangle$ of source and target concepts, denoted $sim(c_S, c_T)$. The similarity value signifies the confidence with which we believe that the two concepts are semantically equivalent. A *matcher* computes the similarity values between all the possible pairs of concepts and stores the results in a similarity matrix. We use the *Advanced Similarity Matcher (ASM)* to compute the similarity matrix, which evaluates the lexical similarity between the two strings that represent both concepts [Cruz *et al.*, 2010]. For each pair of concepts and a threshold th, such that $sim(c_S, c_T) \geq th$, the mapping $\langle c_S, c_T, equiv\rangle$ is included in the set of equivalence mappings EQ_{sim}.

Starting from EQ_{sim}, we build SUB_{sim} and SUP_{sim} by considering subclasses and superclasses of the concepts c_S and c_T that appear in the mappings $\langle c_S, c_T, equiv\rangle \in EQ_{sim}$. We add to the set SUB_{sim} (respectively, SUP_{sim}) all the triples $\langle x_S, c_T, sub\rangle$ (respectively, $\langle c_S, x_T, sup\rangle$) such that x_S is a subclass of c_S (respectively, c_T is a subclass of x_T).

In the context of LOD, we note two important differences from ontologies that have been used for other tasks, such as those used in the OAEI competition [Euzenat *et al.*, 2010]. The first being a consequence of the use of subclass and superclass mappings that are derived from equivalence mappings. This means that a wrong equivalence mapping can propagate an error to all the derived mappings. For this reason, in the LOD domain we set a very high threshold, e.g., 0.95, while in several other domains thresholds in the range [0.6, 0.8] are usually adopted [Cruz *et al.*, 2010].

The second difference is that when compared with the OAEI datasets, LOD ontologies often use several concepts (e.g., *foaf:Person* in the Semantic Web Conference ontology) imported from other ontologies that need to be considered in the matching process. The above method is therefore refined by introducing a *global matching technique* for the external concepts used in an ontology.

For example, several external concepts used in LOD ontologies, such as *wgs84_pos:SpatialThing*—a concept referenced in the GeoNames ontology—are used across different ontologies; they could provide useful information in discovering additional mappings. That is, one could arrive at a mapping between *dbpedia:Person* and *wgs84_pos:SpatialThing* by knowing that *foaf:Person* has been defined as subclass of *wgs84_pos:SpatialThing* elsewhere.

We introduce the following technique. For each concept c_S in S that has been imported from an external ontology E, we search across several LOD ontologies (currently a restricted pool of well-known ontologies, such as DBpedia or FOAF, under the assumption that their concepts are shared in practice by a large community) for all concepts that are defined as subclasses of c_S and we match these concepts with the concepts of the target ontology using ASM. We perform the same for each concept c_T in T. In particular, if there is in some external ontology E a concept x_E such that x_E has been defined as subclass of c_S (respectively, c_T) and for some concept c_T (respectively, c_S), we have that $sim(x_E, c_T) \geq th$ (respectively, $sim(c_S, x_E) \geq th$) then $\langle c_S, c_T, sup\rangle \in SUP_{sim}$ (respectively, $\langle c_S, c_T, sub\rangle \in SUB_{sim}$).

3.2 Mediator-based Mapping Discovery

In order to discover the sets SUB_{med} and SUP_{med} we use an algorithm that takes as input the mediator ontology in addition to the source and target ontologies. The mediator ontology provides *background terminology* organized in a class hierarchy. Each concept is associated with a set of labels, which are synonyms of the concept. Conversely, a label can be associated with one or more concepts. In this paper our mediator ontology is WordNet, with the class hierarchy being the hyperonym hierarchy and the set of labels being the synsets.

As in Subsection 3.1, we compare every concept of the source ontology with every concept in the target ontology. However, this time we perform comparisons involving also the sets of labels and the sets of hyperonyms in the mediator ontology.

Step 1: Each concept in the source (respectively, target) gets associated with a set of concepts in the mediator ontology. This association is made through the concept labels: every time a label matches exactly a concept in the source (respectively, target) ontology, then that mediator concept becomes associated with the source (respectively, target) concept. Given a concept c, the set of mediator concepts associated with it is denoted BST_c (for *Background Synonym Terminology*).

Step 2: Each concept in the source (respectively, target) gets associated with a set of hyperonyms from the mediator ontology. This association is made through the previously built sets of synonyms. Given a concept c, we consider each concept in BST_c and extract its hyperonyms in the mediator ontology. Finally, we union all such sets thus obtaining a set for each concept c denoted BHT_c (for *Background Hyperonym Terminology*).

Step 3: We use the sets obtained in the previous two steps to build the sets of subclass and superclass mappings denoted respectively by SUB_{med} and SUP_{med} as follows: if there is a number $n \geq 1$ of synonyms of a concept c_S (respectively, c_T) that appear as hyperonyms of another concept c_T (respectively, c_S), then c_S is more general than c_T (respectively, c_T is more general than c_S), that is, $\langle c_S, c_T, sup\rangle \in SUP_{med}$ (respectively, $\langle c_S, c_T, sub\rangle \in SUB_{med}$). We experimentally set n to 2, which provides a good trade-off between precision and recall.

This three-step algorithm has potentially two shortcomings. The first one is that we seek labels in the mediator ontology that are similar to a concept in the source (or target) ontology. In certain cases a mediator label (e.g., bank) can be similar to an ontology concept (e.g., bank, meaning financial institution), but the mediator concept (e.g., mound) be semantically unrelated, thus decreasing precision. The second one is that it may not be possible to find a match among the mediator labels for ontology concepts that are noun phrases after tokenization (e.g., *Sports Event*), thus decreasing the algorithm's recall. Therefore, to improve precision and recall of the proposed algorithm, we refine Step 1 by performing "lightweight" semantic disambiguation that involves the superclasses and subclasses of the ontology concepts and the hyponyms and hyperonyms of the mediator concepts. We also add Step 4, which performs lexical analysis of noun phrases.

Step 1 (addition): When possible, we narrow the set BST_c into a subset $\overline{BST}_c$ by checking if some subclass (respectively, superclass) of c is related through synonyms with some hyponym (respectively, hyperonym) of some mediator concept x in BST_c, and, if this is the case, we include only that mediator concept x in $\overline{BST}_c$.

Step 4: In the case of noun phrases, we use a best effort approach that produces good results in practice. The concept denoted by a noun phrase is more specific than the concepts denoted by the individual names occurring in the noun phrases. Since in most cases the noun phrase narrows the scope of the *main noun* occurring in the phrase (e.g., *Sports Event* denotes a narrower concept than *Event*), and since in English the main noun is usually the last name occurring in the noun phrase, we use this knowledge to extract the main nouns and then attempt to find correspondences between them and the names of the concepts using ASM; based on these correspondences, we extrapolate subclass and superclass mappings. In particular, let $noun(c)$ be the main noun of a noun phrase denoting concept c. If $sim(noun(c_S), c_T)) \geq th$, then $\langle c_S, c_T, sub\rangle \in SUB_{med}$; if $sim(c_S, noun(c_T)) \geq th$, then $\langle c_S, c_T, sup\rangle \in SUP_{med}$.

4 Experimental Results

Table 1 lists the ontologies that we have used for our experiments, which are the same that were considered by the BLOOMS system[2] [Jain *et al.*, 2010], as no benchmark has been set otherwise for the LOD domain. The table shows the number of concepts in the ontologies and the number of external ontologies that they import.

[2]http://wiki.knoesis.org/index.php/BLOOMS.

Ontology	# Classes	# External ontologies
AKT Portal	169	1
BBC Program	100	2
DBpedia	257	0
FOAF	16	0
GeoNames	10	0
Music Ontology	123	8
Semantic Web Conference	172	0
SIOC	15	0

Table 1: Ontologies in the experimental dataset.

The evaluation settings consist of seven matching tasks, involving different types of comparisons. For example, Music Ontology and BBC Program are both related to entertainment, whereas some other comparisons involve general purpose ontologies, such as DBpedia.

Table 2 shows the comparison between the results obtained by AgreementMaker and the results previously obtained for the S-Match, AROMA, and BLOOMS ontology matching systems [Jain *et al.*, 2010]. We are omitting other results because they are not competitive [Jain *et al.*, 2010]. In addition, we have modified the reference alignment for the GeoNames–DBpedia matching task and marked such results clearly with "*", while reporting also on the results without this modification. We modified the reference alignment by including the subclass mapping between *dbpedia:Person* and *wgs84_pos:SpatialThing* and a set of mappings consistently derived from this one (namely mappings between all subclasses of *dbpedia:Person* and *wgs84_pos:SpatialThing*). This update makes sense in light of the fact that the subclass relation between *foaf:Person* and *wgs84_pos:SpatialThing* already appears in the reference alignment of several matching tasks. As can be seen in Table 2, our system achieves the best average precision (with or without the modification), while being the second best in average recall after BLOOMS. We comment next on the results obtained for each task.

Task 1. For the FOAF–DBpedia matching task, our system is the best one, both in precision and recall. This is because of our *global matching technique* described in Section 3.1, which finds correct mappings based on external ontologies and propagates those mappings through the subclasses of the involved concepts.

Task 2. For the GeoNames–DBpedia matching task, BLOOMS is not able to find mappings. This is because the GeoNames ontology has very little information in the ontology proper; instead, the actual categories are encoded in properties at the instance level. However, S-Match has perfect recall (100%), though precision is low (20%). Our global matching technique, which uses the external definition of concepts, is the reason why AgreementMaker outperforms all the other systems. With the modification to the reference alignment already mentioned, our results are extremely good, though comparison with the other systems was not possible in this case.

Task 3. For the Music Ontology–BBC Program matching task, the clear winner is BLOOMS, with AgreementMaker second. BLOOMS uses Wikipedia while we use WordNet, a generic background ontology. Wikipedia is very well suited

	S-Match		AROMA		BLOOMS		AgreementMaker	
Matching Task	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec
FOAF–DBpedia	0.11	0.40	0.33	0.04	0.67	0.73	0.72	0.80
GeoNames–DBpedia	0.23	1.00	0.00	0.00	0.00	0.00	0.26	0.68
GeoNames*–DBpedia*	-	-	-	-	-	-	0.88	0.88
Music Ontology–BBC Program	0.04	0.28	0.00	0.00	0.63	0.78	0.48	0.16
Music Ontology–DBpedia	0.08	0.30	0.45	0.01	0.39	0.62	0.62	0.40
Semantic Web Conference–AKT Portal	0.06	0.40	0.38	0.03	0.42	0.59	0.48	0.43
Semantic Web Conference–DBpedia	0.15	0.50	0.27	0.01	0.70	0.40	0.58	0.35
SIOC–FOAF	0.52	0.11	0.30	0.20	0.55	0.64	0.56	0.41
Average	0.17	0.43	0.25	0.04	0.48	0.54	0.53	0.46
Average*	-	-	-	-	-	-	0.62	0.49

Table 2: Comparison of AgreementMaker with other ontology matching systems.

Matching Task	Load	SB	MB	Total
FOAF–DBpedia	6.9	3.1	1.7	11.7
GeoNames–DBpedia	6.6	1.5	1.6	9.8
Music Ontology–BBC Program	16.0	3.7	4.7	24.4
Music Ontology–DBpedia	26.3	18.2	7.5	52.1
Semantic Web Conference–AKT Portal	3.5	2.1	2.8	8.3
Semantic Web Conference–DBpedia	7.9	8.1	2.4	18.5
SIOC–FOAF	0.1	0.2	1.7	2.0

Table 3: Execution times (in seconds) of the matching process (loading, similarity-based, mediator-based, and total).

for this kind of ontologies, because it covers their specific vocabulary.

Task 4. For the Music Ontology–DBpedia matching task, and in contrast with the previous task, our results are comparable with those of BLOOMS. We achieve higher precision, while BLOOMS achieves higher recall, in what constitutes a perfect swap between precision and recall, respectively 39% and 62% for BLOOMS and 62% and 40% for AgreementMaker. That is, our system presents only mappings for which it is very confident, thus favoring precision, while BLOOMS clearly favors recall. The next best system, S-Match, has reasonable recall (30%), albeit at the cost of very low precision (6%).

Task 5. For the Semantic Web Conference–AKT Portal matching task of scientific publications, BLOOMS favors recall again while AgreementMaker favors precision. S-Match favors recall at the cost of very low precision while Aroma favors precision at the cost of very low recall.

Task 6. For the Semantic Web Conference–DBpedia matching task, BLOOMS is the leader in both precision and recall, with AgreementMaker second. The conference domain is the same used in the OAEI competition. BLOOMS performs well because DBpedia is closely related to Wikipedia. S-Match has interesting recall (50%) but low precision (15%).

Task 7. For the SIOC–FOAF matching task, both general linguistic understanding and specific domain vocabulary are needed because SIOC is an ontology related to online communities. BLOOMS, S-Match, and AgreementMaker are close in precision (respectively, 55%, 52%, and 56%), but BLOOMS leads in recall because of its use of Wikipedia.

Table 3 shows the total execution times of AgreementMaker for the seven matching tasks as well as the times for the different subtasks, namely loading, mapping discovery using the similarity-based (SB) method and the mediator-based (MB) method. We note that the total time never exceeds one minute, even when large ontologies like the Music Ontology and DBpedia are being matched.

A complete comparison of all the systems in terms of execution time was not possible. However, we compared the performance of the Semantic Web Conference–AKT Portal matching task in BLOOMS and in AgreementMaker. While BLOOMS took 2 hours and 3 minutes, AgreementMaker performed the same task in only 8.3 seconds. We ran our experiments using an Intel Core2 Duo T7500 2.20GHz with 2GB RAM and Linux kernel 2.6.32-30 32 bits.

5 Conclusions

In this paper we described an efficient approach to schema-level ontology matching, which provides a step forward towards "on the go" ontology matching in the Linked Open Data (LOD) domain. We extended the AgreementMaker ontology matching system with two methods, one based on the evaluation of similarity between concepts and one based on the comparison of ontologies using background terminology.

The current leading system in the LOD domain is BLOOMS [Jain *et al.*, 2010], therefore this is the system with which to compare other systems. The work on BLOOMS has also led to the creation of seven tasks and respective reference alignments in the LOD domain, an important contribution to the field. While AgreementMaker outperforms BLOOMS in "classic" ontology matching settings [Euzenat *et al.*, 2010], in the LOD domain no system clearly outperforms the other. A general observation is that BLOOMS obtains better recall than AgreementMaker (in five of the seven tasks), while AgreementMaker obtains better precision than BLOOMS (in five of the seven tasks). However, given these results, AgreementMaker is the top performer in both "classic" and LOD ontology matching.

In the same way that in "classic" ontology matching no single strategy yields the best results in all cases, the same can be said for the LOD domain. For example, AgreementMaker needs to extend access to background information and in particular to Wikipedia for the matching of certain ontologies (e.g., Music Ontology and BBC Program). In spite of the good results obtained by BLOOMS and by AgreementMaker, precision and recall are lower than in the "classic" domains. The well-known trade-off between precision and recall appears to be more noticeable in the LOD domain.

The problem of ontology matching in the LOD domain is a necessary component of discovering meaning "on the go" in large heterogeneous data—this being the subject of this paper. Another important component consists of the ability to match sentences (for example, expressed in natural language or as RDF statements) against the large knowledge base that can be built from the LOD cloud—this will allow for answering queries, annotating text, and disambiguating terms.

References

[Besana and Robertson, 2005] Paolo Besana and David Robertson. Contexts in Dynamic Ontology Mapping. In *AAAI Workshop on Context and Ontology: Theory, Practice and Applications*, 2005.

[Bizer *et al.*, 2009] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked Data—The Story So Far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3):1–22, 2009.

[Cruz *et al.*, 2009a] Isabel F. Cruz, Flavio Palandri Antonelli, and Cosmin Stroe. AgreementMaker: Efficient Matching for Large Real-World Schemas and Ontologies. *PVLDB*, 2(2):1586–1589, 2009.

[Cruz *et al.*, 2009b] Isabel F. Cruz, Flavio Palandri Antonelli, and Cosmin Stroe. Efficient Selection of Mappings and Automatic Quality-driven Combination of Matching Methods. In *ISWC International Workshop on Ontology Matching (OM)*, volume 551 of *CEUR Workshop Proceedings*, 2009.

[Cruz *et al.*, 2010] Isabel F. Cruz, Cosmin Stroe, Michele Caci, Federico Caimi, Matteo Palmonari, Flavio Palandri Antonelli, and Ulas C. Keles. Using AgreementMaker to Align Ontologies for OAEI 2010. In *ISWC International Workshop on Ontology Matching (OM)*, volume 689 of *CEUR Workshop Proceedings*, 2010.

[David *et al.*, 2006] Jérôme David, Fabrice Guillet, and Henri Briand. Matching directories and OWL ontologies with AROMA. In *International Conference on Information and Knowledge Management (CIKM)*, pages 830–831, 2006.

[Euzenat and Shvaiko, 2007] Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer-Verlag, Heidelberg (DE), 2007.

[Euzenat *et al.*, 2010] Jérôme Euzenat, Alfio Ferrara, Christian Meilicke, Juan Pane, François Scharffe, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Sváb-Zamazal, Vojtech Svátek, and Cássia Trojahn dos Santos. Results of the Ontology Alignment Evaluation Initiative 2010. In *ISWC International Workshop on Ontology Matching (OM)*, volume 689 of *CEUR Workshop Proceedings*, 2010.

[Giunchiglia *et al.*, 2007] Fausto Giunchiglia, Mikalai Yatskevich, and Pavel Shvaiko. Semantic Matching: Algorithms and Implementation. In *Journal on Data Semantics IX*, volume 4601 of *Lecture Notes in Computer Science*, pages 1–38. Springer, 2007.

[Jain *et al.*, 2010] Prateek Jain, Pascal Hitzler, Amit Sheth, Kunal Verma, and Peter Z. Yeh. Ontology Alignment for Linked Open Data. In *International Semantic Web Conference (ISWC)*, volume 6496 of *Lecture Notes in Computer Science*, pages 402–417. Springer, 2010.

[Li *et al.*, 2009] Juanzi Li, Jie Tang, Yi Li, and Qiong Luo. RiMOM: A Dynamic Multistrategy Ontology Alignment Framework. *IEEE Transactions on Data and Knowledge Engineering*, 21(8):1218–1232, 2009.

[Nikolov *et al.*, 2009] Andriy Nikolov, Victoria Uren, Enrico Motta, and Anne Roeck. Overcoming Schema Heterogeneity between Linked Semantic Repositories to Improve Coreference Resolution. In *Asian Semantic Web Conference (ASWC)*, volume 5926 of *Lecture Notes in Computer Science*, pages 332–346. Springer, 2009.

[Parundekar *et al.*, 2010] Rahul Parundekar, Craig Knoblock, and José Luis Ambite. Aligning Geospatial Ontologies on the Linked Data Web. In *Workshop On Linked Spatiotemporal Data in conjunction with the International Conference on Geographic Information Science*, 2010.

[Rahm and Bernstein, 2001] Erhard Rahm and Philip A. Bernstein. A survey of approaches to automatic schema matching. *VLDB Journal*, 10(4):334–350, 2001.

[Sabou *et al.*, 2008] Marta Sabou, Mathieu d'Aquin, and Enrico Motta. Exploring the Semantic Web as Background Knowledge for Ontology Matching. In *Journal on Data Semantics XI*, volume 5383 of *Lecture Notes in Computer Science*, pages 156–190. Springer, 2008.

[Smirnov *et al.*, 2010] Alexander V. Smirnov, Alexey Kashevnik, Nikolay Shilov, Sergey Balandin, Ian Oliver, and Sergey Boldyrev. On-the-Fly Ontology Matching in Smart Spaces: A Multi-model Approach. In *International Conference on Smart Spaces and Next Generation Wired/Wireless Networking (NEW2AN)*, volume 6294 of *Lecture Notes in Computer Science*, pages 72–83. Springer, 2010.

[Togia *et al.*, 2010] Theodosia Togia, Fiona McNeill, and Alan Bundy. Harnessing the Power of Folksonomies for Formal Ontology Matching On-the-Fly. In *ISWC International Workshop on Ontology Matching (OM)*, volume 689 of *CEUR Workshop Proceedings*, 2010.

[Volz *et al.*, 2009] Julius Volz, Christian Bizer, Martin Gaedke, and Georgi Kobilarov. Discovering and Maintaining Links on the Web of Data. In *International Semantic Web Conference (ISWC)*, volume 5823 of *Lecture Notes in Computer Science*, pages 650–665. Springer, 2009.