Putting Instance Matching to the Test: Is Instance Matching Ready for Reliable Data Linking?

Silviu Homoceanu[1], Jan-Christoph Kalo[2], and Wolf-Tilo Balke[1]

IFIS TU Braunschweig, Mühlenpfordstraße 23, 38106 Braunschweig, Germany
{silviu,balke}@ifis.cs.tu-bs.de,
{j-c.kalo}@tu-bs.de

Abstract. To extend the scope of retrieval and reasoning spanning several linked data stores, it is necessary to find out whether information in different collections actually points to the same real world object. Thus, data stores are interlinked through owl:sameAs relations. Unfortunately, this cross-linkage is not as extensive as one would hope. To remedy this problem, instance matching systems automatically discovering owl:sameAs links, have been proposed recently. According to results on existing benchmarks, such systems seem to have reached a convincing level of maturity. But the evaluations miss out on some important characteristics encountered in real-world data. To establish if instance matching systems are really ready for real-world data interlinking, we analyzed the main challenges of instance matching. We built a representative data set that emphasizes these challenges and evaluated the global quality of instance matching systems on the example of a top performer from last year's Instance Matching track organized by the Ontology Alignment Evaluation Initiative (OAEI).

Keywords: Instance matching, owl:sameAs, link discovery, linked data.

1 Introduction

Fostered by the W3C Semantic Web group's initiative to build the "Web of Data", a massive amount of information is currently being published in structured form on the Web. Currently, more than 53 billion triples in over 300 data stores are available in the largest Virtuoso-based Semantic Web database (SWDB)[1]. The key point of this initiative and an important design principle of Linked Open Data (LOD) is that data from different sources is extensively inter-linked. This way queries can join information available in disjoint data stores with high precision. For example, for a query on biographic data and work of the film producer Martin Scorsese, biographic data could come from DBpedia while data about his work could come from LinkedMDB. All this is possible provided that DBpedia and LinkedMDB are inter-linked at least with respect to the entity of Martin Scorsese. The typical way for such cross-linkage

[1] The Virtuoso SWDB is accessible at `http://lod.openlinksw.com/` through a SPARQL endpoint.

T. Andreasen et al. (Eds.): ISMIS 2014, LNAI 8502, pp. 274–284, 2014.

between LOD sources is through owl:sameAs links. Unfortunately, today cross-linkage is not nearly as extensive as one would hope: The number of unique owl:sameAs links we counted on the aforementioned SWDB, is about 570 million. Many links are missing and from the ones available, a large part are trivial links between DBpedia, Freebase, and YAGO ([1]).

Under the name of *entity reconciliation* or *instance matching* (usually mixed up with instance-based ontology matching because instance matching is often required for ontology matching), the problem of finding identity links (owl:sameAs) between identifiers of the same entity in various data stores has been heavily researched (see [2–7]). These systems make use of techniques like probabilistic matching, logic-based matching, contextual matching, or heuristic matching based on natural language processing (NLP). Each approach shows strengths and weaknesses. But these particularities are hard to assess, since each system was evaluated on different data samples. The choice of data for the evaluation has a big influence on the results. For instance, there is a large number of class equivalence links between DBpedia and YAGO. If these two data sources build a significant part of the evaluation data then approaches like the one presented in [5] are favored. The verbose nature of the URIs also helps shallow NLP techniques favoring for instance the system presented in [7]. The situation is different for other selections like LinkedMDB and YAGO since the URIs provided in LinkedMDB are more cryptic and links to and from YAGO are rare.

Of course, instance-matching approaches have to be able to work with all kinds of entities from multiple data-stores. Again, this may boost the performance of some systems, since different aspects of an entity can be learned iteratively from various stores. On the other hand it can be detrimental to the overall data quality, since the more entities and entity types are available, the more probable it becomes for systems to generate incorrect identity links. Take for instance LINDA [5] which heavily exploits transitive links to support the inter-linking process. When it was evaluated on the Billion Triple Challenge corpus comprising entities from various stores the respective precision was about 0.8. For relaxed similarity constraints the precision even drops to 0.66. But with every third identity link being incorrect, this level of quality does not seem satisfactory for performing join queries or reasoning. In contrast, SLINT+ [7] reports an average precision of 0.96 on DBpedia and Freebase data.

But does this really mean that SLINT+ performs better? The respective precision was achieved on a biased set, representing a highly inter-linked extract from DBpedia and Freebase! It is therefore impossible to directly compare the performance of the two systems. To make systems comparable to one another, the Ontology Alignment Evaluation Initiative (OAEI) organizes a yearly evaluation event including an Instance Matching track. For the last year's evaluation[2] there were evaluation tests involving data value differences, structural heterogeneity and language heterogeneity. With small data value and structure alterations and involving a small extract (1744 triples and 430 URIs) from a single high quality data source (DBpedia), we will show that the tests do not accurately reflect the problems encountered in real-world data.

[2] http://www.instancematching.org/oaei/imei2013/results.html

Actually, judging by the 2013's OAEI evaluation results (sustained precision of over 0.9), instance matching systems seem to have reached a level of maturity. But considering the modest precision achieved by systems like LINDA on real-world data, this raises the question: Is instance matching ready for reliable data interlinking? To answer this question, we perform extensive real-world experiments on instance matching using a system which has proven very successful in OAEI tests. To the best of our knowledge, this is the first study that provides an in depth analysis over how effective instance matching systems are on real-world data.

2 The Instance Matching Problem

Instance matching is about finding and reconciling instances of the same entity in heterogeneous data. It is of special interest to LOD because the same entity may be identified with different URIs in different data stores and the owl:samesAs property useful for interlinking URIs of the same entity is not as wide-spread as needed.

In the context of LOD, given multiple sets of URIs D_1, D_2, …, D_n, with each set comprising all unique URIs of a data store, *matching* two instances of an entity can formally be defined as a function $match$:$URI \times URI \rightarrow$ {false, true} with:

$$match(URI_i, URI_j) := \begin{cases} true, if\ sim(URI_i, URI_j) > \theta \\ false, otherwise \end{cases} \quad with\ URI_i \in D_i\ , URI_j \in D_j$$

where $1 \leq i, j \leq n$, and *sim()* is a system dependent, complex similarity metric involving structural, value-based, contextual and other similarity criteria, and θ is a parameter regulating the necessary quality level for a match.

Based on this function, instance matching systems build an *equivalence class* for each entity. An equivalence class comprises all URIs used by any source to refer to some corresponding unique entity. For instance, considering only DBpedia, Freebase, YAGO and LinkedMDB, the equivalence class for the entity "Martin Scorsese" is:

{http://dbpedia.org/resource/Martin_Scorsese,
http://yago-knowledge.org/resource/Martin_Scorsese,
http://rdf.freebase.com/ns/m.04sry,
http://data.linkedmdb.org/resource/producer/9726,
http://data.linkedmdb.org/resource/actor/29575,
http://data.linkedmdb.org/resource/editor/2321}.

It's worth noticing that in contrast to general purpose knowledge bases like Freebase or DBpedia, specialized data stores like LinkedMDB have finer granularity, differentiating between Martin Scorsese as actor, editor, or producer. According to the owl:sameAs property definition in the OWL standard, all URIs referring to the same real world object should be connected through owl:sameAs. In consequence, all six URIs from the previous example should be linked by owl:sameAs relations. Of course one could argue that finer, context-based identity is required and that "Martin Scorsese, the producer" may not be the same as "Martin Scorsese, the actor". For further discussions regarding context-based similarity and identity see [8]. In this paper we adopt the definition as provided by the OWL standard for the owl:sameAs property.

Instance matching is an iterative process. Once some of the instances are matched either manually or by some system and owl:sameAs links have been established, more identity links can be found by exploiting the transitivity inherent in identities: Given that URI_A and URI_B represent the same real world object, the same applying for URI_B and URI_C implies that also URI_A and URI_C represent the same real world entity. Consequently, an owl:sameAs link between URI_A and URI_C can be created. However, the actual process of discovering sameAs links *is based on some similarity function and not on identity*. Similarity functions, however, are usually not transitive!

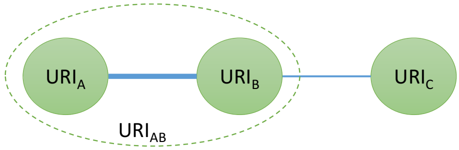

Fig. 1. Three URIs matching in a chain (URI_A and URI_C don't match). The similarity between URI_A and URI_B is stronger than the similarity between URI_B and URI_C.

Let us give a simplified example where the matching function relies on the Levenshtein distance on the rdfs:label property as similarity metric. Consider that a URI with rdfs:label "Scorsese, Martin" referring to the well-known movie producer, is matched with a URI with rdfs:label "Scorsese, Cartin" (which could be a typo). This last URI matches a URI with rdfs:label "Scorsese, Chartin" and the match process goes on up to a URI with rdfs:label "Scorsese, Charles". Charles Scorsese is an actor known for his role in Goodfellas and actually Martin's father. This problem is well known in the area of single link clustering: similarity clustering may lead to chains of URIs where neighboring URIs in the chain are similar, but for long enough chains the ends of the chain have almost nothing in common. Linking the URIs of Martin and Charles Scorsese with owl:sameAs would obviously be incorrect. Of course this example is constructed, but the danger of transitively matching unrelated instances in the context of large amounts of data is real. In consequence, evaluation data involving triples from multiple stores is necessary for exposing such weaknesses.

From the instance matching systems we found that only LINDA specifically addresses the problem of transitivity and selects only those matches consistent with transitivity as follows: On the example in Figure 1, considering that $sim(URI_A, URI_B) > sim(URI_B, URI_C)$, the equivalence class of URI_A comprises only URI_B and vice versa, i.e. both URIs refer the same entity and all properties valid for URI_A are also valid for URI_B and all properties valid for URI_B are also valid for URI_A. To express this we can denote the entity referred by URI_A and URI_B through URI_{AB}. Even though URI_A and URI_C don't show a large enough similarity, they are considered to refer the same entity if $match(URI_{AB}, URI_C)$ is true. Then, URI_C will also be added to the equivalence class. The process of finding identity links continues iteratively up to convergence.

Borrowing from hierarchical clustering, also the complete-linkage criteria could for instance be easily adopted to enforce transitivity. Assume after pairwise comparing all URIs we find three URIs matching in a chain like presented in Figure 1. Any set of n linked URIs satisfies the complete-linkage criteria, iff all n URIs match in a pairwise. Obviously this is not the case for chains. In consequence, chains are broken up by removing the weaker links. In the case of links of equal strength one of them is broken at random. Consider $sim(URI_A, URI_B) > sim(URI_B, URI_C)$. Since $match(URI_A, URI_C)$ is false, the link between URI_B and URI_C has to be removed. As a rule, the list of URIs being *weakly linked* to an URI_x is:

$$WL_{URI_x} = \{URI_y \mid \exists\, z: match(URI_x, URI_z) = true \land match(URI_y, URI_z) = false \land sim(URI_x, URI_z) \geq sim(URI_x, URI_y)\}.$$

After all weak links are broken for all URIs, the *equivalence class* of an URI is given by a function $E: URI \rightarrow \{URIs\}$ where:

$$E(URI_k) := \{URI_l \mid match(URI_k, URI_l)\}$$

3 Related Work

Instance matching is crucial for several applications like data integration, identity recognition and more important, for ontology alignment. Recognizing the lack of evaluation data, OAEI provided a reference benchmark for ontology alignment since 2004. Probably fostered by advances in Linked Data, four years later, [9] is one of the first publications to address this problem for instance matching. The authors discuss the particularities of instance matching and name main challenges. Based on these challenges, they design a benchmark with movie data from IMDb that emphasizes on data value differences, structure and logical heterogeneity. Finally, they compare the results for two instance matching algorithms to show the applicability of the data set.

In 2009, OAEI introduced an instance matching track and provided first generated benchmarks[3]: One comprising three datasets with instances from the domain of scientific publications built on Digital Bibliography & Library Project (DBLP), one with three datasets covering several topics, structured according to different ontologies from DBpedia and one generated benchmark obtained by modifying a dataset according to the data value, structure and logical heterogeneity criteria introduced in [9]. Evaluation data has gradually improved and last year's benchmark comprised five test cases: One for value transformation, where the value of five properties was changed by randomly deleting or adding characters; one for structure transformation, where the length of property paths between resources and values has been changed; a languages test where comments and labels were provided in French instead of English; one set combining value and structure transformation using French text and one where besides the value, structure and language challenges, some entities have none or multiple counterparts (a cardinality test). The data for the tests was extracted from DBpedia: it comprised 1744 triples, 430 URIs and only 11 predicates. It involves only one type of entity: Personalities from the field of computer science like Alan Turing, Donald E. Knuth, or Grace Hopper and is limited to triples having such personalities as a subject. Four instance matching systems have been evaluated on this benchmark. Out of the four, SLINT+ [7] and RiMOM [4, 10, 11] achieved outstanding results with an average precision and recall over all test of more than 0.9.

---

[3] http://oaei.ontologymatching.org/2009/instances/

While these results are quite promising, similar systems have proven weaker performance on real-world larger in size and involving multiple data stores. To assess the performance of such systems with real-world data, we built an evaluation set comprising 90,000 entities, from four domains, extracted from five data stores. In contrast to the OAEI test cases, all domains were included in all tests rendering cross-domain false positive matches (e.g. person being matched to movie) possible. The data stores were all-purpose knowledge bases like DBpedia and Freebase as well as domain focused stores like LinkedMBD and DrugBase. Some sources have cryptic URI naming conventions while some are more explicit. Also the granularity of properties varies between sources. We believe this is a more appropriate way of measuring the success of instance matching algorithms.

Table 1. Number of entities and properties per data store and entity type

Types	Freebase	DBpedia	LMDB	NYT	DrugBase
	#entities / properties				
Person	10,000 / 1,006	10,000 / 2,537	10,000 / 10	4,979 / 11	0
Film	10,000 / 465	10,000 / 565	10,000 / 48	0	0
Drug	5,000 / 435	5,000 / 247	0	0	6,712 / 36
Org.	5,000 / 641	0	0	3,044 / 11	0
#entities	30,000	25,000	20,000	8,023	6,712
#triples	1,749,433	2,461,263	264,902	90,850	314,108

4 Evaluation Data

For evaluating instance matching systems we rely on real-world data comprising entities of types Person, Film, Drug and Organization. The data was extracted from five stores: Freebase, DBpedia, LinkedMDB, DrugBase and NewYork Times. A detailed description of the data set is presented in Table 1. Instance matching systems are quite resource demanding ([5, 7]). For this reason, the evaluation data has a manageable size of about 90 thousand entities. This translates to about 4.9 million triple representing all relations having one of the selected entities as a subject. Such volume can be matched in a matter of minutes on commodity hardware. A similar number of entities was selected from each data store. The size difference between entity types was considered, too: Overall, the data set comprises about 35 thousand entities of type person, 30 thousand entities of type film, about 15 thousand drug entities, and about 8 thousand organizations. To emphasize data value problems, entities were selected after alphabetically ordering them on their labels. This way, almost all entities have labels starting with the letter 'A'. Due to the small number of entities, DrugBase and NewYork Times have been selected in full. The number of properties per entity type is, with a maximum of 2,537 unique properties for persons, significantly higher than in the OAEI tests. This stresses out structure heterogeneity of real-world data. The ontology differences between data sources, different aggregation levels introduced by LinkedMDB, or the fact that persons are being matched with actors add to the challenges this data set poses. Furthermore, in contrast to OAEI tests, having data form multiple stores increases the risk of building wrong transitive links. At the same time, the fact that multiple domains are compared, the possibility of creating bad links between entities of different types also exists. Finally, the selected data is not heavily interlinked. There are 5,855 owl:sameAs links between entities in our data set. 5,264 of them are between DBpedia and Freebase entities, 548 between DBpedia and LinkedMDB entities and 43 between entities from DBpedia and the NewYork Times.

To encourage further research on this topic, we made this data set, and data generated by our experiments, available at: http://www.ifis.cs.tu-bs.de/node/2906.

Table 2. The number of owl:sameAs links, the number of owl:sameAs links between entities of different types and precision obtained by SLINT+ and by performing the transitive closure on inks created by SLINT+ respectively

θ	SLINT+			cl_{TR}		
	#sameAs	Inter-domain	Prec.	#sameAs	Inter-domain	Prec.
0.95	8,020	33	0.91	2,055	89	0.20
0.75	16,739	119	0.71	5,498	216	0.15
0.50	17,436	230	0.76	7,038	396	0.09
0.25	25,113	1,734	0.67	14,879	2,408	0.02

5 Instance Matching - Experiments

To assess the quality of instance matching systems, we performed instance matching on the data presented in the previous chapter and measured *sampled precision*. We computed the transitive closure of the resulting owl:sameAs links and measured the quality of the newly created links. We paid special attention to the resulting equivalence classes as well as to entities of different types that have been matched. All tests were performed for high to low similarity thresholds. Since one of the characteristics of the data set was that it is not highly interlinked, there were not enough owl:sameAs links available to also measure recall.

The instance matching system is a black box from our perspective. Any domain independent system can be used. SLINT+ is one of the systems to achieve exceptional results in instance matching tasks. It is training-free and domain-independent. It builds on thorough predicate alignment and selection, shallow NLP and correlation based instance matching. It has already been successfully tested on selections from DBpedia and Freebase and it is available online for download[4].

For a similarity threshold of 0.95, SLINT+ creates 8,020 owl:sameAs links (see Table 2). 33 of them link drugs or movies to persons. They are obviously wrong. Overall, we observed a sampled precision of 0.91 for this threshold. The lower the similarity threshold, the more links are found. For a similarity threshold of 0.25, 25,113 links are found. Even for such a low similarity threshold the precision is with a

[4] `http://ri-www.nii.ac.jp/SLINT/index.html`

value of 0.67 quite impressive. According to the OWL standard, owl:sameAs links are transitive. Like most instance matching systems, SLINT+ ignores this aspect, probably because few bad links may lead to an explosion of bad links through transitivity. On the other hand completely ignoring transitive links is dangerous since any query engine using the links created by SLINT+ may transitively link sources to solve join queries. Computing the transitive closure of the owl:sameAs relations discovered by SLINT+ for a threshold of 0.95 we obtained an additional 2,055 links. However, the precision measured for these transitive links is only 0.20.

Table 3. Number of equivalence classes per number of URIs in the equivalence class, for various similarity thresholds

#URIs	# equivalence classes			
per class	θ=0.95	θ=0.75	θ=0.5	θ=0.25
2	4,168	5,054	7,008	8,180
3	529	1,160	2,023	2,781
4	54	222	315	648
5	15	110	136	303
6	7	49	67	167
7	1	24	38	89
8	4	22	22	52
9	5	12	17	43
10	2	11	12	27
11	2	4	8	13
12	2	8	9	9
13	0	1	3	12
14	0	3	1	7
15	1	6	4	6
16	1	1	3	5
17	0	1	3	7
18	1	1	2	4
19	1	1	2	1
20	1	2	2	4
21	1	1	2	2
22	0	1	2	3
23	1	1	1	2
24	0	0	2	1
27	0	1	0	1
29	0	1	1	1
31	0	0	0	1
38	0	0	1	1

But how is this possible? As discussed in Section 2, due to the non-transitive nature of the similarity function, long chains of entities belonging to the same equivalence class may be created. The longer the chain, the higher the probability that URIs that are far apart in the chain refer different entities. Even for high precision oriented similarity thresholds like 0.95, SLINT+ produces 11 equivalence classes with more than 10 URIs each. Actually, the largest equivalence class has 23 URIs, while for lower similarity thresholds there are equivalence classes with 38 URIs (see Table 3). One false owl:sameAs link connecting two smaller equivalence classes in such a large class creates a huge explosion of false links. Assuming two equivalence classes each having 10 URIs, one false link created by SLINT+ connecting the two classes may generate up to 100 incorrect links (all pairwise combinations developing between the two classes: C_2^{20}- $2 \cdot C_2^{10}$). Considering the high precision for 8,020 links but the low precision for all transitive links, the real, overall precision achieved by SLINT+ for a threshold of 0.95 is $\frac{8{,}020*0.91+2{,}055*0.20}{8{,}020+2{,}055} = 0.77$ and thus quite comparable to LINDA.

Not knowing all owl:sameAs links for all entities from our data set it is impossible to accurately measure *recall*. However, if we take into consideration that 25,113 entities were found with a precision of 0.67 and that an additional 14,879 were found with a precision of 0.02, we can assume that the data set should have, when correctly interlinked, at least 17,123 links (25,113 * 0.67 + 14,879 * 0.02). Assuming that 8,020 * 0.91 + 2,055 * 0.20 = 7,709 correct links have been discovered for a threshold of 0.95, this translates into a recall of at best 0.45. This is significantly lower than the results observed on the OAEI benchmark.

To sum up, results for today's instance matching systems seem quite impressive. But if the problem of transitivity is not properly considered, even for very high similarity thresholds the precision on links obtained through transitivity is catastrophic.

6 Conclusions and Future Work

The most important benefit of linked open data is that it creates a unified view of entities by tapping into information from different data stores. The standard mechanism for connecting instances of the same entity is to transitively exploit owl:sameAs properties. But to do this, first all individual instances of real-world entities have to be linked. Since manually creating all sameAs links is hardly feasible, instance matching systems mostly rely on similarities to automatically create sameAs links for subsequent traversal. The slight problem is that similarity functions are not transitive.

In this paper, we have shown that, even for high similarity thresholds (θ=0.95), ignoring the missing transitivity may have catastrophic effects over the quality of the discovered links. In our experiments for a top-rated system it translated into an overall precision of less than 0.8 for a recall lower than 0.45. In conclusion, unfortunately, today, instance matching is not yet ready for reliable automatic data interlinking.

While our results on one hand call for ways of enforcing transitivity in instance matching systems, they also call for better evaluation within the OAEI instance matching track. A starting point is the data set constructed in this paper. But transitivity problems are by no means the only problems that have to be reflected in the evaluation benchmark. Similar challenges for instance matching, first introduced in [9], are:

- **Data Value Differences:** The same data may be represented differently in different sources. For instance a company's name may be "IBM" in one source and "International Business Machines Corporation" in another.
- **Structural Heterogeneity:** A data type property in one source may be defined as an object property in another source. Multiple properties from one source (first name and last name) may be composed into a single property in other sources (name). One source may have three values for a property while in another source the same property has just one value.
- **Logical Heterogeneity:** Instances of the same real-world object may belong to different concepts. These concepts may be subclasses of the same superclass. Two instances having the same property values may belong to disjoint classes.

Considering all this, a proper data set for evaluating instance matching systems should have triples from multiple stores for transitivity reasons, with a certain level of overlap between domains and different levels of data quality to address data value differences, and it should include sources having properties with different levels of cardinality and granularity to address structural and logical heterogeneity.

In the near future, after building a benchmark for proper matching evaluation we plan to analyze the transitive closures and their respective precision/recall also for RiMOM2013 and LogMap, two other state of the art systems. Moreover, we will thoroughly analyze ways of enforcing transitivity by design in instance matching.

References

1. Ding, L., Shinavier, J., Shangguan, Z., McGuinness, D.L.: SameAs Networks and Beyond: Analyzing Deployment Status and Implications of owl:sameAs in Linked Data. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010, Part I. LNCS, vol. 6496, pp. 145–160. Springer, Heidelberg (2010)
2. 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)
3. Isele, R., Jentzsch, A., Bizer, C.: Silk Server - Adding missing Links while consuming Linked Data. In: Int. Workshop on Consuming Linked Data, COLD (2010)
4. Tang, J., Li, J., Liang, B., Huang, X., Li, Y., Wang, K.: Using Bayesian decision for ontology mapping. Journal Web Semant. Sci. Serv. Agents World Wide Web. 4, 243–262 (2006)
5. Böhm, C., de Melo, G.: LINDA: Distributed web-of-data-scale entity matching. In: Conf. Information and Knowledge Management, CIKM (2012)
6. Jiménez-Ruiz, E., Cuenca Grau, B.: LogMap: Logic-Based and Scalable Ontology Matching. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 273–288. Springer, Heidelberg (2011)

7. Nguyen, K., Ichise, R., Le, B.: Interlinking Linked Data Sources Using a Domain-Independent System. In: Takeda, H., Qu, Y., Mizoguchi, R., Kitamura, Y. (eds.) JIST 2012. LNCS, vol. 7774, pp. 113–128. Springer, Heidelberg (2013)
8. Halpin, H., Hayes, P.J.: When owl:sameas isn't the same: An analysis of identity links on the semantic web. In: Linked Data on the Web, LDOW (2010)
9. Alfio Ferrara, D.L.: Towards a Benchmark for Instance Matching. In: Int. Workshop on Ontology Matching, OM (2008)
10. Tang, J., Liang, B.-Y., Li, J., Wang, K.-H.: Risk Minimization based Ontology Mapping. In: Chi, C.-H., Lam, K.-Y. (eds.) AWCC 2004. LNCS, vol. 3309, pp. 469–480. Springer, Heidelberg (2004)
11. Zheng, Q., Shao, C., Li, J., Wang, Z., Hu, L.: RiMOM2013 Results for OAEI 2013. In: Int. Workshop on Ontology Matching, OM (2013)