Avoiding Chinese Whispers: Controlling End-to-End Join Quality in Linked Open Data Stores

Jan-Christoph Kalo, Silviu Homoceanu, Jewgeni Rose, Wolf-Tilo Balke

Technische Universität Braunschweig
Mühlenpfordstraße 23
38106 Braunschweig, Germany
{kalo, silviu, rose, balke}@ifis.cs.tu-bs.de

ABSTRACT

Today Linked Open Data is a central trend in information provisioning. Data is collected in distributed data stores, individually curated with high quality, and made available over the Web for a wide variety of Web applications providing their own business logic for data utilization. Thus, the key promise of Linked Open Data is to provide a holistic view for a wide range of data items or entities. But parallel to the problems of database integration or schema matching, *linking data over several sources remains a challenge* and is currently severely hampering the vision of a working Semantic Web. One possible solution are instance matching systems that automatically create *owl:sameAs* links between data stores. According to existing benchmarks, the matching quality has even reached a satisfying level. However, our extensive analysis shows that *instance matching systems are not yet ready for large-scale data interlinking*. This is because query processors joining even via a single incorrectly created link implicitly use also *all transitive owl:sameAs* links that may in turn be mismatched again. The result is similar to the game Chinese Whispers: watered-down *sameAs* semantics step-by-step lead to a terrible end-to-end quality of joins. We develop innovative structural mechanisms on top of instance matching systems to significantly improve query processing avoiding Chinese Whispers.

Categories and Subject Descriptors

H.2.4 [**Database Management**]: Query Processing

General Terms

Algorithms, Measurement, Performance

Keywords

Semantic Web, Linked Open Data, Entity Resolution, Instance Matching, Distributed Query Processing, Joins

WebSci '15, June 28 - July 01, 2015, Oxford, United Kingdom
 ISBN 978-1-4503-3672-7/15/06…$15.00
DOI: http://dx.doi.org/10.1145/2786451.2786466

1. INTRODUCTION

Linked Open Data (LOD) has become a central trend in information provisioning with currently more than a thousand sources. Backed by the W3C Semantic Web group's initiative to connect and share data on the Web, LOD now has grown to more than 5 billion triples[1]. Mostly entity-centric data (about *persons*, *scientific publications*, *movies*, or *proteins*) is collected in distributed data stores, individually curated with high quality, and made available over the Web for a wide variety of Web applications providing their own business logic for data utilization. However, also collection and curation of data follows the individual logic of the data store, which thus may strongly differ between sources, i.e. what data a source is interested in and how it is represented.

One key goal of the Semantic Web is to disambiguate real-world objects on the Web by uniform resource identifiers (URIs) [2]. Yet, different LOD stores may represent the same entity with different URIs, e.g., http://dbpedia.org/resource/Goldfinger_(film) and http://data.linkedmdb.org/resource/film/9974 both represent the James Bond movie 'Goldfinger' in two different stores, namely DBpedia and LinkedMDB. Moreover, both stores contain only partial information about the movie: while LinkedMDB provides more details about the actors, some basic properties like movie runtime, the music composer or the movie's budget are only provided by DBpedia. *Here an entity-centric join operation would realize the actual promise of LOD: combining and integrating the information of several stores to get a global view of any entity.*

To deal with ambiguous URIs, the Semantic Web introduced a special language construct: *owl:sameAs*[2] relations specifying that two different URIs represent the same real-world entity. In theory using these relations any query processor should be able to join the information about a given entity from LOD sources. However, analyses show that *owl:sameAs* interlinkage is not as extensive as one would hope: it mainly relies on manual labor. Furthermore, a large part are trivial links between DBpedia, Freebase, and YAGO [8, 9]. Because a lack of interlinkage is causing low information recall in join queries, the problem of automatically finding identity links (*owl:sameAs*) between URIs of the same entity in various

[1] http://stats.lod2.eu/

[2] http://www.w3.org/TR/2004/REC-owl-guide-20040210/#owl_sameAs

data stores has been heavily researched under the name of entity reconciliation or instance matching [17, 18, 22, 24, 26].

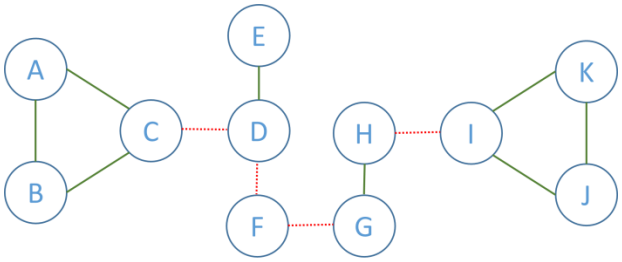


Figure 1: Graphical representation of *owl:sameAs* links created by PARIS on movie data. Nodes represent instances, edges correct owl:sameAs links and dotted edges incorrect links.

In fact, since 2009 there is a dedicated track for instance matching research[3] in the Ontology Alignment Evaluation Initiative (OAEI) co-located with the International Semantic Web Conference. The goal of the track was to encourage the research and development of automatic systems for creating new links to improve recall while maintaining high precision. The latest results are quite promising: three of the four participating systems constantly achieve precision and recall of more than 0.9 with respect to a standardized benchmark. With such positive results, these systems should be able to automatically create high quality *owl:sameAs* links for the whole LOD cloud, therefore improving information recall tremendously. Unfortunately, this seems not to be the case: despite the systems showing high precision during benchmarking, we observed that the quality of the information obtained by joining data outside of the benchmark is low [16].

How can this be? In a nutshell, instance matching systems compare all URIs for similarity in a pairwise fashion. Once the similarity of some pair meets a certain threshold, the two URIs are considered to refer to the same entity and a respective *owl:sameAs* link is created. But query processors joining entity data with the help of these links also use their respective transitivity. If A is sameAs B and B is sameAs C, then A *must be* sameAs C according to the sameAs semantics, whether some instance matching system has actually declared A sameAs C does not matter.

Let us look at a real world example in the field of *movies*: Over LinkedMDB, Freebase and Yago, the state-of-the-art instance matching system PARIS [26] finds highly similar entity pairs (Figure 1) between the movies '49 up' (marked as D) and '39 up' (marked as G), which are two episodes in Michael Apted's 'Up series' (marked as F) of documentary movies. While these mistakes are fully explainable (given that both movies and the series share a lot of characteristic attributes), PARIS matches the movies to some more instances. In fact, PARIS creates correct *owl:sameAs* links (green edges) between instances A, B, C and D, E and G, H and also I, J, K. Furthermore four incorrect links (dotted edges) between C and D, between D and F, between F and G, and between H and I are created. Even though 8 out of the 12 explicitly created links are correct (a precision of 0.67), an entity-centric join would implicitly also rely on *all pairwise transitive links* between the groups ABC, DE, GH, and IJK: adding a total of 43 more incorrect links (AD, AE, BD; BE, AF, BF, CF, EF,…) and deteriorating the join precision to a mere 0.15.

[3] http://islab.di.unimi.it/im_oaei_2014/index.html

Thus, in LOD clouds with distributed entity information, adding only a few incorrect *owl:sameAs* links, can result in chains connecting groups of entities, where entities at located at the respective ends have virtually nothing in common anymore. This phenomenon – known from the game Chinese Whispers – is obviously destroying a vital facet of LOD's usefulness and thus endangering the Semantic Web's success at large. The main contributions of this paper are:

1. We address the lack of large-scale benchmarks with real-world multi-source LOD: we designed a *new instance matching benchmark* containing more than 110,000 instances comprising general LOD data, as well as data from expert knowledge LOD stores. Containing multi-source entities, such a benchmark is vital for practical research on Linked Open Data interlinkage.
2. We prove the *quality problem* of entity-centric joins on Linked Open Data using *owl:sameAs* links. Our analysis shows that even though state-of-the-art instance matching systems create links with a precision above 90%, the average quality of joins on this data is much worse.
3. We present *four novel approaches* on top of instance matching systems or on *owl:sameAs* links to improve the quality of joins over the created links. In particular, our approaches improve the end-to-end quality of evaluated instance matching systems to over 90% precision.

This paper is structured as follows: Section 2 defines basic terms for instance matching and LOD joins. Current matching systems and benchmarks, as well as a large real-world LOD benchmark are discussed in Section 3. Additionally an analysis of two state-of-the-art systems on this dataset is provided. Section 4 presents approaches for avoiding the Chinese Whispers effect. Finally in Section 5 we evaluate our approaches on a real-world benchmark.

2. BASIC NOTIONS & CONSIDERATIONS

Interlinking data sources by identifying different descriptions of the same entity is referred to as 'instance matching'. With regard to Linked (Open) Data, descriptions in the form of RDF triples from several data providers are given as input.

Definition 1 (Instance Matching Problem): *Given a set of entities* $E = \{e_1, e_2, \dots, e_n\}$ *together with their attributes and relations, Instance Matching is the task of finding a partition* $P = \{p_1, p_2, \dots, p_m\}$ *for* $m \leq n$*, such that every entity is contained in exactly one partition and that every partition only contains entities describing the same real-world object.*

Interlinking LOD sources on entity level has been identified as one of the major challenges in the Semantic Web community [3–5]. To overcome the lack of interlinkage, several systems for solving the Instance Matching Problem by creating new *owl:sameAs* links have been developed in the last years. As input, RDF data for a set of entities $E = \{e_1, e_2, \dots, e_n\}$ from two or more data sources are provided. Without loss of generality, for each pair of entities $(e_i, e_j) \in E \times E$ a similarity value $(sim(e_i, e_j))$ between 0 and 1 is computed. A high similarity value indicates that the corresponding instances are very likely to be identical, a low value on the other hand is a strong indication that they are not. Obviously, similarity is reflexive and with regard to instance matching, it is even symmetric. However, similarity is *not* a transitive relation (unlike 'sameAs'). If transitivity is assumed, it would be possible that small differences between two instances form a long chain, where the error (difference between

the instances) is propagated step by step. For better understanding we describe a well-known example by Luce [21]: Consider 401 cups of coffees, containing $\left(1 + \frac{i}{400}\right)$ for $i = 0, \ldots, 400$ cubes of sugar. The amount of sugar in cup i and $i + 1$ is nearly identical, but for cup 0 and 400 it is totally different. A similar phenomena can be observed for single-linkage clustering and is known under the name of *chaining effect*. Points being clustered via single-linkage clustering can form long chains where neighboring points are always similar, but the cluster itself is very heterogeneous.

Nevertheless, in instance matching similarity functions are used to create new *owl:sameAs* links, when the similarity value is above a certain threshold.

Definition 2 (Matching Function)*: Given binary entity similarity function $sim(.,.)$, threshold Θ and similarity value ($sim(e_i, e_j)$) for two arbitrary entities e_i and e_j, a matching function can be defined as $match: E \times E \rightarrow \{true, false\}$ with:*

$$match(e_i, e_j) = \begin{cases} true, & sim(e_i, e_j) > \Theta \\ false, & sim(e_i, e_j) \leq \Theta \end{cases}$$

Unlike similarity relations, *owl:sameAs* is an equality relation. If URI *http://dbpedia.org/resource/Always_Outnumbered* has an *owl:sameAs* link to *http://data.linkedmdb.org/page/film/49169* these two instances really represent the same entity. Moreover, an additional *owl:sameAs* link from DBpedia to *http://yago-knowledge.org/resource/Always_Outnumbered_(film)* would also be a valid identity link, since both URIs describe the same movie. Since all equality relations are transitive, also the URI in LinkedMDB and YAGO must represent the same movie.

As we have seen, using a matching function for creating new *owl:sameAs* links with similarity functions that are not 100% reliable, can result in chaining effects. Hence, not all entities connected via such links are identical, cf. the example in Figure 1. Still, *owl:sameAs* is an equality relation, so instances connected via *owl:sameAs* links form an equivalence class, where every instance describes the identical real-world object.

Definition 3 (Equivalence Class)*: An equivalence class of instance matching results can be considered as a connected graph $G = (I, M)$ where $I = \{i_1, i_2, \ldots, i_q\}$ is a set of instances representing the vertices of a graph and $M = \{m_1, m_2, \ldots, m_p\} \subseteq I \times I$ is a set that represents the owl:sameAs links between these vertices.*

In theory, any equivalence class should be a set of instances that all describe identical entities. Therefore, in query processing a join over an equivalence class G has to use all transitive links, too. The set of transitive links can be computed by a transitive closure:

Definition 4 (Transitive Closure)*: The transitive closure of the equivalence class G is defined as:*

$$G^+ = \{(i_a, i_b): i_a, i_b \in I\}$$

As an example, we illustrate the computation of the transitive closure with the aid of the equivalence class shown in Figure 1. This equivalence class contains four incorrect links (the dotted edges in the illustration). A transitive closure would consist of links between any pair of instances of this class. Hence, it would contain 43 new transitive links which are *all incorrect*. A query processor joining over the existing links, would compute these transitive links. Therefore, the quality of the join is much worse, since it does not only contain four explicit incorrect links, but also the 43 transitively computed links. To measure this overall join quality, we define the end-to-end quality of joins:

Definition 5 (End-to-End Join Quality): *An entity-centric join using the matches of some equivalence class G with size n has a end-to-end quality defined through:*

$$Q(G) = \frac{2 \cdot \#correct\ links}{n \cdot (n-1)}$$

The end-to-end quality measures the precision of the existing *owl:sameAs* plus the precision of the transitive links. The end-to-end quality of our example class is only 0.15 even though the precision of the explicit links was 0.67. To prevent these quality issues for joins, *owl:sameAs* links between not identical instances have to be identified and removed.

3. ANALYSIS OF INSTANCE MATCHING

For more than 50 years Instance Matching has been heavily researched under the names of entity reconciliation, record linkage, and duplicate detection in the database community [11]. Linked Open Data is similar to distributed data management, as it contains structural heterogeneities, has different vocabularies and covers multiple domains. But, the size of the data and the number of data sets is usually larger than in classical databases. For automatically interlinking LOD sources on entity level, researchers have therefore developed specialized instance matching systems. These systems make use of a variety of techniques like probabilistic matching, logic-based matching, contextual matching, or heuristic matching based on natural language processing (NLP). As usual, each approach shows specific strengths and weaknesses.

For example SLINT+ by Nguyen et al. relies on the matching of predicates of corresponding instances [24]. For interlinking two data sources, first the important predicates for each source are filtered and a predicate alignment is built. In the next step, candidate pairs of instances are generated to avoid a quadratic number of comparisons between the two input sources. On this candidate set a matching using the aligned predicates is performed to compute a similarity value. In the paper, the performance has been evaluated on 9 datasets extracted from the LOD cloud. The average precision and recall are 0.97. With these results, SLINT+ clearly outperforms similar instance matching systems, namely Agreement Maker [7], SERIMI [1] and Zhishi.Links [25].

The PARIS (Probabilistic Alignment of Relations, Instances, and Schema) system uses probability estimates to create matchings [26]. With respect to instances, the probability that two instances are equivalent is computed by measuring the similarity of attributes and iteratively taking the probabilities for related instances into account. For evaluation, data from DBpedia, YAGO and IMDb was used and matched against each other. Between YAGO and DBpedia a precision of 90%, between YAGO and IMDB a precision of 97% and between DBpedia and IMDB a precision of even 100% was measured. The good pairwise matching results can be explained by the authors' assumption that each individual LOD ontology (i.e. source) does not contain any equivalent instances. Hence, every instance in a source has at most one matching partner in some other source. According to the authors this prevents chains of links. But of course this does not hold when more than two data sets are interlinked simultaneously.

Lately, two systems, SiGMa [18] and ARIA [19] have been presented. Both systems have achieved slightly better results than PARIS. SiGMa uses a simple greedy approach to allow a highly efficient matching process on datasets with millions of entities. The similarity of instances is measured using string metrics between literals and the similarity of neighbored instances. The

quality of results is comparable to PARIS, but SiGMa performs fifty times faster.

ARIA (Asymmetry Resistant Instance Alignment) addresses three basic asymmetries in instance matching that have been identified as major challenges by the authors: concept, feature and structure asymmetry. For the alignment process, first blocks of instances belonging to the same concept are created. Afterwards discriminative feature combinations are used to split up these into smaller groups, such that they only contain highly similar instances. Finally, the similarity of triples is computed by combining literal similarities, instance similarities of neighbors and concept similarity. The approach leads to a 19% higher precision than PARIS on a dataset extracted from YAGO and DBpedia.

As we have seen, almost every instance matching system available today only performs matchings between two data sets at the same time. Moreover, the resulting quality of created links is almost perfect with measured similarities often above 0.95. Yet, controlling the quality of *transitive links* on the LOD cloud with a size of over 1000 data sources, is not possible when making pairwise matchings of all data sources. To our knowledge, the matching system LINDA is the only system available that is able to perform matchings between arbitrary numbers of sources at the same time [6]. Unfortunately, the solution is not available for evaluating also transitive links. But in contrast to the nearly perfect results of other systems, LINDA pays the price with a practical precision of only 0.83 on a real-world data set comprising 350 million triples from several hundreds of Linked Data sources. For higher recalls the precision even decreases to 0.66, i.e. every third link is incorrect.

3.1 Evaluation of Systems

For measuring the performance of instance matching systems with regard to quality, in 2009, the Ontology Alignment Evaluation Initiative (OAEI) created a benchmark for Linked Data interlinking on entity level [12]. The benchmark of 2013 consists of five test cases, where links between two RDF data sets have to be created: For value transformation, structure transformation, language transformation and two test cases where the transformations have been combined [14]. The data itself comprises 1744 RDF triples describing 430 persons extracted from DBpedia. Each entity has none, one or several matching partners in the data set to be matched. Four instance matching systems have been evaluated on this data: RiMOM [20], SLINT+, LogMAP [17], LilyIOM . Most outstanding are the results for RiMOM and SLINT+ with an average precision and recall of over 0.90 in all five test cases.

Is Instance Matching ready for reliable data interlinking? To answer this question, we have performed an analysis of instance matching on real-world data extracted from 5 LOD stores in a pre-study [16]. In this work, we have analyzed the matching quality of SLINT+ on 90,000 entities from the domains persons, movies, drugs, and organizations by analyzing random samples from the result set. Indeed, the quality at a similarity threshold of 0.95 was high with a precision of 0.91 and even at lower thresholds SLINT+ still 67% of the created links were correct. By computing the transitive closure, 2,055 additional links at a threshold of 0.95 have been created. However, only every fifth of these transitive links was correct. The end-to-end quality of a join on SLINT+'s results is 77%. Using these links to perform a join on the data, would result in an end-to-end quality of under 0.50. In this paper, we have shown that even though the quality of instance matching systems is very high, the quality of transitive extremely low.

In addition to the problems with the systems, the OAEI benchmark does not meet the challenges of large-scale Linked Data instance matching, since it only comprises data from two sources. Because no transitive links are possible there, the Chinese Whispers phenomenon cannot be observed.

But not only the number of sources in the existing benchmark is problematic, also the restriction to one domain, the small number of triples and the automatic data transformation following strict criteria are not reflecting the real-world interlinkage task.

3.2 Building the Benchmark

In this paper, we have revised the dataset from our pre-study to build a large-scale real-world data benchmark for instance matching that meets the aforementioned requirements (available for download[4]). We present a benchmark that is built on a mix of general and highly specific data coming from Freebase[5], DBpedia[6], YAGO[7], New York Times[8], Linked Movie Data Base (LinkedMDB)[9], Drugbank and KEGG (both from the bio2rdf project[10]), and consists of over 110,000 instances spread over the five domains person, movie, book, organization and drug. Whereas the first four data sources are thematically strongly related, the drugs serves as a contrast to emphasize the heterogeneity of web data. As another difficulty, movies are often adaptations from books, which in turn are written by other persons. Because many instance matching systems heavily rely on string-based similarity measures, this interaction of data makes it very hard to distinguish and match the different instances correctly. Furthermore, data from seven different data stores allows analyzing the Chinese Whispers phenomenon in more detail.

Source	Target	Type of Link
DBpedia	YAGO	Wikipedia
DBpedia	Freebase	owl:sameAs
DBpedia	KEGG	dbpedia:kegg
DBpedia	Drugbank	Chemical Abstract Service Number
Drugbank	KEGG	drugbank_vocabulary:xref
NYTimes	Freebase	owl:sameAs
NYTimes	DBpedia	owl:sameAs
LinkedMDB	Freebase	foaf:page

Table 1: List of existing link types for creating the benchmark

To accomplish the task of building a benchmark, a lot of manual work had to be done. We started with extracting URIs for entities belonging to one of the five domains using their respective *rdf:type*. The overlap between the data stores was maximized, by extracting the URIs ordered alphabetically by their *rdfs:label*. For each of these URIs, every RDF triple having the URI as a subject was extracted, resulting in a total number of about 5.5 million triples.

[4] http://www.ifis.cs.tu-bs.de/staff/jan-kalo/benchmark

[5] http://freebase.com

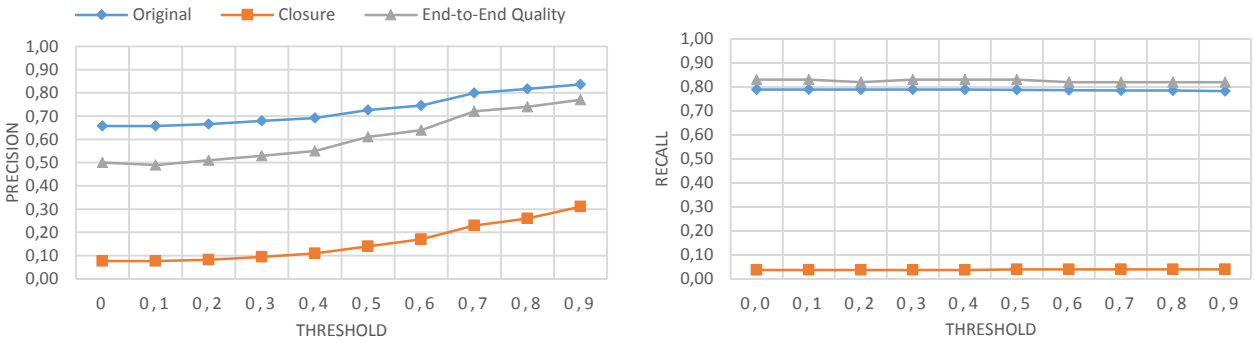
[6] http://dbpedia.org

[7] http://yago-knowledge.org

[8] http://data.nytimes.com

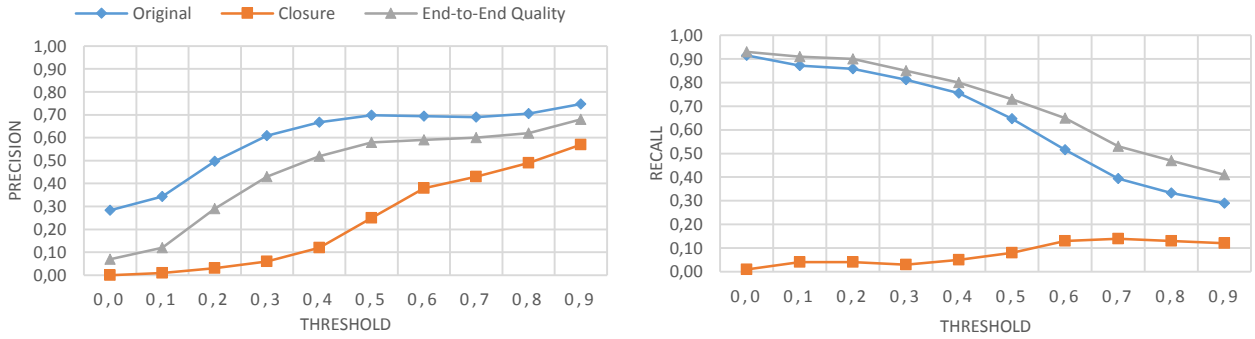
[9] http://linkedmdb.org

[10] http://bio2rdf.org



(a) (b)

Figure 2: Precision (a) and Recall (b) for PARIS



(a) (b)

Figure 3: Precision (a) and Recall (b) for SLINT+.

In order to create the gold standard, we started with exploring existing links between the data stores. We chose reliable and already existing link sources (see Table 1) and extracted 19,000 links. These existing links are *owl:sameAs* links, *foaf:page* identity links from LinkedMDB, the Chemical Abstract Service Number for linking chemical substances like drugs, Wikipedia to interlink YAGO and DBpedia and some data store specific identity links.

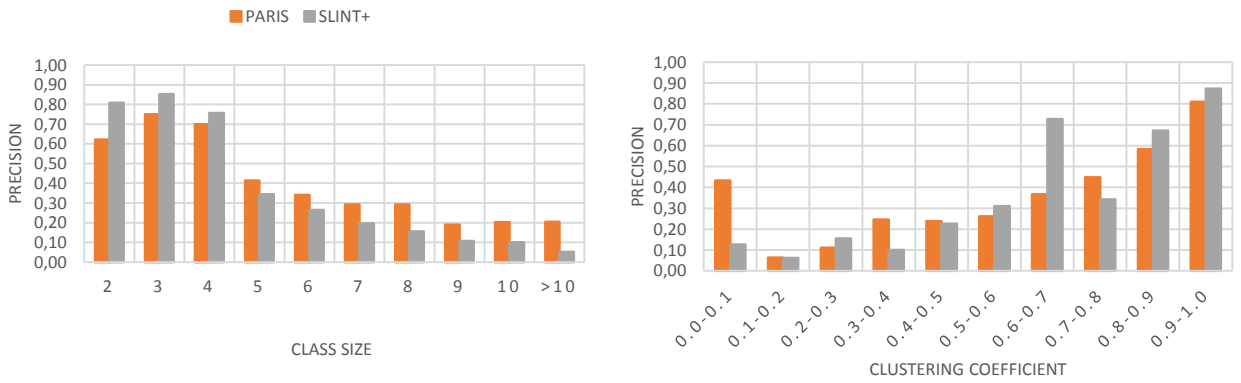
For assuring their quality, we took random samples of size 200 from each of the eight existing link sources. By manually inspecting the data, we verified that no link in the sample sets has been incorrect. During this manual check, no border case, where it was difficult to make the right decision about the correctness of an identity link was found. Consequently, the possibility of creating false positives in this step is very small.

Instances that were linked to more than one instance in another single data store have been removed. In that way duplicate URIs could be removed from the benchmark data. Furthermore, we tested the data for duplicates by performing instance matching with PARIS and SLINT+ on the data sets themselves. That way, we could identify another 29 duplicate instances within the data. We believe, that after these steps all synonyms and duplicate instances have been removed from the data. As a consequence, we can be sure that every instance can have at most one corresponding instance in another data set which allows us to exclude the possibility of false negatives much easier.

In the next step, we computed the transitive closure for the existing 19,000 links. In this step another, 13,000 additional links have been discovered. The correctness of these links was manually inspected taking a random sample of size 200. All of these 200 links could be clearly identified as correct.

In the final step, we had to identify missing links that have not been found transitively. We demonstrate our approach with an example: Links between the data stores Freebase and DBpedia have been found using exiting owl:sameAs links. To align DBpedia and YAGO, we proceeded similar to Suchanek et al. in [26]. Both data sources use Wikipedia identifiers as URIs, so the matching can be trivially computed. But the transitively identified links between YAGO and Freebase are correct, but not necessarily complete: If there are corresponding instances within Freebase and YAGO, but not in DBpedia, the corresponding identity link, cannot be found transitively. Therefore, we manually matched all instances that have not been linked yet in these two datasets. To minimize the manual workload, we did only match instances from one domain to instances from the corresponding domain in the other data store. Furthermore, for every domain and data store, we manually chose a set of attributes that were automatically compared, to further reduce the number of manual comparisons. For example, we did not compare persons with totally different names or birthdates. The manual comparisons have been performed for every pair of the 7 datasets that has been linked transitively. In total, 123 additional correct links could be found in this step.

After these steps, the number of false negatives in our benchmark data should be minimal. To access the quality of the existing links, we again took a random sample of size 900 out of the 30,633 links which all have been identified as correct.



(a) (b)

Figure 4: (a) Histogram of the precision per equivalence class. (b) Histogram of precision for different clustering coefficients.

3.3 Analysis

The created benchmark consists of 30,633 *owl:sameAs* links between 7 different LOD stores. With the help of the new benchmark, we were able to perform an analysis of two state-of-the-art instance matching systems (SLINT+ and PARIS) with regard to their end-to-end quality. Unfortunately no other instance matching system has been publicly available. But since both systems have outperformed several other systems, we believe that they are good representatives of instance matching systems.

SLINT+ did return 98,799 and PARIS did return 105,168 links with similarity values between 0 and 1. At first, the quality of these links is evaluated for similarity thresholds from 0.0 to 0.9 (Figure 2(a) and 3(a)). We can observe that both systems have achieved a precision of over 0.75 for the highest threshold. Looking back at the evaluations on the OAEI benchmark, we observe an about 15% lower precision on this real-world data set. Also the recall is not even near the 100%, with 78% for PARIS and only 29% for SLINT for results with a similarity value above 0.9 (Figure 2(b) and 3(b)). Focusing on the end-to-end quality of joins, we have evaluated the transitively created links. The transitive closure for SLINT+ has a size of over 300,000 additional links, whereas PARIS has created 14,000. However, many of those transitive links from SLINT+ are incorrect: For high similarity values ($\Theta = 0.9$) these links have a precision of 0.57 for SLINT and 0.31 for PARIS, see Figure 2(a) and 3(a). Combining these results with the explicitly created links to obtain the end-to-end quality of joins, we present the results in Figure 2(a) and 3(a): The precision at $\Theta = 0.9$ is 0.77 for PARIS, which is 7% worse than for the explicitly created results. In comparison to these results, for SLINT+ we also report a decrease of 7%. The quality of the transitive links and therefore also the end-to-end quality for joins on Linked Data is low compared to *owl:sameAs* links that has been output by the instance matching systems.

Our results show that even the explicitly created links by the instance matching systems are about 15% worse than the measured results from the OAEI benchmark. The end-to-end quality of joins is even worse, with having a precision of under 80%. With these results instance matching is far from being reliable for interlinking the entire LOD cloud.

To get ideas for improving the end-to-end quality of the equivalence classes, we performed a structural analysis. The experiments have been performed for a similarity threshold of 0.50. For other thresholds similar results can be observed.

In a first experiment, we have analyzed the end-to-end quality of equivalence classes by their size, see Figure 4(a). The evaluation shows a precision of over 0.60 for the small equivalence classes. The maximum quality is obtained for classes of size 3. With increasing size, the precision is decreasing. Classes with more than 10 instances have a precision below 0.20. The results show that large equivalence classes lack of quality.

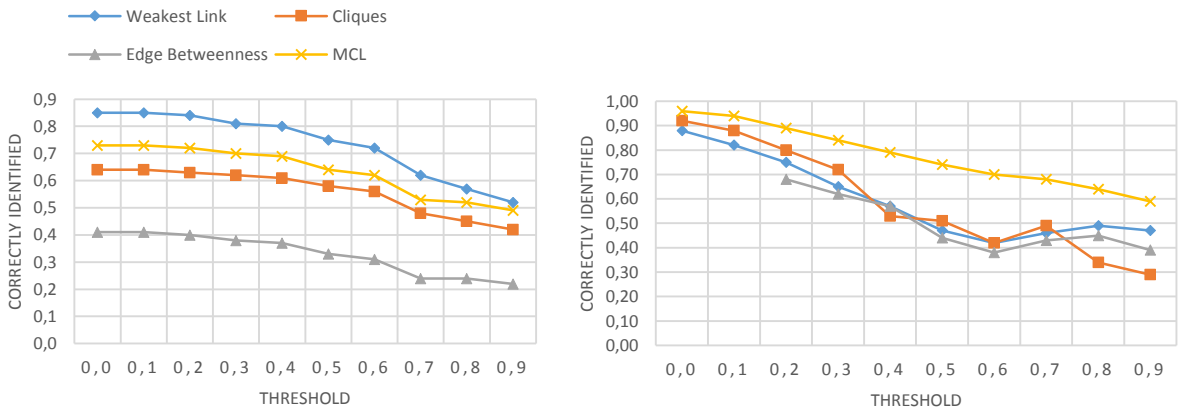
In a second experiment, we evaluated the correlation of the connectedness of an equivalence class and its end-to-end quality. For each equivalence class, we have computed the clustering coefficient, a measure between 0 and 1 which denotes the degree the instances of a class tend to cluster together. We analyzed intervals of size intervals of size 0.1 starting with all equivalence classes with clustering coefficients between 0.0 to 0.1 up to equivalence classes with coefficients between 0.9 and 1.0. For SLINT+ and PARIS, the precision of well-connected classes is above 0.9 (Figure 4(b)). The lower the clustering coefficient of the equivalence classes, the lower its precision. Classes with clustering coefficient between 0.0 and 0.1 have extremely low end-to-end join quality with a precision of under 20%. This experiment has shown highly connected classes consist of high quality links, thus most of the contained instances correspond to the same entity. Looking at it the other way around: Instances of the same entity indeed form highly connected groups.

Consequently, approaches for avoiding Chinese Whispers should identify groups describing the same entity, having a high clustering coefficient, and cut all links between these groups to avoid Chinese Whispers and facilitate high quality entity-centric joins on Linked Data.

4. AVOIDING CHINESE WHISPERS

Benchmarking two state-of-the-art instance matching systems has confirmed our assumption: The end-to-end quality of joins comprising more than only two data stores is significantly worse than isolated evaluations of instance matching systems in literature might suggest. For our benchmark with only 7 out of over thousand LOD stores, the end-to-end quality for both systems is far from being good having a measured precision of under 80%.

To avoid the Chinese Whispers effect, we have to identify wrong *owl:sameAs* links and remove them. As an example, the equivalence class illustrated in Figure 1 has an end-to-end quality of 0.15. The removal of any incorrect link, would significantly improve the end-to-end quality But also removing a correct link can improve the quality, since it can prevent incorrect transitive links.



(a) (b)

Figure 5: Proportion of correct identified links for PARIS in (a) and for SLINT+ in (b).

Removing the correct link between H and G would remove several incorrect links (all links between the component on the left and on the right in Figure 6).

In this section, we present four approaches that work on equivalence classes that have been created by an instance matching system. Each of the approaches tries to identify incorrect links in an equivalence class using the structure and the similarity values that have been computed by the matching systems. Later on these links can be removed to improve the end-to-end quality. The first approach tries to identify links by using the similarity values, the three other approaches try to find correct groups of instances in equivalence classes using clustering-like techniques.

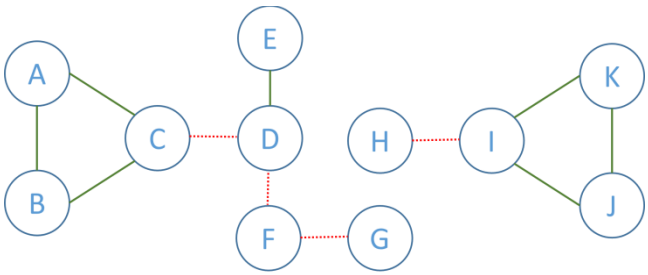


Figure 6: Example from Figure 1, where a correct link has been removed.

4.1 Weakest Links

This first approach tries to identify wrong links by only considering the similarity values that have been output by the instance matching systems. We assume that a similarity value between two instances expresses the resemblance of two instances. Given two links with their similarity value that have been output by an instance matching system ($\text{sim}(A,B) = 0.80$ and $\text{sim}(C,D) = 0.90$), it is more likely that the link between A and B is incorrect, than that the link between C and D is incorrect. With regard to this approach, we get equivalence classes with low quality (only classes with size above 3 and clustering coefficient smaller than 0.9) as an input and remove links with lowest similarity value until the class is split into two or more components.

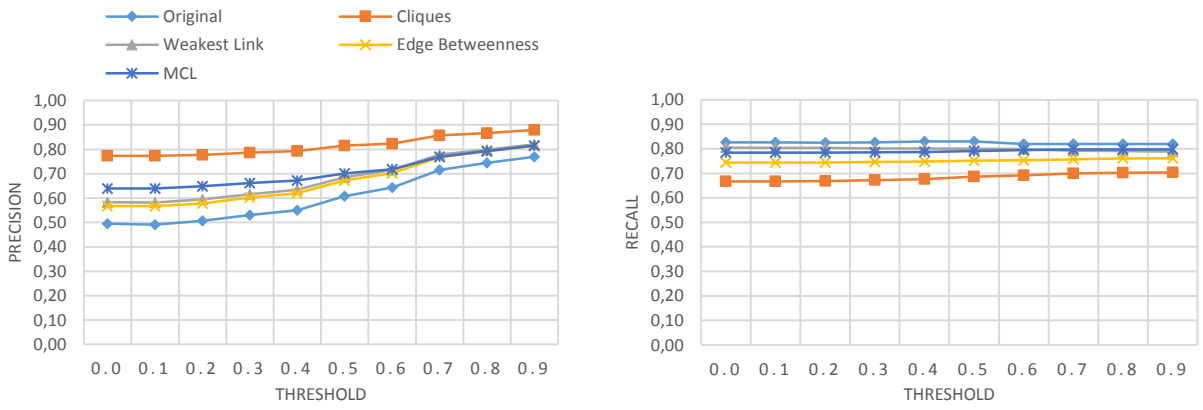
4.2 Edge Betweenness

As a more advanced approach, we take the structure of the equivalence classes into consideration. A high number of links within a group of instances, is a strong indication that these instances are representing the same entity (for example the group A, B, C in Figure 6). We believe that identifying groups within an equivalence class that are better intraconnected than interconnected and removing every other link can improve the end-to-end quality. In a graph theoretical sense these groups are called communities in a graph. Detecting communities has been heavily researched by Girvan and Newman[13, 23]. Their approach (Girvan-Newman algorithm) relies on the identification of links (or edges in a graph) that are between communities, but are not part of a community itself. By removing these links, an equivalence class can be partitioned into highly connected components. To find the edge that is most between two communities, the so called Edge Betweenness of a link is computed, by counting the number of shortest paths in the graph that runs through this link. If only a few number of edges connect highly connected communities, shortest paths between such communities, all go along the few connections. Therefore, their edge betweenness is higher than the ones of links within a community. By alternately removing the link with the highest edge betweenness, then recomputing the edge betweenness values, a graph can be partitioned into communities.

We use the same assumptions already used for the weakest link approach. We only apply the edge betweenness approach to equivalence classes with low quality and remove links until the class is split into two or more components.

4.3 Cliques

The Clique approach is similar to the previous approach, since it also tries to identify groups of instances that are highly intraconnected. Similar to complete-link clustering, for this approach, we try to identify groups, where the distance between all of the contained points (instances) is small enough. With regard to instance matching, these groups are required to only contain instances, which pairwise similarity above the similarity threshold. Compared to the edge betweenness approach, this approach is much stricter. Every detected community is a complete graph with a clustering coefficient of 1.0, hence is highly intraconnected. Given the results of some instance matching system, this approach removes all links from an equivalence class if it does not belong to a complete sub-component of size minimum 3. Sub-components of size 2 are not considered, because that would obviously be valid for every link in an equivalence class. But for classes that already have size 2, no link is removed.



(a) (b)

Figure 7: Precision (a) and Recall (b) for the four approaches compared to the original output of PARIS.

4.4 Markov Clustering

Another approach for detecting groups of instances of the same entity is Markov Clustering (MCL) [10]. MCL has already performed well on duplicate detection [15], without tuning the parameters. It is an algorithm for detecting dense regions in weighted graphs by simulating random walks. In other words, it also detects communities in a graph, but without giving the number of communities as an input. MCL uses a stochastic matrix which corresponds to the equivalence class which then is transformed by alternating two simple algebraic operations. Expansion corresponds to squaring the matrix, which simulates a random walk. Random walks tend to end in the same dense region of the graph they started. Thus, expansion increases the probabilities of intra-cluster edges. Inflation corresponds to taking the entry wise power of the matrix and normalizing the resulting matrix to be stochastic again. This step boosts edges with high probability values and damps edges with low probabilities. At the bottom line, edges within dense regions are favored over the edges between these clusters. After alternating these steps for some iterations (in practice 3-10 iterations), the matrix converges. The resulting matrix corresponds to a clustering of nodes (instances) to dense regions. The granularity of the resulting clusters can be steered by varying the inflation power. Higher inflation parameters lead to a higher number of clusters. Our evaluation has shown that high inflation parameters lead to better precision on the instance matching results, but that increasing the parameter above 8.0 only slightly changes the outcome of the clustering. Because of that, we performed the cluster process with an inflation parameter of 8.0.

5. EVALUATION

The approaches presented in this paper have the objective to improve the quality of end-to-end joins on entity-centric Linked Data queries which rely on *owl:sameAs* links between entities across different data stores. For evaluation, we let two state-of-the-art instance matching systems PARIS and SLINT+ perform matchings on our benchmark. Taking the results of the systems as the input, the approaches identify incorrect links that are then removed from the result set. For similarity thresholds from 0.0 to 0.9, the performance of the presented approaches is evaluated.

5.1 Identification of Incorrect Links

The main goal of any approach for avoiding Chinese Whispers should be improving the end-to-end quality for joins. Obviously, this increase of precision, can be achieved by reducing the number of incorrect links in an equivalence class. In this subsection, we evaluate the performance of our 4 approaches for identifying wrong links correctly.

In this first experiment, we present the percentage of correct links out of all removed links that have been removed by the four approaches, see Figure 5. We can observe that for PARIS the weakest link approach performs best. For the lowest threshold more than 90% of the removed links have been incorrect. However, most of the links at those threshold are incorrect, so the performance is not remarkably good. For the high thresholds, the weakest link approach has identified 52% of the removed links correctly as being incorrect. The MCL and Clique approach has worked under 10% worse at the highest thresholds, with results around 50%. The edge betweenness approach has identified 22% of the removed links correctly. With regard to SLINT+, the MCL approach has performed best. 59% of the removed links has been correctly identified as being wrong. The three other approaches have identified less than 50% of the removed links correctly for links with similarity above 0.90.

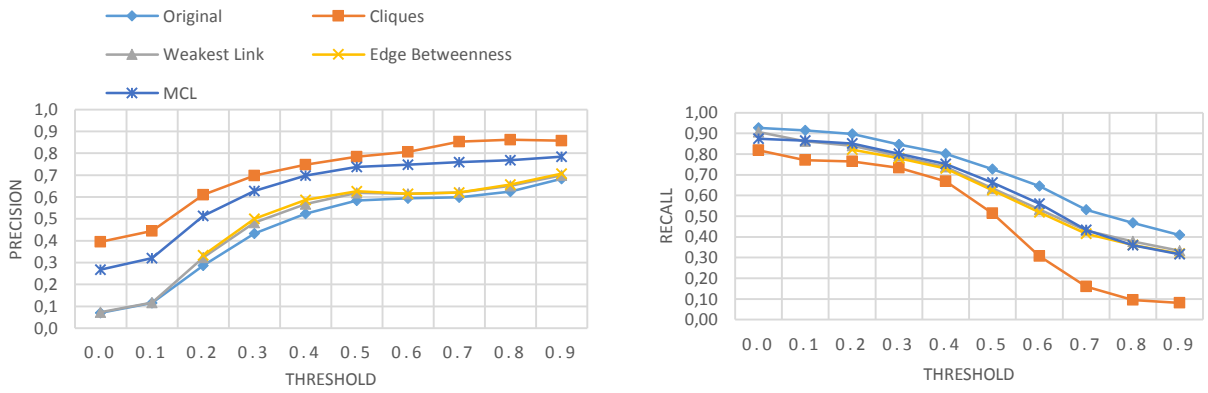
Each approach seems to perform best for low similarity thresholds, but obviously the number of incorrect links in the input data when considering links with low similarity is huge. Therefore, identifying 90% of the removed links correctly, when most of the input links are incorrect is not a remarkable achievement. More interesting are the results for high thresholds. Here, none of the systems has performed remarkably good. None of the presented approaches has identified more than 60% of the removed links correctly. That does not imply that the approaches cannot improve the end-to-end quality, but it implies that they will remove a large proportion of correct *owl:sameAs* links. Nevertheless, a large improvement of the join quality is possible.

5.2 Improving the End-to-End Quality

In this last evaluation, we finally measure the End-to-End quality after applying the presented approaches.

For each approach the results of the instance matching systems is provided as the input. After removing links from this result set, we evaluate the end-to-end quality and recall of each approach using the created ground truth.

The results for PARIS are presented in Figure 7. In Figure 7(a) we can observe that the precision for all approaches compared to the original quality of PARIS is improved. For the lowest similarity threshold the precision is improved from 0.50 to 0.77, at high



(a) (b)

Figure 8: Precision (a) and Recall (b) for the four approaches compared to the original output of SLINT+.

similarity thresholds still an increase from 0.77 to 0.88 can be measured. The highest quality results have been achieved by the Clique approach. However, this approach has also been the most restrictive which resulted in the highest loss in recall, see Figure 7(b).

SLINT+'s results are presented in Figure 8. The highest increase in quality can be measured for the Clique approach: The end-to-end quality of 0.07 at the lowest threshold has been improved up to 0.40 with a small loss in recall. For $\Theta = 0.90$, the original precision is at 0.70. The Clique approach increases the precision to 0.86. The evaluation of the recall is totally different to the results of PARIS. The loss in recall using the Clique approach at high similarity thresholds decreases from 0.41 to 0.08. All other approaches have a recall about 0.30. Markov Clustering has caused the smallest loss in recall with still achieving the second best results with regard to precision: The recall is decreases from 0.41 to 0.32, but the precision is improved from 0.70 to 0.78 for $\Theta = 0.90$.

We observe that the Clique approach lead to the best results in terms of precision, but also to the biggest loss in recall. The end-to-end quality of joins on links created by instance matching systems is improved by more than 10% from under 0.80 to nearly 0.90. We believe that using a less precision oriented approach, like the MCL algorithm, is the way to improve the instance matching problem. Its loss in recall is significantly smaller than for the Clique algorithm, but it still improves the precision of the resulting links by more than 5%. At higher recall values, the precision can be improved up to 30%.

The results of our experiments show that using techniques for identifying groups of instances belonging to the same entity, indeed improve the end-to-end quality by avoiding Chinese Whispers by over 10%. An interesting observation is that the algorithms perform well for every similarity threshold and for both instance matching systems. Since other state-of-the-art systems (SIGMa and ARIA) use similar approaches for matching as PARIS does, we believe that their performance can also be improved by our presented algorithms.

6. CONCLUSION

One of the biggest advantages of Linked Open Data, is the possibility to use information from various information sources like querying one single database. Unfortunately, this interlinkage of heterogeneous data sources has become the biggest challenge for realizing this vision. Especially creating identity links on entity level by creating new *owl:sameAs* has been heavily researched in the last years. Several instance matching systems, all achieving very good results in benchmarks, have been developed. State-of-the art systems are even achieving precision and recall percentages in the high 90ies on the standard instance matching benchmark used at OAEI.

But as our evaluation clearly shows, these systems unfortunately perform much worse on the real-world interlinking tasks comprising several data sources from the LOD cloud, because they cannot avoid the Chinese Whispers effect. Even for our small test dataset comprising only 7 different data stores, the systems created links which lead to an end-to-end quality of joins of under 80%. This gap between matching precision and end-to-end join precision is obviously destroying a vital facet of LOD's usefulness and thus endangering the Semantic Web's success at large

We have shown how even a small number of incorrect links can result in a vastly deteriorated quality during LOD query processing when performing joins. We also proved that structural clustering approaches, like presented in this paper, can be used to identify at least some central incorrect links in the existing data to avoid Chinese Whispers. But as long as everyone is allowed to create *owl:sameAs* links without any restrictions, a central quality control is hardly possible.

With the growing size of the LOD cloud, creating links and maintaining a high quality interlinkage is bound to get more and more difficult. Since our methods to avoid the Chinese Whispers phenomenon can improve the overall link quality of any instance matching system, they can also be used when interlinking data over several sources. The presented methods can improve even high quality results of state-of-the-art matching systems with a precision of around 0.80 by over 10% and thus at least boost precision into the 90% region. We are confident, that these approaches have to be incorporated in any modern instance matching systems to improve the resulting end-to-end matching quality.

7. REFERENCES

[1] Araujo, S. and Schwabe, D. 2012. SERIMI : Class-based Disambiguation for Effective Instance Matching over Heterogeneous Web Data Categories and Subject Descriptors. *WebDB*. (2012), 25–30.

[2] Berners-Lee, T. and Fischetti, M.B.-D.M.L. 2000. Weaving the Web: The Original Design and Ultimate

Destiny of the World Wide Web by Its Inventor. (May 2000).

[3] Berners-Lee, T., Hendler, J. and Lassila, O. 2001. The Semantic Web. *Scientific American*. 284, 5 (May 2001), 34–43.

[4] Bizer, C., Heath, T., Kingdom, U. and Berners-lee, T. 2009. Linked Data - The Story So Far. *International journal on semantic web and information systems*. 5, 3 (2009), 1–22.

[5] Bizer, Chris and Heath, Tom and Ayers, Danny and Raimond, Y. 2007. Interlinking open data on the web. *4th European Semantic Web Conference*. (2007).

[6] Böhm, C., de Melo, G., Naumann, F. and Weikum, G. 2012. LINDA: Distributed Web-of-Data-Scale Entity Matching Christoph. *Proceedings of the 21st ACM international conference on Information and knowledge management - CIKM '12* (New York, New York, USA, Oct. 2012), 2104.

[7] Cruz, I.F., Antonelli, F.P. and Stroe, C. 2009. AgreementMaker: efficient matching for large real-world schemas and ontologies. *Proceedings of the VLDB Endowment*. 2, 2 (2009), 1586–1589.

[8] Ding, L., Shinavier, J., Finin, T. and McGuinness, D.L. 2010. owl: sameAs and Linked Data: An empirical study. *WebSci10: Extending the Frontiers of Society On-Line*. (2010).

[9] Ding, L., Shinavier, J., Shangguan, Z. and McGuinness, D.L. 2010. SameAs networks and beyond: analyzing deployment status and implications of owl: sameAs in linked data. *ISWC 2010*. (2010), 145–160.

[10] Dongen, S.M. van 2001. *Graph clustering by flow simulation*.

[11] Elmagarmid, A.K., Ipeirotis, P.G., Verykios, V.S. and Member, S. 2007. Duplicate Record Detection: A Survey. *IEEE Transactions on Knowledge and Data Engineering*. 19, 1 (Jan. 2007), 1–16.

[12] Ferrara, A., Lorusso, D., Montanelli, S. and Varese, G. 2008. Towards a benchmark for instance matching. *ISWC 2008*. (2008), 37.

[13] Girvan, M. and Newman, M.E.J. 2002. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*. 99, 12 (2002), 7821–7826.

[14] Grau, B., Dragisic, Z., Eckert, K. and Euzenat, J. 2013. Results of the Ontology Alignment Evaluation Initiative 2013. *Proc. 8th ISWC workshop on ontology matching (OM)*. (2013), 61–100.

[15] Hassanzadeh, O., Chiang, F., Lee, H.C. and Miller, R.J. 2009. Framework for evaluating clustering algorithms in duplicate detection. *Proceedings of the VLDB Endowment*. 2, 1 (Aug. 2009), 1282–1293.

[16] Homoceanu, S., Kalo, J. and Balke, W. 2014. Putting Instance Matching to the Test: Is Instance Matching Ready for Reliable Data Linking? *Foundations of Intelligent Systems*. (2014), 274–284.

[17] Jiménez-Ruiz, E. and Grau, B.C. 2011. LogMap: logic-based and scalable ontology matching. *ISWC 2011*. (Oct. 2011), 273–288.

[18] Lacoste-Julien, S., Palla, K., Davies, A., Kasneci, G., Graepel, T. and Ghahramani, Z. 2013. Sigma: Simple greedy matching for aligning large knowledge bases. *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, New York, USA, Aug. 2013), 572.

[19] Lee, S. and Hwang, S. 2014. ARIA: Asymmetry Resistant Instance Alignment. *AAAI Conference on Artificial Intelligence*. (2014), 94–100.

[20] Li, J., Tang, J., Li, Y. and Luo, Q. 2009. Rimom: A dynamic multistrategy ontology alignment framework. *IEEE Transactions on Knowledge and Data Engineering*. 21, 8 (2009), 1218–1232.

[21] Luce, R.D. 1956. Semiorders and a theory of utility discrimination. *Econometrica, Journal of the Econometric Society*. 24, 2 (1956), 178–191.

[22] Melo, G. De 2013. Not Quite the Same: Identity Constraints for the Web of Linked Data. *AAAI Conference on Artificial Intelligence*. (2013), 1092–1098.

[23] Newman, M. and Girvan, M. 2004. Finding and evaluating community structure in networks. *Physical review E*. (2004), 1–15.

[24] Nguyen, K., Ichise, R. and Le, B. 2012. SLINT: a schema-independent linked data interlinking system. *Ontology Matching*. (2012), 1–12.

[25] Niu, X., Rong, S., Zhang, Y. and Wang, H. 2011. Zhishi. links results for OAEI 2011. *Proceedings of the Sixth International Workshop on Ontology Matching*. (2011), 220–227.

[26] Suchanek, F., Abiteboul, S. and Senellart, P. 2011. Paris: Probabilistic alignment of relations, instances, and schema. *Proceedings of the VLDB Endowment*. 5, 3 (2011), 157–168.