A Clustering Approach for Holistic Link Discovery (Project overview)

Markus Nentwig, Anika Groß, and Erhard Rahm

Database Group, Department of Computer Science, University of Leipzig

Abstract. Pairwise link discovery approaches for the Web of Data do not scale to many sources thereby limiting the potential for data integration. We thus propose a holistic approach for linking many data sources based on a clustering of entities representing the same real-world object. Our clustering approach utilizes existing links and can deal with entities of different semantic types. The approach is able to identify errors in existing links and can find numerous additional links. An initial evaluation on real-world linked data shows the effectiveness of the proposed holistic entity matching.

1 Introduction

Linking entities between sources has been a major effort in recent years to support data integration in the so-called Web of Data. A large number of tools for semi-automatic link discovery has been developed to facilitate the generation of new links (mostly of type `owl:sameAs`) [5]. Repositories such as BioPortal [7] or LinkLion [6] collect numerous links for many sources to improve their availability and re-usability without having to repeatedly determine such links for new applications and use cases.

Despite the advances made, there are significant limitations in the achieved inter-linking of data sources and in the current approaches for link discovery. First, the degree of inter-linking is still low and automatically generated links are wrong in many cases [2]. Current approaches for link discovery only match two data sources at a time (pairwise linking) resulting in a poor scalability to many sources [8]. This is because the number of pairwise mappings increases quadratically with the number of sources, e.g., one would need almost 20,000 mappings to fully interconnect 200 sources.

Most of the current link discovery approaches process only two data sources at a time restricting scalability for many sources while few approaches natively support multiple data sources. In [3] the quality of joins on Linked Open Data is improved by determining highly connected entity groups in a set of given links using metrics such as edge betweenness. The joint entity matching approach in [1] aims at finding links between multiple data sources based on an iteratively adopted matrix of pairwise similarity values. Existing approaches to determine `owl:sameAs` links also focus on entities of the same type while many sources contain entities of different types (bibliographic datasets contain publication and author entities, geographical datasets contain numerous kinds of entities such as countries, lakes, etc.). Furthermore, existing links are hardly utilized when additional links need to be determined.

The need for holistic approaches to integrate many data sources has been outlined in [8] with the suggestion to use clustering-based approaches to link and fuse matching

entities for improved scalability. We are working on such clustering-based approaches for the Web of Data [4] and summarize the approach and initial evaluation results in this short project overview. The approach utilizes already existing links and supports the integration of entities of different semantic types. All matching entities from different sources are grouped into a single cluster thereby supporting a much more compact representation of match results than with binary links. Furthermore, the cluster-based approach facilitates the integration of additional sources and entities since they only need to be matched with the set of already existing clusters rather than adopting a pairwise linking with numerous different sources.

We consider a set of k data sources containing entities of different types. Each entity e is referenced by an URI and has a set of describing semantic properties (i.e., RDF vocabulary). Two entities of different sources can be connected by a `owl:sameAs` link if they were found to represent the same real-world object. All same-as links between two sources S_i and S_j ($1 \leq i, j \leq k$) constitute a binary equivalence mapping $M_{i,j} = \{(e_1, e_2, sim) | e_1 \in S_i, e_2 \in S_j, sim[0,1], i \neq j\}$. Link discovery tools can assign a similarity value sim to indicate the strength of a connection with 1 denoting equality (highest similarity). For k data sources, there can be up to $\frac{k \cdot (k-1)}{2}$ such equivalence mappings. For holistic entity clustering, we use a set of existing mappings $\mathcal{M} = \bigcup_{i,j=1}^{k} M_{i,j}$ and the set of associated entities $\mathcal{E}$ of the k data sources as input. The goal is to compute a set of n clusters $\mathcal{C} = \{c_1^r, \ldots, c_n^r\}$ such that each cluster only includes matching entities (denoting the same real-world object) and that different clusters represent different entities. In this paper, we consider duplicate-free data sources, such that a cluster can contain at most k entities. For each cluster we determine a cluster *representative* r derived from the cluster entities to simplify the comparison between clusters.

The following Sec. 2 will describe and illustrate the workflow for the proposed holistic entity clustering. We then present preliminary evaluation results in Sec. 3 and conclude.

2 Holistic Clustering

Our holistic clustering approach utilizing existing links consists of four main steps: preprocessing, initial clustering based on connected components, cluster splitting and iterative cluster merging. We illustrate the approach in Fig. 1 for partially linked geographical entities from four data sources. Due to space restrictions, linked entities (and corresponding properties) are shortened to their IDs and clusters are represented by thick bordered boxes. While our algorithm is generic, it can be customized to specific domains by providing appropriate background knowledge, similarity functions and thresholds to determine relevant entities and clusters. For the considered geographical domain, the similarity function determines a combined similarity from the string (trigram) similarity on normalized labels, the similarity of the semantic entity type and the normalized geographical distance. The details of the workflow will be described in the rest of this section.

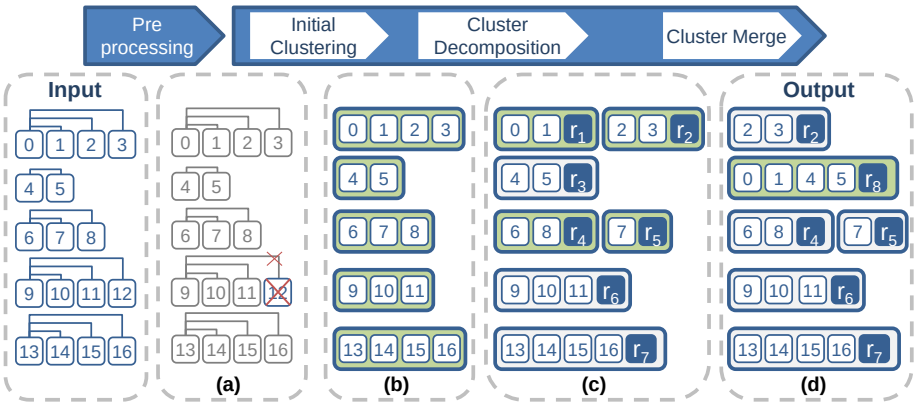


Fig. 1: Application of holistic clustering to running example.

2.1 Preprocessing

During preprocessing we normalize property values needed for the similarity computation, i.e., we simplify entity labels, harmonize information about the semantic types of entities and check that the input mappings do not violate the assumption of duplicate-free data sources.

Information about the semantic type of entities differs substantially between sources or may be missing. For instance, DBpedia uses *City* and *Town* whereas Freebase has a type *citytown* and other related types. To overcome such differences, we use background knowledge about the equivalence and comparability of entity types of different sources to harmonize the type information. We manually determined this type mapping for our geographical sources although it could be constructed with the help of ontology matching approaches. Based on the type mapping we simplified numerous types to more general ones, e.g., the types *city* or *suburb* are treated as type *Settlement*. After harmonizing the type information, we remove all links where the linked entities have incompatible types. Note that we do not exclude links to entities with missing type information.

With the assumption of duplicate-free data sources in place we check if all input mappings comply with the restriction. In Fig. 1, entities `11` and `12` come from the same source so that the links (`9-11`) and (`9-12`) violate the 1:1 assumption. In such cases, we only keep the best-matching link (`9-11`) and drop weaker links (`9-12`) as shown in Fig. 1a.

2.2 Initial Clustering

Using the preprocessed entities and mappings we first identify a set of initial clusters by computing all connected components as the transitive closure from the given links. Each resulting connected component builds an initial cluster C covering all entities that are directly or indirectly connected via a same-as link in $\mathcal{M}$. In our running example, we create five different clusters covering 2-4 entities (see Fig. 1 b).

2.3 Cluster Decomposition

The initially created clusters can contain entities that should actually be separated, e.g., due to wrong input links or because of an insufficiently high transitive similarity between entities. For this reason we decompose clusters (1) based on incompatible semantic types and (2) exclusion of entities based on intra-cluster similarity values. Finally, for each resulting cluster a cluster representative is created.

Type-based Grouping: While we eliminate links with incompatible semantic types during preprocessing, there can be entities without type information (e.g., entity `(0)`) leading to clusters with entities of different types during the initial clustering. We split such clusters into several smaller sub-clusters with entities of the same type. Entities without semantic types are then added to the sub-cluster of their most similar neighbor using computed similarities between cluster members. For the considered cluster of our example, we first build sub-clusters `(2,3)` and the singleton cluster `(1)` of different types (e.g. *Settlement* vs. *BodyOfWater*). The untyped entity `(0)` is assigned to the cluster with the more similar (geographically closer) entity `(1)` resulting in sub-cluster `(0,1)`.

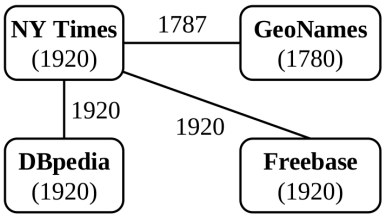
Similarity-based Refinement: We further split clusters based on the computed intra-cluster similarity between entities For each entity, we determine the average similarity of its links to other cluster members and separate an entity if the average similarity is below a given threshold t_s. This process is executed iteratively as long as the average similarity of an entity is smaller than t_s. In the merge phase, such separated entities may be added to other more similar clusters. As shown in Fig. 1 c we separate entity `(7)` from the cluster `(6,7,8)` since the entity had a low label similarity to `(6)` and `(8)`.

Cluster Representative: For each resulting cluster we create a cluster representative to (1) facilitate the computation of inter-cluster similarities in the merge step and (2) to efficiently match new entities, e.g., from additional data sources. We create the representative by combining the properties from all entities in a cluster and select a preferred value for each property, e.g., based on a majority consensus, the maximal length of labels or pre-determined source priorities (for geo-coordinates). We also keep track of the data sources represented in the cluster to avoid unnecessary merges for already covered data sources.

2.4 Cluster Merge

Lastly we merge similar clusters below the maximal possible cluster size k. Therefore we determine the similarity between clusters by applying the domain-specific similarity function on the cluster representatives. Given the typically large number of clusters, this is an expensive operation if we consider all pairs of clusters (quadratic complexity). We avoid unneeded comparisons by not considering pairs of clusters with incompatible types, overlapping data sources and clusters with $> k$ resulting elements. The cluster mapping $\mathcal{CM}$ computed for the remaining cluster pairs is restricted to the most similar pairs of clusters with a similarity exceeding the merge similarity threshold t_m.

(a)



(b)

cluster size	initial clustering	decomposition type-based	decomposition sim-based	cluster merge
1	-	50	174	159
2	-	115	153	154
3	140	180	228	229
4	1780	1680	1594	1597

Fig. 2: (a) Data set structure: number of entities and links, (b) Cluster sizes in workflow phases.

Cluster merging is an iterative process that continues as long as there are merge candidates in $\mathcal{CM}$. In each iteration we select the pair of clusters (c_1, c_2) with the highest similarity from $\mathcal{CM}$, merge it into a new cluster c_m and compute a new representative for it. c_1 and c_2 are removed from $\mathcal{C}$ and appropriate cluster pairs are removed from $\mathcal{CM}$. Furthermore, c_m is added to $\mathcal{C}$ and $\mathcal{CM}$ is extended by similar cluster pairs for the new cluster c_m obeying the restriction for new cluster pairs. The termination of the loop and the merge step is guaranteed since we reduce the number of clusters in each iteration. Applying the approach to our example leads to the merging of `(0,1,`r_1`)` and `(4,5,`r_3`)` into the new cluster `(0,1,4,5,`r_8`)` (see Fig. 1 c,d) due to a high similarity of all properties.

For the given example, we clustered entities from four different data sources thereby finding previously unknown links and eliminating wrong existing links for improved data quality. The six resulting clusters (Fig. 1 d) implicitly represent 17 pairwise entity links compared to 12 initially given links from which 3 turned out to be incorrect. In particular, we could now identify matches between previously unconnected sources.

3 Initial Evaluation

We evaluate our holistic clustering approach using the location subset of the OAEI 2011 Instance Matching benchmark with links of presumed high quality. Fig. 2 a shows the number of links between the four geographical data sources and the number of entities that are interconnected by these links. We retrieved additional entity properties via SPARQL endpoints or REST APIs in the respective sources in 2015. Still, the geo-coordinates were missing for 1009 entities (13.4%) and the type information even for 2525 entities (33.5%), including all entities from the NY Times dataset. We use the similarity function described in Sec. 2; the similarity thresholds t_s, t_m are set to 0.7.

We first evaluate the resulting cluster sizes for the different phases of our holistic clustering approach applied to these datasets (Fig. 2 b). During the preprocessing (not shown in the Fig.), we already removed seven wrong NYT-GeoNames links based on the one-to-one cardinality restriction; the missing type information for NYT did not allow removal of type-incompatible links during preprocessing. The initial clustering results only in clusters of sizes 3 and 4 since each NYT entity is linked with an entity in Freebase and DBpedia. Applying the type-based grouping and similarity-based re-

finement results in a significant number of cluster splits and clusters of size 1 and 2 due to incompatible entity types and partially low intra-cluster similarity. During the merge phase some of the smaller clusters can be merged into larger ones leading to more clusters of sizes 3 and 4. In particular, 15 singleton clusters could be merged into clusters of size 2 and 3. Overall, the resulting clusters represent 9510 links with 4596 new links and 713 deleted links compared to the input link set. In particular, we could cluster many entities from the previously unconnected sources GeoNames, DBpedia and Freebase.

4 Conclusion

We proposed a new holistic approach for clustering-based link discovery for many data sources. The approach utilizes existing links and can match entities of different semantic types. The determined entity clusters facilitate the integration of more data sources without having to individually link them to each other data source. An initial evaluation for linked data from the geographical domain confirmed that the new approach holds great promise as it can identify wrong links and many additional links even between previously unconnected sources. In the future, we will evaluate the scalability and quality of our approach on larger datasets and more sources from different domains based on a parallel Hadoop-based implementation that is currently under development. We will also study specific aspects such as improving the quality of current mapping collections like BioPortal and the incremental extension of entity clusters when integrating new data sources.

References

1. Christoph Böhm, Gerard de Melo, Felix Naumann, and Gerhard Weikum. LINDA: Distributed Web-of-Data-Scale Entity Matching. In *Proc. of the 21st ACM CIKM*, pages 2104–2108. ACM, 2012.
2. Daniel Faria, Ernesto Jiménez-Ruiz, Catia Pesquita, Emanuel Santos, and Francisco M Couto. Towards annotating potential incoherences in BioPortal mappings. In *The Semantic Web–ISWC 2014*, pages 17–32. Springer, 2014.
3. Jan-Christoph Kalo, Silviu Homoceanu, Jewgeni Rose, and Wolf-Tilo Balke. Avoiding chinese whispers: Controlling end-to-end join quality in linked open data stores. In *ACM Web Science 2015*, 2015.
4. Markus Nentwig, Anika Groß, Axel-Cyrille Ngonga Ngomo, and Erhard Rahm. Holistic Entity Clustering for Linked Data. Technical report, 2016. submitted for publication.
5. Markus Nentwig, Michael Hartung, Axel-Cyrille Ngonga Ngomo, and Erhard Rahm. A Survey of Current Link Discovery Frameworks. *Semantic Web J.*, 2016.
6. Markus Nentwig, Tommaso Soru, Axel-Cyrille Ngonga Ngonga Ngomo, and Erhard Rahm. LinkLion: A Link Repository for the Web of Data. In *ESWC 2014 Posters & Demo session*.
7. Natalya F Noy, Nigam H Shah, Patricia L Whetzel, Benjamin Dai, Michael Dorf, et al. BioPortal: ontologies and integrated data resources at the click of a mouse. *Nucleic Acids Res*, 37:W170–W173, 2009.
8. Erhard Rahm. The Case for Holistic Data Integration. In *Proc. ADBIS*. Springer LNCS, 2016.