Scalable and Domain-Independent Entity Coreference: Establishing High Quality Data Linkages across Heterogeneous Data Sources*

Dezhao Song

Department of Computer Science and Engineering, Lehigh University
19 Memorial Drive West, Bethlehem, PA 18015, USA
des308@cse.lehigh.edu

Abstract. Due to the decentralized nature of the Semantic Web, the same real world entity may be described in various data sources and assigned syntactically distinct identifiers. In order to facilitate data utilization in the Semantic Web, without compromising the freedom of people to publish their data, one critical problem is to appropriately interlink such heterogeneous data. This interlinking process can also be referred to as *Entity Coreference*, i.e., finding which identifiers refer to the same real world entity. This proposal will investigate algorithms to solve this entity coreference problem in the Semantic Web in several aspects. The essence of entity coreference is to compute the similarity of instance pairs. Given the diversity of domains of existing datasets, it is important that an entity coreference algorithm be able to achieve good precision and recall across domains represented in various ways. Furthermore, in order to scale to large datasets, an algorithm should be able to intelligently select what information to utilize for comparison and determine whether to compare a pair of instances to reduce the overall complexity. Finally, appropriate evaluation strategies need to be chosen to verify the effectiveness of the algorithms.

Keywords: Entity Coreference, Linked Data, Domain-Independence, Scalability, Candidate Selection, Pruning.

1 Introduction, Challenges and Expected Contributions

Linked Data [3], which encourages the sharing of data and publishing of links to other datasets, has reached an impressive size: 295 datasets with about 31 billion triples and 500 million links across these datasets[1]. Since the same real world entity (e.g., people, locations, etc.) may be described by more than one data source with syntactically distinct identifiers, the biggest benefit of Linked Data is to enable people to walk from one dataset to others by following the linkages in order to obtain a relatively comprehensive view of the entities of interest.

To really facilitate the utilization of this large-scale and decentralized Linked Data, one critical problem is how to appropriately interlink such heterogeneous

* Advisor: Professor Jeff Heflin.

[1] http://www4.wiwiss.fu-berlin.de/lodcloud/state

P. Cudré-Mauroux et al. (Eds.): ISWC 2012, Part II, LNCS 7650, pp. 424–432, 2012.

data with automated approaches. This interlinking problem has been well studied in Databases (*Record Linkage*) and Natural Language Processing (*Entity Coreference*) to find out which identifiers refer to the same real world entity. In this paper, we use the term *Entity Coreference* to refer to the process of finding ontology instances that describe the same real world entity in the Semantic Web.

Challenges. First of all, in order to detect coreferent instances precisely and comprehensively, it is important to locate and utilize the relevant information (the context) of the instances appropriately. Various situations can mislead the entity coreference results, such as name variations, the use of abbreviations, misspellings, etc. Also, the collected data may come from heterogeneous data sources and may be incomplete. To ensure the quality of the generated links, an entity coreference algorithm needs to address such challenges appropriately.

Making this context selection and utilization process domain-independent is equally important. A domain refers to the category (e.g., People, Geographic, etc.) and the usage (e.g., academic people, politics, etc.) of the data. In the past, domain-specific techniques have successfully helped to achieve good entity coreference results, e.g., relying on matching person names to identify coreferent person instances. However, when considering various domains, humans may lack the knowledge or time to specify what information to utilize and thus coreference tools are less likely to be available for all domains end users deal with.

Furthermore, scalability needs to be taken into account when designing entity coreference algorithms. Considering the scale of Linked Data, approaches that perform a brute-force comparison on every pair of instances [1, 16] are less likely to succeed. As a key part of this proposal, we will explore novel approaches to scaling entity coreference on large datasets: Candidate selection (*CS*) and context pruning (*CP*), i.e., doing fewer comparisons vs. doing faster comparisons. *CS* selects instance pairs that are likely to be coreferent in a lightweight manner and we only apply the expensive entity coreference algorithms on selected candidates. The key point of *CP* techniques is to compare an appropriately selected portion of the context to speed up the comparison for a single pair of instances.

Contributions. We propose to develop scalable and domain-independent algorithms for precisely and comprehensively detecting coreferent ontology instances from heterogeneous data sources with the following contributions:

- Developing mechanisms for automatically collecting and weighting context information of ontology instances in a domain-independent manner;
- Developing algorithms for detecting coreferent instances based upon the collected context, achieving precision and recall comparable to that of the state-of-the-art across various domains (e.g., >90% precision and recall);
- Devising techniques to link datasets without discriminative labels by exploring how to appropriately combine individually non-discriminating predicates;
- Developing effective pruning algorithms on the context of ontology instances in order to speed up the computation for a single pair of instances;
- Devising lightweight and domain-independent candidate selection algorithms targeting BTC-scale datasets (billions of triples and 400 million instances

[9]). Furthermore, the coreference results should not be affected much by applying such pruning techniques (e.g., 1-2% lower F1-score).

2 Related Work

Entity Coreference. The system by Aswani et al. [1] needs to interact with search engines to retrieve context information and thus may not scale to large datasets. RiMOM [23] and Silk [20] rely on human provided matching rules and thus costly to customize to new domains. RiMOM matches instances by comparing their property value pairs with Edit distance or the Vector Space model; to the best of our knowledge, it requires domain configuration to assign property weights. Silk is a general framework for users to specify rules for matching instances, but it may be difficult for users to specify such rules for all domains. Compared to these systems, we try to reduce the need of human input in developing entity coreference systems.

Hu et al. [7] build a kernel by adopting the formal semantics of the Semantic Web that is then extended iteratively in terms of discriminative property-value pairs in the descriptions of URIs. Algorithms that combine formal semantics of the Semantic Web and string matching techniques also include Zhishi.me [11], LN2R [15], CODI [12] and ASMOV [8]. These systems can be applied to datasets in different domains without human provided matching rules, such as People, Location, Organization and Restaurant. One disadvantage of reasoning based approaches is that they highly depend on the correct expressions of the ontologies. For example, as reported by the developers of the ASMOV system, in some dataset, the *surname* property was declared to be functional, yet if a person takes a spouses name, they will have different surnames for data collected at different times. According to our current experiments, our proposed algorithm is able to outperform several of these state-of-the-art systems on some benchmark datasets; however, further experiments are needed for a more comprehensive comparison on more diverse datasets.

Candidate Selection. Candidate selection selects instance pairs that are likely to be coreferent to reduce the overall complexity. ASN [26] relies on human input for identifying a candidate selection key; but sufficient domain expertise may not be available for various domains. Supervised [10] or partially-supervised [4] approaches have been explored to learn the candidate selection key; however, obtaining a sufficiently-sized groundtruth data is impractical for large datasets. Compared to these systems, our proposed candidate selection algorithm is unsupervised and is able to automatically learn the candidate selection key.

Indexing techniques have also been well-adopted for candidate selection [5]. PPJoin+ [25] adopts a positional filtering principle that exploits the ordering of tokens in a record. EdJoin [24] employs filtering methods that explore the locations and contents of mismatching n-grams. BiTrieJoin [21] is a trie-based method to support efficient edit similarity joins with sub-trie pruning. FastJoin [22] adopts fuzzy matching techniques that consider both token and character level similarity. Similar algorithms also include AllPairs [2] and IndexChunk

[14]. Although our proposed candidate selection algorithm also adopts indexing techniques, a secondary filtering on the looked-up candidates from the index significantly reduces the size of the final candidate set.

3 Research Accomplished

In this section, we present the core idea of each accomplished work.

Exhaustive Pairwise Entity Coreference based on Weighted Neighborhood Graph (EPWNG). *EPWNG* detects coreferent ontology instances by computing the similarity for every instance pair between datasets based on a set of paths (the context, Fig. 1) [16, 18]. $path = (x, P_1, N_1, ..., P_n, N_n)$, where x is an ontology instance; N_i and P_i are any expanded RDF node and predicate in the path. Each node N_i has a weight W_i computed based on the discriminability of its associated predicate P_i and the path weight is the multiplication of all its node weights. *EPWNG* compares the comparable paths in the context of two instances x and y. For each path m of x, we find the path n from y that is comparable and has the highest string similarity to m. We call the similarity between m and n the path score; the average path weight of m and n is treated as the weight of this score. This process is repeated for every path of x and the weighted average on such (path score, path weight) pairs is computed as the final similarity score for x and y. Here, two paths are comparable if their predicates at corresponding positions are comparable, i.e., having the same semantics. E.g., predicate *CiteSeer:name* is comparable to *DBLP:name*. Although the mapping axioms of predicate comparability were manually created in our experiments, they can also be automatically derived from ontology alignment systems [13].

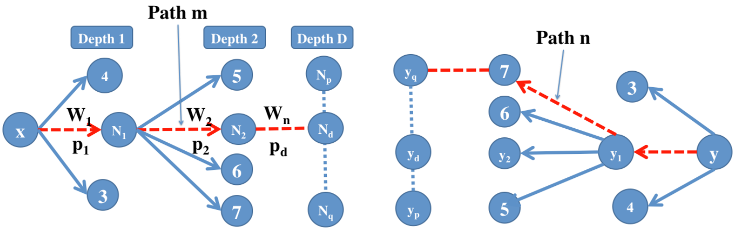

Fig. 1. Weighted Neighborhood Graph (G)

If we assume that multiple heterogeneous sources contain n instances in total, and that the context graphs have branching factor b and depth d, then the time complexity of *EPWNG* is $O(n^2 b^{2d})$, making it prohibitively expensive for dealing with large contexts and datasets.

Context Pruning. Given the complexity of *EPWNG*, one question is: *Can we speed up the comparison on a single instance pair by only considering the context that could potentially make a significant contribution to their final similarity score, i.e., reducing the impact of the branching factor* b? Therefore, we propose a sampling based context pruning technique [19]. Instead of actually computing the string similarity between the last nodes of all pairs of comparable paths of two instances, we estimate how similar a node could be to another (the potential contribution) with a small sample from the entire dataset. When comparing two instances, before computing the end node similarity, we estimate if the potential contribution of the rest of the context would enable the similarity of two instances to go above a threshold. If so, we continue processing the remaining context; otherwise, we simply stop to save computational cost. Given that performing an estimation takes time itself, we further design a utility function to judge if it is worth estimating, which additionally provides 10% runtime savings.

Candidate Selection (CS). We further propose a candidate selection technique to reduce the impact of number of instances [17]. Ideally, a candidate selection algorithm should be able to automatically determine what information to utilize to select candidates, cover as many true matches as possible, and also select fewest pairs to help to scale the entire entity coreference process. Our proposed algorithm selects candidate instance pairs by computing a similarity on their character bigrams extracted from discriminating literal values that are chosen using domain-independent unsupervised learning. With unsupervised learning, we learn a set of datatype properties as the candidate selection key that both discriminates and covers the instances well. We then utilize the object values of such predicates for candidate selection. Instances are indexed on these object values to enable efficient look-up for similar instances. This algorithm has been shown to possess the properties discussed above on datasets in several domains (People, Publications, Hotel and Restaurant) with up to 1M instances.

4 Evaluation and Preliminary Results

Metrics. The standard metrics for evaluating entity coreference algorithms include: *Precision*: the number of correctly detected pairs divided by the total number of detected pairs; *Recall*: the number of correctly detected pairs divided by the number of coreferent pairs according to the groundtruth; and their *F1-score* calculated as $2*\frac{Precision*Recall}{Precision+Recall}$. Since it could be difficult to obtain perfect groundtruth for large datasets, sampled precision (*sP*) and relative recall (*relR*) could be adopted. *relR* is calculated as $\frac{|correctly\ detected\ pairs\ from\ one\ system|}{|correctly\ detected\ pairs\ from\ all\ systems|}$; to measure *sP*, we can manually check the correctness of a subset of the detected links. The idea of *wisdom of the crowd* can be adopted for assessing precision while having perfect groundtruth to measure recall could still be difficult.

For candidate selection, Reduction Ratio (RR)=1-$\frac{|candidate\ set|}{N*M}$, Pairwise Completeness (PC)=$\frac{|true\ matches\ in\ candidate\ set|}{|true\ matches|}$, and their F1-score ($F_{cs}$) [10, 26] are three commonly used metrics. N and M are the size of two instance sets that are matched to one another. PC evaluates how many true positives are returned by an algorithm, RR is the degree to which it reduces the number of comparisons needed, and F_{cs} gives a comprehensive view of how well a system performs. Finally, runtime is an important metric for evaluating both types of systems.

Since the size of groundtruth and $N*M$ in RR may not be at the same order of magnitude, the calculated numbers of RR, PC and F_{cs} might not indicate the actual differences of two systems appropriately. Particularly, when applied to large datasets, a large change in the size of the candidate set may only be reflected by a small change in RR due to its large denominator. Thus, in addition to evaluating candidate selection results with RR, PC and F_{cs}, we could apply an actual entity coreference algorithm to the selected candidates to measure the precision and recall of the final coreference results and the overall runtime.

Datasets. The Ontology Alignment Evaluation Initiative (OAEI) provides benchmark datasets for evaluating entity coreference systems. DBpedia, New York Times, Freebase, RKB and SWAT[2] are all suitable datasets as well. Finally, the entire LOD should be perfect for testing entity coreference algorithms.

Preliminary Results. In Table 1, *EPWNG* outperforms a few other coreference algorithms on three datasets (<2K instances) from OAEI2010 (left); compared to state-of-the-art candidate selection systems on 100K instances (right), *CS* enables the entire coreference process to run the fastest with the best coreference F1-scores. To further demonstrate and improve the domain-independence of *EPWNG* and *CS*, we will apply them to other diverse datasets from OAEI, including Location, Organization and Medicine.

Table 1. Evaluating Against State-of-the-Art Systems

Dataset	System	P(%)	R(%)	$F1$(%)
Person1	EPWNG [18]	**100**	**100**	**100**
	RiMOM [23]	**100**	**100**	**100**
	ObjectCoref [7]	**100**	99.8	99.9
	LN2R [15]	**100**	**100**	**100**
	CODI [12]	87	96	91
Person2	EPWNG [18]	98.52	**99.75**	**99.13**
	RiMOM [23]	95.2	99	97.1
	ObjectCoref [7]	**100**	90	94.7
	LN2R [15]	99.4	88.25	93
	CODI [12]	83	22	36
Restaurant	EPWNG [18]	74.58	98.88	**85.02**
	RiMOM [23]	**86**	76.8	81.1
	LN2R [15]	75.67	75	75.3
	CODI [12]	71	72	72

Dataset	System	F_{cs}	*Coref F1* (%)	*Time* (s)
RKB Person	CS [17]	**99.68**	**93.63**	**12.25**
	AllPairs [2]	99.36	92.52	83.76
	PPJoin+ [25]	99.36	92.52	82.96
	EdJoin [24]	99.59	92.84	63.31
SWAT Person	CS [17]	99.32	94.90	**12.63**
	AllPairs [2]	99.52	**94.99**	108.34
	PPJoin+ [25]	99.52	**94.99**	106.72
	EdJoin [24]	**99.59**	94.94	102.77
RKB Pub	CS [17]	**99.99**	**99.74**	**15.05**
	AllPairs [2]	99.02	99.27	340.14
	PPJoin+ [25]	99.02	99.27	342.21
	EdJoin [24]	97.97	98.90	1330.20

5 Proposed Research

On-the-Fly Candidate Selection. Instead of pre-selecting candidate pairs, we are exploring candidate selection techniques at runtime. Consider that during the entity coreference process, an instance is compared to many other instances; the results of these prior comparisons could be useful in determining whether two instances might be coreferent. At any point in time, each instance should then have a *Matching History*, i.e., a set of other instances that it is somewhat similar to. One hypothesis is that two coreferent instances should share a sufficient amount of common instances in their histories. Therefore, the intuition of this on-the-fly candidate selection idea is that before actually computing the similarity

[2] `http://swat.cse.lehigh.edu/resources/data`

for an instance pair with expensive techniques, it might be worthwhile to spend a little effort to examine if their histories are similar enough for filtering purposes. Furthermore, as we process more instances, more true matches should be covered, thus it might make sense to gradually increase the threshold on such similarity of instances' matching histories to better balance F1-score and runtime.

Towards Linking the Entire Linked Open Data (LOD). With the goal of being able to handle the entire LOD, we will explore the following problems.

First, when handling the entire LOD, automated methods are needed to determine predicate comparability. As an alternative to complex ontology alignment systems, one idea is to determine predicate comparability based upon their value space, such that predicates with similar value spaces are comparable. Take the *fullname* predicate as an example. Rather than treating the names on their whole as values, tokens or n-grams can be extracted to form the value space.

Furthermore, given that LOD covers datasets from various domains, one might imagine how would the coreference results of one type of instances impact the others. For example, academic publications and researchers are generally correlated in academic datasets. Suppose we start from publications (since titles are generally very discriminating), could we then be able to achieve higher recall on matching person data by being able to provide better hints for person instance pairs with non-discriminative names (due to abbreviation, misspelling, etc.) but sharing coreferent publication instances (represented with syntactically distinct URIs) in their context? One step further, could we come up with approaches to automatically prioritize the domains to process, i.e., determining which domains should be processed first so that the other domains could benefit most? For scalability reasons, we could start with the existing linkages in the most influential domain instead of detecting everything from scratch. Since the existing links in LOD are of questionable quality [6], a lightweight verification step might be needed to firstly check the correctness of such links.

Last but not least, in prior work [16–19], the data we try to integrate generally contains some discriminative labels, e.g., names for people, hotel and restaurant and titles for publications. The question is what if we try to address domains that lack such discriminating labels? Maybe all predicates would then have relatively the same weight and thus *EPWNG* erroneously treats every piece of information the same? Or maybe all datatype properties will be selected for candidate selection and therefore no reduction will be achieved by having to deal with every single triple? One preliminary idea to handling non-discriminative data is to combine values from multiple properties, expecting the combined values could be more discriminating than that of any individual property.

References

1. Aswani, N., Bontcheva, K., Cunningham, H.: Mining Information for Instance Unification. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 329–342. Springer, Heidelberg (2006)

2. Bayardo, R.J., Ma, Y., Srikant, R.: Scaling up all pairs similarity search. In: Proceedings of the 16th International Conference on World Wide Web (WWW), pp. 131–140 (2007)
3. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data - the story so far. Int. J. Semantic Web Inf. Syst. 5(3), 1–22 (2009)
4. Cao, Y., Chen, Z., Zhu, J., Yue, P., Lin, C.Y., Yu, Y.: Leveraging unlabeled data to scale blocking for record linkage. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI), pp. 2211–2217 (2011)
5. Christen, P.: A survey of indexing techniques for scalable record linkage and deduplication. IEEE Transactions on Knowledge and Data Engineering, TKDE (2011)
6. Halpin, H., Hayes, P.J., McCusker, J.P., McGuinness, D.L., Thompson, H.S.: When owl:sameAs Isn't the Same: An Analysis of Identity 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. 305–320. Springer, Heidelberg (2010)
7. Hu, W., Chen, J., Qu, Y.: A self-training approach for resolving object coreference on the semantic web. In: Proceedings of the 20th International Conference on World Wide Web (WWW), pp. 87–96 (2011)
8. Jean-Mary, Y.R., Shironoshita, E.P., Kabuka, M.R.: Ontology matching with semantic verification. Journal of Web Semantics 7(3), 235–251 (2009)
9. Khatchadourian, S., Consens, M.P.: Understanding billions of triples with usage summaries. In: Semantic Web Challenge (2011)
10. Michelson, M., Knoblock, C.A.: Creating relational data from unstructured and ungrammatical data sources. J. Artif. Intell. Res. 31, 543–590 (2008)
11. Niu, X., Sun, X., Wang, H., Rong, S., Qi, G., Yu, Y.: Zhishi.me - Weaving Chinese Linking Open Data. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part II. LNCS, vol. 7032, pp. 205–220. Springer, Heidelberg (2011)
12. Noessner, J., Niepert, M., Meilicke, C., Stuckenschmidt, H.: Leveraging Terminological Structure for Object Reconciliation. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010, Part II. LNCS, vol. 6089, pp. 334–348. Springer, Heidelberg (2010)
13. Pavel, S., Euzenat, J.: Ontology matching: State of the art and future challenges. IEEE Transactions on Knowledge and Data Engineering, TKDE (2011)
14. Qin, J., Wang, W., Lu, Y., Xiao, C., Lin, X.: Efficient exact edit similarity query processing with the asymmetric signature scheme. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 1033–1044 (2011)
15. Saïs, F., Pernelle, N., Rousset, M.-C.: Combining a Logical and a Numerical Method for Data Reconciliation. In: Spaccapietra, S. (ed.) Journal on Data Semantics XII. LNCS, vol. 5480, pp. 66–94. Springer, Heidelberg (2009)
16. Song, D., Heflin, J.: Domain-independent entity coreference in RDF graphs. In: Proceedings of the 19th ACM Conference on Information and Knowledge Management (CIKM), pp. 1821–1824 (2010)
17. Song, D., Heflin, J.: Automatically Generating Data Linkages Using a Domain-Independent Candidate Selection Approach. 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. 649–664. Springer, Heidelberg (2011)
18. Song, D., Heflin, J.: Domain-independent entity coreference for linking ontology instances. ACM Journal of Data and Information Quality, ACM JDIQ (2012)

19. Song, D., Heflin, J.: A pruning based approach for scalable entity coreference. In: Proceedings of the Twenty-Fourth International Florida Artificial Intelligence Research Society Conference (FLAIRS), pp. 98–103 (2012)
20. 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)
21. Wang, J., Li, G., Feng, J.: Trie-join: Efficient trie-based string similarity joins with edit-distance constraints. PVLDB 3(1), 1219–1230 (2010)
22. Wang, J., Li, G., Feng, J.: Fast-join: An efficient method for fuzzy token matching based string similarity join. In: Proceedings of the 27th International Conference on Data Engineering (ICDE), pp. 458–469 (2011)
23. Wang, Z., Zhang, X., Hou, L., Zhao, Y., Li, J., Qi, Y., Tang, J.: RiMOM results for OAEI 2010. In: Proceedings of the 5th International Workshop on Ontology Matching (2010)
24. Xiao, C., Wang, W., Lin, X.: Ed-join: an efficient algorithm for similarity joins with edit distance constraints. Proc. VLDB Endow. 1(1), 933–944 (2008)
25. Xiao, C., Wang, W., Lin, X., Yu, J.X., Wang, G.: Efficient similarity joins for near-duplicate detection. ACM Trans. Database Syst. 36(3), 15 (2011)
26. Yan, S., Lee, D., Kan, M.Y., Giles, C.L.: Adaptive sorted neighborhood methods for efficient record linkage. In: ACM/IEEE Joint Conference on Digital Libraries (JCDL), pp. 185–194 (2007)