Fria: Fast and Robust Instance Alignment

Sanghoon Lee
POSTECH
Pohang, Republic of Korea
sanghoon@postech.edu

Jongwuk Lee*
The Penn State University
University Park, PA, USA
jxl90@psu.edu

Seung-won Hwang
POSTECH
Pohang, Republic of Korea
swhwang@postech.edu

ABSTRACT

This paper proposes Fria, a fast and robust instance alignment framework across two independently built knowledge bases (KBs). Our objective is two-fold: (1) to design an effective instance similarity measure and (2) to build a fast and robust alignment framework. Specifically, Fria consists of two-phases. Fria first achieves *high-precision* alignment for seed matches which have strong evidence for aligning. To obtain *high-recall* alignment, Fria then divides non-matched instances according to the types identified from seeds, and gives additional chances to the same-typed instances to be matched. Experimental results show that Fria is fast and robust, by achieving comparable accuracy to state-of-the-arts and a 10-times speed up.

Categories and Subject Descriptors

H.2.5 [**Database Management**]: Heterogeneous Databases

Keywords

Instance alignment; entity matching; knowledge base; hierarchical partitioning

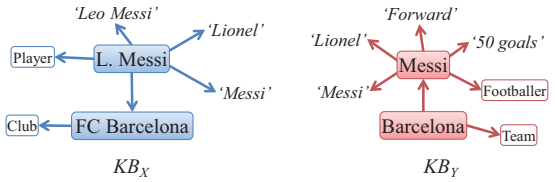
1. INTRODUCTION

An abundance of public linked data, *i.e.*, DBpedia [1] and YAGO [3], poses the challenge of interlinking such data toward an ultimate knowledge base (KB). This problem, known as *instance alignment*, has been actively studied in the literature [4, 5]. PARIS [5] is a holistic approach to aligning relations, concepts and instances in a probabilistic fashion. ObjCoref [4] is a self-training approach using a kernel built from OWL semantics. However, despite a rich body of existing research, there is little work for pursuing both efficiency and robustness.

To achieve this goal, we propose a two-phase framework with a new instance similarity, called Fria. Specifically, the first *high-precision* phase identifies a small seed set with near-perfect precision. The second *high-recall* phase follows to give second chances to false positives, by comparing again to entities of the same types.

*This work was done while the author was at POSTECH.

WWW 2013 Companion, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2038-2/13/05.



(a) Two sub-graphs for instances `L.Messi` and `Messi`

	'Lionel'	***'Messi'***	*'Forward'*	*'50 goals'*
'Lionel'	**1**	0	0	0.125
'Messi'	0	**1**	0	0
'Leo Messi'	0.333	0.556	0.111	0.111

(b) Literal similarities using normalized Levenshtein distance

Figure 1: Computing the similarity score between instances `L.Messi` and `Messi` from KB_X and KB_Y

2. INSTANCE SIMILARITY MEASURE

This section first introduces a new instance similarity measure which is robust for the asymmetry between KBs. The KB refers to a collection of knowledge with instances, literals, and their relationships. To illustrate this, Figure 1(a) depicts two KBs in graphs, where nodes are instances (rectangles) and literals (italics), and edges represent relationships or properties between them.

To design a robust similarity measure, a challenging issue is how to adapt the *asymmetric* structures of KBs. As shown in Figure 1(a), KB_Y is much richer than KB_X for the properties on Messi. In this case, mapping literals and penalizing for unmatched one, *e.g.*, '*50 goals*', may lead to significantly underestimating the similarity of `L.Messi` and `Messi` due to the presence of asymmetry.

To address this problem, we aggregate literal-level mappings with high string similarity scores. Given two instances, we first extract some related literals for each instance, and then choose the literal pairs that represent the same knowledge with one-to-one constraint as discussed in [2]. Among all possible mapping pairs, we thus selectively consider some pairs whose string similarities exceed a certain threshold. Unmatched literals are pruned out so that asymmetric literals do not penalize the overall similarity score of aligning instances. The overall instance similarity can be calculated by the sum of the string similarity of matched literal pairs whose values are greater than the threshold.

Figure 1(b) describes string similarities for all possible literal pairs. When the threshold is set as 0.2, only the literal pairs (bold) about *'Lionel'* and *'Messi'* are matched, and the instance similarity score is thus 2, where asymmetric literals such as '*Forward*' and '*50 goals*' are pruned out.

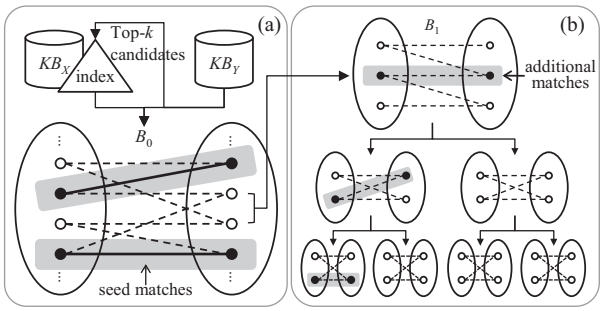


Figure 2: The overview of our proposed framework

3. INSTANCE ALIGNMENT FRAMEWORK

Using the proposed similarity measure, we develop a fast and robust instance alignment framework Fria. The problem is viewed as matching a weighted bipartite graph in which edges represent matching candidates with weights of similarity scores.

To address this problem, Fria consists of two phases. In the first phase, Fria uses a widely adopted abstraction of generating a bipartite graph. In this phase, it can attempt to find the maximal bipartite matching, incurring quadratic time complexity. Instead, Fria is built upon a *k-regular bipartite graph* by connecting only top-k similar instances per each, and then only finds the edges representing mutual top-1 matches (Figure 2a). Because it is much more conservative than general matching, this process is highly efficient with near-perfect precision, but suffers from low recall.

To boost recall, the second phase (Figure 2b) hierarchically partitions non-matched instances into clusters with the same types. The type pairs are mined from seed matches using a feature selection method, *e.g.*, *Pearson's chi-square*. For example, in our example KBs, if `Player` and `Footballer` are the mostly found type pair in seeds, non-matched instances are divided by these types. That is, the instances having `Player` of KB_X and `Footballer` of KB_Y are collected into the same bipartite graph, and they have the second chances to be matched within the subgraph. As illustrated in Figure 2(b), such hierarchical partitioning can be done until no instance matches.

To summarize, Fria can achieve both efficiency and robustness using the two-phase approach that first identifies high-precision seed instances and then collects non-matched instances by types. In particular, in the second phase, hierarchical partitioning is able to collect smaller instance sets to be matched (low running time) without compromising recall.

4. EVALUATION

We evaluated our proposed framework Fria in real-world large-scale KBs, *i.e.*, YAGO [3] and DBpedia [1]. These KBs include millions of high-precision instances. To simulate the alignment for asymmetric instances, we prepared two different sized KBs for DBpedia. Specifically, DBpedia_S is comprised of high-quality refined infobox types and properties, which is same with one used in PARIS [5]. DBpedia_L includes about five times of instances and literals than DBpedia_S. Our experiments were conducted in Java on Intel i7 3.6 GHz CPU and 64 GB RAM.

To evaluate the robustness of our proposed similarity measure, we first computed the similarity scores for every possible instance pair (*i.e.*, a complete bipartite graph), and selected mutually top-1 instance pairs. The threshold is empirically set as 0.9 in this task. We then compared our measure with three other set similarity measures such as Jaccard similarity, Cosine similarity with TF-IDF, and Jensen-Shannon divergence (JSD). Figure 3 depicts precision-recall graphs for each similarity measure. It is clear that our similarity measure outperforms all the other measures over various parameter settings in both tasks.

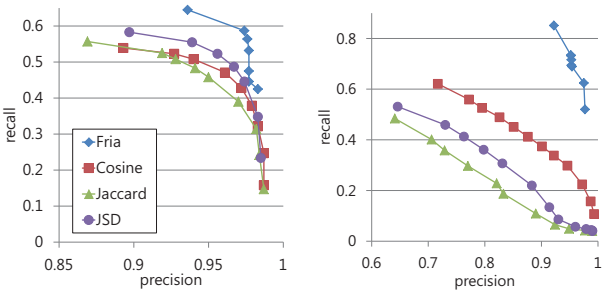


Figure 3: Precision-recall graphs of instance similarity measures of YAGO-DBpedia_S(left) and YAGO-DBpedia_L(right)

Table 1: Instance alignment results for YAGO-DBpedia_S dataset

Framework	Phase	Prec.	Rec.	F1.	Time
Fria	1	0.974	0.587	0.732	35 min
	2	0.904	0.702	0.791	30 min
PARIS	1	0.86	0.69	0.80	4 h
	2	0.89	0.73	0.81	5 h
	3	0.90	0.73	0.81	5 h

We then compared Fria with PARIS for YAGO-DBpedia_S task (as reported in [5]). Fria built a 10-regular bipartite graph in the first phase (*i.e.*, high-precision matching). Table 1 shows instance matching results for each phase. After the second phase (*i.e.*, high-recall matching) was performed, the recall of Fria increased 0.115 without significantly compromising precision (*i.e.*, -0.07). It is clear that Fria achieved comparable accuracy, and was faster than PARIS by one order of magnitude.

Acknowledgements

This research was supported by the MKE(The Ministry of Knowledge Economy), Korea and Microsoft Research, under IT/SW Creative research program supervised by the NIPA(National IT Industry Promotion Agency) (NIPA-2012-H0503-12-1036)

5. REFERENCES

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. DBpedia: a nucleus for a web of open data. In *ISWC/ASWC*, 2007.

[2] J. Gemmell, B. I. P. Rubinstein, and A. K. Chandra. Improving entity resolution with global constraints. *CoRR*, 2011.

[3] J. Hoffart, F. M. Suchanek, K. Berberich, E. Lewis-Kelham, G. de Melo, and G. Weikum. YAGO2: exploring and querying world knowledge in time, space, context, and many languages. In *WWW*, 2011.

[4] W. Hu, J. Chen, and Y. Qu. A self-training approach for resolving object coreference on the semantic web. In *ACM WWW*, 2011.

[5] F. M. Suchanek, S. Abiteboul, and P. Senellart. PARIS: Probabilistic Alignment of Relations, Instances, and Schema. *PVLDB*, 2011.