Exploiting Transitivity for Entity Matching

J. Baas[1], M. M. Dastani[1], and A. J. Feelders[1]

[1] Utrecht University, Heidelberglaan 8, 3584 CS Utrecht, Netherlands
[2] {j.baas,m.m.dastani,a.j.feelders}@uu.nl

Abstract. The goal of entity matching in knowledge graphs is to identify sets of entities that refer to the same real-world object. Methods for entity matching in knowledge graphs, however, produce a collection of pairs of entities claimed to be duplicates. This collection may fail to satisfy transitivity, and hence may fail to represent a valid solution. We show that an ad-hoc enforcement of transitivity on the set of identified entity pairs may decrease precision dramatically. We therefore propose a methodology that starts with a given similarity measure, generates a set of entity pairs, and applies cluster editing to enforce transitivity, leading to overall improved performance.

1 Introduction

Many datasets use different identifies to refer to the same real life entities, or may contain duplicates themselves. Automated methods for identifying and linking duplicate entities, also known as entity matching, in the knowledge graphs are necessary. A considerable difficulty with entity matching is that the total number of possible entity pairs is much larger than the number of actual (duplicate) entity pairs, also known as the problem of skewness [8, 1]. This extreme skewness can cause false positive results to overwhelm the true positives, even for highly accurate classifiers. This has caused many other works to use ranking techniques, and their associated metrics, to sort the possible entity pairs with some similarity measure, where duplicate entity pairs are expected to appear on top of the ranking [6, 9, 3, 7]. Other works, such as Saeedi et al. [5], perform blocking in the first stages to reduce the number of pairs that are evaluated.

The identified set of pairs are generally required to satisfy some structural properties, in particular transitivity. However, taking the transitive closure of the entity pairs identified by entity matching techniques may not work as this may possibly conclude many spurious entity pairs. We propose the application of cluster editing for entity matching and set up a number of experiments to evaluate our proposal. We show that compared to an ad-hoc enforcement of transitive closure on identified pairs, our approach always results fewer distinct entity pairs (i.e. they have a higher precision) while retaining duplicate entity pairs (i.e. recall is not lowered). The experiments are performed on semi-synthetic datasets that are generated by introducing duplicates in an existing dataset in a controlled manner. This results in a range of different cluster distributions, where we measure the effects of the number of clusters and different cluster sizes.

2 Applying Cluster Editing on Matched Entities

Approach: An overview of our overall method is given in Fig. 1. We start with an embedding of a set of entities E, some of which may be duplicates, and use Euclidean distance to measure their proximity (panel A of Fig. 1). Let $N_k(i)$ denote the index set of the k nearest neighbors of e_i. For each entity e_i, we make k candidate pairs (e_i, e_j), $j \in N_k(i)$, thereby addressing skewness by ruling out the vast majority of pairs. The dotted lines in panel B illustrate the candidate pairs for $k = 1$. Moreover, we assume that a (small) subset of these candidate pairs is labeled by a domain expert (blue lines in panel B). The labeled pairs are used to train a probabilistic classifier. This classifier is used to determine, for each candidate pair (e_i, e_j), the fitted probability p_{ij} that e_i and e_j are duplicates. Depending on the features used by the classifier, and its complexity, the fitted probabilities need not be proportional to the distance between entities.We do however assume that the features used by the classifier are symmetric so that $p_{ij} = p_{ji}$, and therefore we can indeed regard a pair of entities as unordered. We then use a cut-off value θ so that if $p_{ij} > \theta$, then e_i and e_j are predicted to be duplicates (panel C). This "raw outcome" of the pairwise classifier may however violate the transitivity constraint. Obviously, an ad-hoc application of transitive closure to the links predicted by the classifier never removes any links, but can only add new links (panel G). This may result many spurious entity pairs. A more principled method to restore transitivity is to use the cluster editing technique [2]. Here, we compute a weight $w(i,j) = \log(\frac{p_{ij}}{1-p_{ij}}) - \log(\frac{\theta}{1-\theta})$ for each pair of entities (e_i, e_j) within the same connected component (regardless of whether it is a candidate pair or not), such that $w(i,j)$ is positive if $p_{ij} > \theta$, and negative otherwise (panel D). If $w(i,j)$ is positive (negative), a link between i and j is provisionally assumed to be present (absent). The resulting set of links may however again violate the transitivity constraint. Cluster editing is used to restore transitivity by adding and/or removing links in such a way that the total score $\sum_{(i,j)} w(i,j)x_{ij}$ is maximized, where $x_{ij} = 1$ if a link between i and j is present in the solution, and $x_{ij} = 0$ otherwise (panels E and F).

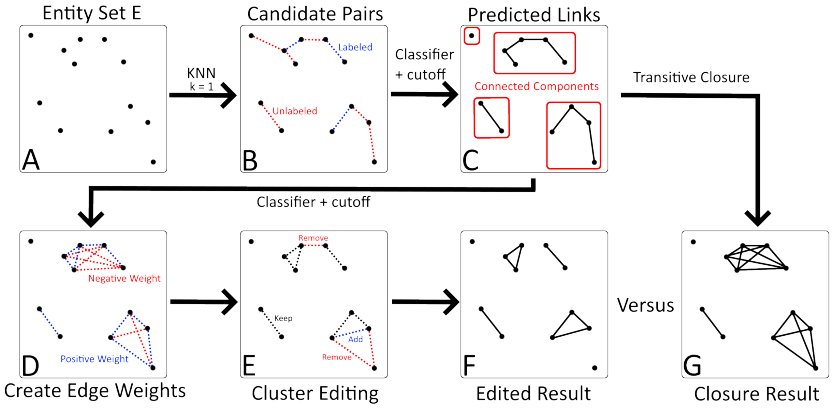


Fig. 1: An overview of the entity matching process.

Results: Table 1 shows the results of our experiments. We denote the application of transitive closure with the subscript TC and the application of cluster editing with the subscript CE. We have created three versions of a semi-synthetic dataset, denoted $\mathcal{D}_{10}$, $\mathcal{D}_{25}$ and $\mathcal{D}_{50}$. Each version has a different distribution of cluster sizes, as shown in Fig. 2. For every value of $\theta \in (0, 1)$ (in steps of 0.01) we generate an associated $F\frac{1}{2}$-score, as it is our experience that a low precision has a larger negative impact (than low recall) on the performance of downstream systems such as SPARQL engines. The $F\frac{1}{2}$-score weights precision twice as heavy as recall. We average the $F\frac{1}{2}$-score for all values of θ (100 values between 0 and 1) to denote the performance of a given combination of cluster distribution, classifier and features. We experimented with a range of classifiers, and show the best performing: logistic regression (LR) and support vector machine (SVM). All were trained using cosine similarity as the sole feature. Furthermore, we used just 100 pairs to train each classifier, limiting the burden on the domain expert as much as possible.

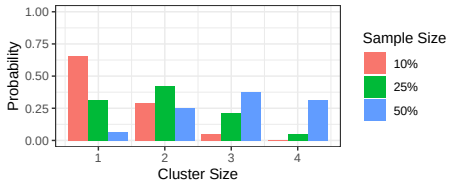
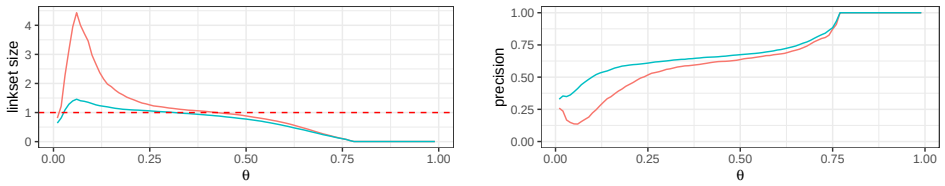


Fig. 2: Generated probability distributions of entity clusters of size 1 to 4 in the synthetic data. The values of a color sum to one.



(a) $\mathcal{D}_{50}$, LR - Cosine Similarity

(b) $\mathcal{D}_{50}$, LR - Cosine Similarity

Fig. 3: Left: relative number of pairs predicted vs actual number (red dotted line). Right: precision for transitive closure (red lines) and edited closure (blue lines).

In all cases we observe that the application of cluster editing improves the resulting set of duplicates over the application of transitive closure. Furthermore, the optimal value for θ is in most cases reduced when cluster editing is applied, suggesting that a more lenient cutoff can be used, while at the same time improving performance. Furthermore, Fig. 3a shows how closure of the entity pair set tend to overestimate the number of duplicate pairs and Fig. 3b shows that the cluster editing method consistently outperforms transitive closure in precision.

3 Conclusion and Future Work

In practice, entity matching methods are applied and the resulting entity pairs are used by, e.g., a reasoner in a SPARQL engine, which applies the transitive

classifier	dataset	θ_{TC}	θ_{CE}	Mean_{TC}	Mean_{CE}	Max_{TC}	Max_{CE}
LR	D_{10}	0.43	0.51	0.46	0.50	0.69	0.70
	D_{25}	0.40	0.35	0.37	0.41	0.62	0.64
	D_{50}	0.51	0.41	0.38	0.44	0.62	0.64
SVM	D_{10}	0.72	0.83	0.61	0.64	0.68	0.70
	D_{25}	0.70	0.66	0.43	0.51	0.62	0.64
	D_{50}	0.78	0.67	0.45	0.54	0.62	0.64

Table 1: A comparison of the mean and maximum $F\frac{1}{2}$-scores, and associated value for θ, per classifier and cluster distributions.

closure. This may introduce many spurious links, potentially creating large clusters of unrelated entities. We propose to apply cluster editing to create a set of links that is closed under transitivityand show that the application of cluster editing, compared to the transitive closure, always results in a set of duplicates that contains fewer distinct entity pairs (i.e. they have a higher precision) while retaining duplicate entity pairs (i.e. recall is not lowered). The NP-Hardness of cluster editing limits us to solving only relatively small connected components. There are, however, heuristic methods which enables larger components to be solved. These heuristic methods can effectively reduce the instance size of the problem and are fast in case a small number of edits is allowed [4].

References

1. Al Hasan, M., Zaki, M.J.: A survey of link prediction in social networks. In: Social network data analytics, pp. 243–275. Springer (2011)
2. Böcker, S., Baumbach, J.: Cluster editing. In: Conference on Computability in Europe. pp. 33–44. Springer (2013)
3. Chen, M., Tian, Y., Chang, K.W., Skiena, S., Zaniolo, C.: Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment. arXiv preprint arXiv:1806.06478 (2018)
4. Rahmann, S., Wittkop, T., Baumbach, J., Martin, M., Truss, A., Böcker, S.: Exact and heuristic algorithms for weighted cluster editing. In: Computational Systems Bioinformatics: (Volume 6), pp. 391–401. World Scientific (2007)
5. Saeedi, A., Nentwig, M., Peukert, E., Rahm, E.: Scalable matching and clustering of entities with famer. Complex Systems Informatics and Modeling Quarterly (16), 61–83 (2018)
6. Sun, Z., Hu, W., Li, C.: Cross-lingual entity alignment via joint attribute-preserving embedding. In: ISWC. pp. 628–644. Springer (2017)
7. Trisedya, B.D., Qi, J., Zhang, R.: Entity alignment between knowledge graphs using attribute embeddings. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 297–304 (2019)
8. Weiss, G.M.: Mining with rarity: a unifying framework. ACM Sigkdd Explorations Newsletter **6**(1), 7–19 (2004)
9. Zhu, H., Xie, R., Liu, Z., Sun, M.: Iterative entity alignment via joint knowledge embeddings. In: IJCAI. vol. 17, pp. 4258–4264 (2017)