6th International Conference on Computational Social Science IC²S²
July 17-20, 2020, 75 Amherst St. Cambridge, MA USA

Large-scale author name disambiguation using approximate network structures

Tong Zeng[1,2] and Daniel Acuna[2,*]

[1] School of Information Management, Nanjing University, Nanjing, China

[2] School of Information Studies, Syracuse University, Syracuse, USA

Keywords: Author name disambiguation, Entity resolution, Clustering, Approximate nearest neighbor, Minimum-spanning tree

Introduction. Properly identifying the author of a scientific article is an important task for giving credit, tracking progress, and identifying ideas' lineages. Usually, publications and citations do not provide unique identifiers to authors but only the raw string character representation of their name and affiliation. The fundamental problem is that an author might change the string representations due to changing in name spelling (e.g., removing accents), journal limitations (e.g., only allow first letter of first name), or simply two people having the same name. Several researchers have proposed methods to solve this problem[1,2,3], but most methods do not scale well and are not open to the community. In this work, we develop a scalable method that we make publicly available to disambiguate large-scale publications.

Methods. Given a set of publications $P = \{p_0, p_1, ..., p_n\}$, we extract all the authors in P to form a set consisting of unique individuals $A = \{a_0, a_1, ..., a_m\}$. Further, a unique authorship can be represented by a signature—a set of fields (e.g., the author name, affiliation, title, abstract) extracted from p. Let us denote all the signatures extracted from P by $S=\{s \mid s \in p\}$. The task of author name disambiguation can be defined as partitioning S into a number of clusters $C = \{c_0, c_1, ..., c_m\}$, so that $S = \cup c_i$, $c_i \cap c_j = \emptyset$ for all $i \neq j$ and associate each cluster c_i with an author so that all the signatures in the cluster belong to the same individual a_i. We proposed a method named ANNGC (Approximate Nearest Neighbors Graph Clustering), which contains three components: blocking, linkage, and clustering.

* Corresponding author.

E-mail address: deacuna@syr.edu (D.E. Acuna).

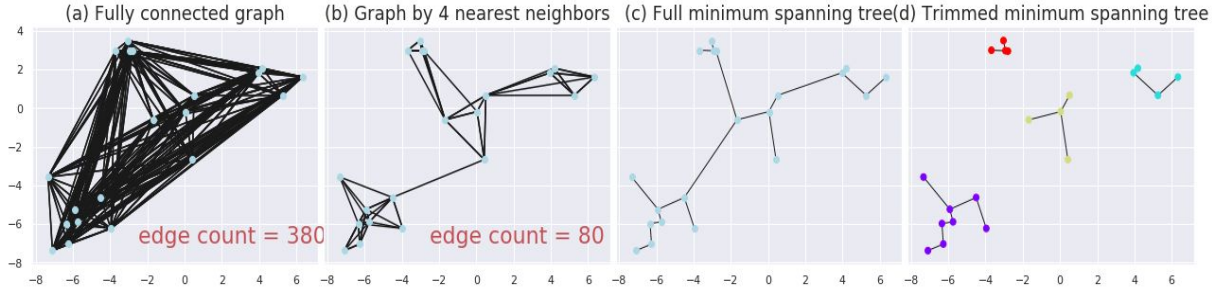
Blocking: The aim of blocking is to roughly partition S into a number of groups called blocks $B = \{b_1 b_2, ...b_k\}$ and then perform disambiguation only within a block. By blocking, the computational complexity of a typical clustering algorithm $O(|S|^2)$ is reduced to $O(\Sigma_{i=0}^{|B|}|b_i|^2)$. By choosing $|b_i| << |S|$, the difference in complexity is substantial. The most common blocking for name disambiguation is Surname and First name Initial (SFI). This strategy may fail for names with variations such as fluctuations (e.g., Acuña vs Acuna) or forms (Tchaikovsky vs Czajkowski). We solve these problems by doing name normalization and name phonetic representation. The name normalization converts Unicode characters into ASCII code which removes the fluctuations. The phonetic algorithm we use is Double Metaphone[3], which maps the names with the same pronunciation into the same form.

Linkage: Training a function predicts the probability of two signatures belonging to the same author. In order to convert the signature pairs into feature vectors, we convert co-author names and abstract into vectors using word-wise tfidf, for the first name, full name, affiliation, title, and journal we use character-wise tfidf. We then compute the cosine similarity between those tfidf vectors of the signature pairs. We also use absolute year differences as features. All the features are fed into a logistic regression classifier.

Clustering. Several previous studies have used Hierarchical Agglomerative Clustering (HAC) to partition signatures into groups. However, as the time complexity of HAC is $O(n^3)$, it is difficult to scale to large blocks. Also, each merging step in HAC requires global distance information of the whole block, thus difficult to distribute the model into multiple nodes. In order to overcome these problems, we propose a scalable and distributable algorithm whose steps are described now (see Figure 1). Step 1) Find top-k Approximate Nearest Neighbors (ANN) for each signature in the feature space. Step 2) connect each signature with its k neighbors, use the linkage function to predict the *dissimilarity* of the same author for each of the k signature pairs. Step 3) form a Minimum Spanning Tree (MST) from the partially connected graph, so that each signature is connected to one another signature while minimizing the total sum of distance between connected signatures. Step 4) Choose a cut-off threshold to disconnect the MST. The threshold can be chosen globally for all blocks (global

cut) or locally for each block (local cut). In the final Step 5, find all the connected components in the graph, where each connected component corresponds to an author.

Figure 1: Main steps of ANNGC algorithm



Results. *Data***:** 1. *ORCID dataset*. The orcid.org is a service that provides persistent identifiers to distinguish one author from every other. 2. PubMed Open Access subset (PMOA) is a collection of publications in biomedical science. We use ORCID —publication mapping as the group truth, and use PMOA to extract the signatures. *Evaluation***:** We adopt the 3-fold cross-validation, the performance is reported by average the metric scores on test set. We use B3 precision, recall and F1 as evaluation metrics, the higher the score, the better. *Baseline model***:** The HAC for clustering is used as a baseline. The blocking and linkage function are the same as ANNGC. As shown in Table 1, ANNGC local cut has the best precision, but the overall performance is slightly lower than the HAC. The local cut is better than global cut in both methods, as the thresholds are customized for each block.

Table 1: Performance on test set

Description	P	R	F1
Baseline-Global cut	0.978	0.9793	0.9788
Baseline-Local cut	0.997	**0.9946**	**0.9962**
ANNGC-Global cut	0.968	0.9766	0.9727
ANNGC-Local cut	**0.998**	0.9523	0.9747

Conclusion. While the performance of ANNGC is slightly lower than HAC, our approach has three advantages: 1) less time complexity. The complexity is $O(V \cdot h \cdot h\_root(V))$ for ANN plus $O(E \cdot logV)$ for MST, much lower than HAC ($O(V^3)$), given V the number of vertices, E the number of edges. 2) distributable, all the steps (ANN, MST, connected component) could be implemented without having the information of the whole blocks. 3) fewer hyper-parameters. There are only two parameters, the k for nearest neighbors and the cut-off threshold. These features are critical for large scale author name disambiguation which is not capable for HAC. In the future, we will experiment with network embeddings which could reduce our vector dimension from million to thousands. Also, we could use more information about the paper, such as the reference list. We could also apply more advanced models for the linkage function.

Acknowledgements

Tong Zeng was funded by the China Scholarship Council #201706190067. Daniel E. Acuna was funded by the National Science Foundation awards #1646763 and #1800956.

References

[1] Neil R Smalheiser,Vetle I Torvik. Author name disambiguation. Annual review of information science and technology, 43(1):1–43, 2009.

[2] Yutao Zhang, Fanjin Zhang, Peiran Yao, and Jie Tang. Name disambiguation in aminer: Clustering, maintenance, and human in the loop. In Proceedings of the 24th ACM SIGKDD

[3] Gilles Louppe, Hussein T Al-Natsheh, Mateusz Susik, and Eamonn James Maguire. Ethnicity sensitive author disambiguation using semi-supervised learning. In international conference on knowledge engineering and the semantic web, pages 272–287. Springer, 2016.