Strong Baselines for Author Name Disambiguation with and without Neural Networks

First Author[1,2], Second Author[1,2], and Third Author[1]

[1] First Institute
[2] Second Institute
{email}@email.email

Abstract. Author name disambiguation (AND) is one of the most vital problems in scientometrics, which has become a great challenge with the rapid growth of academic digital libraries. Existing approaches for this task substantially rely on complex clustering-like architectures, and they usually assume the number of cluster is known beforehand or predict the number by applying another model, which involve increasingly complex and time-consuming architectures. In this paper, we combine simple neural networks with two sets of heuristic rules to explore strong baselines for the author name disambiguation problem without any priori knowledge or estimation about cluster size, which frees the model from unnecessary complexity. On a popular benchmark dataset AMiner, our solution significantly outperforms several state-of-the-art methods both in performance and efficiency, and it still achieves comparable performance with many complex models when only a group of rules is utilized. Experimental results also indicate that gains from sophisticated deep learning techniques are quite modest in the name disambiguation problem.

Keywords: Author Name Disambiguation · Heuristic Rules · Clustering Problem · Baseline Methods.

1 Introduction

There has been significant historic and recent interest in the author name disambiguation (AND) problem, which can be defined as the problem of clustering unique authors using the metadata of publication records (title, venue, keyword, author name and affiliation, etc.) [10, 18, 22]. With the fast growth of scientific literature, the disambiguation problem has become an imminent issue since numerous downstream applications are affected by its preferences, such as information retrieval and bibliographic data analysis [5, 12]. But unfortunately, AND is not an elementary problem because distinct authors may share the same name, which is quite common for Asians, especially Chinese researchers [8], since different Chinese names will be the same when mapped to English (*e.g.*, 王伟 and 汪卫 share the same English name Wei Wang).

The problem of disambiguating *who is who* dates back at least few decades, and it is typically viewed as a clustering problem and solved by various clustering models, such models have to answer two questions inevitably, that is how to quantify the similarity and how to determine cluster size [7]. Many existing literatures mainly focus on answering the first question, such as feature-based methods [11, 12] and graph-based methods [3, 15, 19]. Actually, quite a few of them involve increasingly complex and time-consuming architectures that yield progressively smaller gains over the previous state-of-the-art. When it comes to the second question, most previous approaches assume the number of clusters is known beforehand or predict the number by applying another model [24]. However, there is no doubt that the former is unrealistic in real situations and the latter may lead to error propagation.

Lost in this push, we argue that author name disambiguation is not a typical clustering task. From the source of this problem, we should pay more attention to the precision, followed by recall, since that once two clusters are merged incorrectly, re-splitting them is an almost impossible process. Cast in this light, many existing clustering models are not very suitable for the author name disambiguation problem. Meanwhile, cost-effective blocking technique [1] and lightweight rule-based methods [2, 21] are worthy of research as they have been proven to achieve convincing precision in this problem.

In line with an existing research that aims to improve empirical rigor by focusing on insights and knowledge, as opposed to simply "winning" [16], we peel away unnecessary components until we arrive at the simplest model that works well without any priori knowledge about cluster size, which only consists of simple neural networks and some heuristic rules. Furthermore, the hierarchical agglomerative clustering (HAC) algorithm is adopted as the guiding ideology to cluster publications. On the benchmark dataset AMiner [24], we find that our proposed solution achieves significantly better performance than several state-of-the-art methods. Experiments on another public dataset show that such rules conform to the natural law and are applicable to the whole author name disambiguation task rather than just the AMiner dataset. Experimental results also suggest that while complex models do indeed contribute to meaningful advances towards this problem, some of them exhibit unnecessary complexity and rules play a role that cannot be ignored in this task.

2 Problem Definition

Given an author name α and a set of publication records $\mathcal{P} = \{p_1, p_2, ..., p_l\}$ with the name α, the problem of author name disambiguation is to partition the publication records $\mathcal{P}$ into different clusters $\{C_1, C_2, ..., C_K\}$ such that:

- All the records in C_k belong to the same author α_k.
- All the records in $\mathcal{P}$ by α_k are in C_k.

where $\{\alpha_1, ..., \alpha_K\}$ are K different people with the same name α.

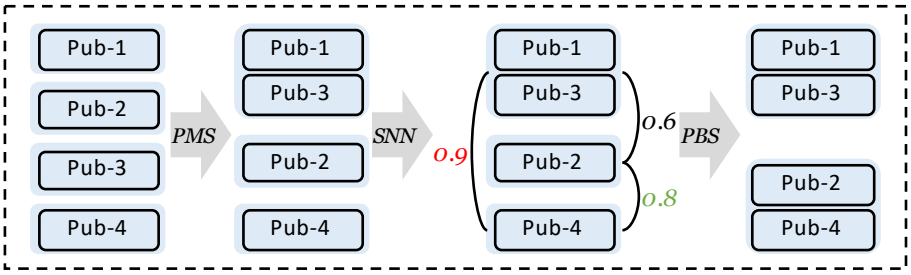


Fig. 1. A concrete process of our proposed approach.

3 Methodology

In this section, we discuss the design and implementation of our solution in detail, whose design philosophy is based on the observation that the interests of researchers usually do not change too frequently, and in particular, he/she would stay in the same institution for a relatively long time [3]. For this purpose, we can infer that researchers usually have relatively stable sets of coauthors, and topics of publications belong to a researcher should be close in the semantic space during a certain period. This is also in line with the law of human social activities in the real world, that is, friends and interests of a person are usually relatively fixed [**?**].

With this in mind, we first scatter the publication records $\mathcal{P} = \{p_1, p_2, ..., p_l\}$ into l sets, and there is only one unique publication p in each original set. Next, a pre-merging strategy (PMS) is proposed to make preliminary merge decisions according to coauthors. Furthermore, simple neural networks (SNN) are further employed to measure the semantic similarity between two clusters by publication titles, since titles naturally convey the main point of publications. Finally, we introduce a post-blocking strategy (PBS) to determine the final clusters elegantly. Figure 1 shows a concrete process of our proposed approach.

3.1 Pre-Merging Strategy

This step aims to merge the initial publication sets preliminarily using the point-to-point and cluster-to-cluster rules. For convenience, we set an identity constraint $\mathcal{M}(i, j) \in \{1, 0\}$ to indicate that i and j will (not) be merged into a cluster, where i and j refer to two publications or clusters.

- **Point-to-Point**: Given two publications p_i and p_j, if $|S_n(p_i) \cap S_n(p_j)| > \lambda_1$, or $A_\alpha(p_i) = A_\alpha(p_j)$ & $|S_n(p_i) \cap S_n(p_j)| > 1$, then $\mathcal{M}(p_i, p_j) = 1$. For a publication p_i, $S_n(p_i)$ and $A_\alpha(p_i)$ denote the set of author names and the affiliations of current author name α, respectively.
- **Cluster-to-Cluster:** Given two clusters C_i and C_j, if $\mathcal{O}_n(C_i, C_j) > \lambda_2$, or $\mathcal{O}_a(C_i, C_j) > \lambda_2$, then $\mathcal{M}(C_i, C_j) = 1$, where $\mathcal{O}_x(C_i, C_j)$ denotes the overlap

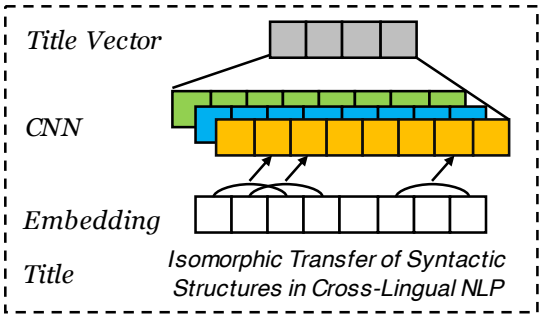


Fig. 2. An illustration of our simple neural networks.

ratio of two clusters in the aspect of x, and $x \in \{n, a\}$ denotes the name or affiliation of authors. We define the overlap ratio $\mathcal{O}_x(C_i, C_j)$ as:

$$\mathcal{O}_x(C_i, C_j) = \frac{\sum_{\bar{x} \in (S_x(C_i) \cap S_x(C_j))} (F_{\bar{x}}(C_i) + F_{\bar{x}}(C_j))}{\min(\sum_{\bar{x} \in S_x(C_i)} F_{\bar{x}}(C_i), \sum_{\bar{x} \in S_x(C_j)} F_{\bar{x}}(C_j))} \tag{1}$$

where $F_{\bar{x}}(C_i)$ is the occurrence number of $\bar{x}$ in the cluster C_i.

Intuitively, the point-to-point stage can be understood as that we regard two publications p_i, p_j belong to the same cluster when the number of coauthors of p_i and p_j exceeds a threshold λ_1, and if the affiliations of current author name α are identical in p_i and p_j, the threshold is relaxed to 1, which means that only one coauthor except α is needed to satisfy the merge condition. For ease of exposition, we take author name as an example here (*i.e.*, $x = n$) to describe the process of cluster-to-cluster stage. To calculate the numerator of overlap ratio $O_n(C_i, C_j)$, we consider names that appear in the intersection of two name sets $S_n(C_i), S_n(C_j)$, and calculate the total occurrence number of such names in these two clusters. In addition, the minimum of total occurrence number of all author names in $S_n(C_i)$ and $S_n(C_j)$ is defined as the denominator. Considering that the total number of author names in two clusters may vary greatly, such a minimum value selection strategy can effectively avoid the problem that a small cluster cannot be merged with a large cluster.

3.2 Simple Neural Networks

As mentioned above, it is a natural idea to determine whether two publications belong to the same author by their topic similarity, since topic reflects the interest and direction of a researcher. In order to quantify the similarity effectively, we design a simple model based on convolutional neural networks (CNN) to project publications into a low-dimensional latent common space.

We believe that *title* contains enough information to express the topic of a publication, thus we first transform each $p_i \in \mathcal{P}$ into a sequence of vectors

$[\mathbf{w}_1, ..., \mathbf{w}_n]$, where $\mathbf{w}_j$ is the embedding of j-th word in title. Note that we employ CBOW [13] to pre-train initial word embeddings, which will be fine-tuned afterwards. Then a standard CNN is utilized to produce title vector for each publication, in which convolutional operations are performed with m different filters, and the final representation $\mathbf{p}_i \in \mathbb{R}^m$ is computed by a max-pooling layer. Next, we follow the basic idea of [24] to train this representation model. Let (p_i, p_{i_+}, p_{i_-}) be a triplet where p_{i_+} and p_i are publications authored by the same person, while p_{i_-} is a randomly selected negative example belonging to another person. Hence, our training data $\mathcal{T}$ consists of a set of triplets and we optimize a margin-based loss function as follows:

$$\mathcal{L} = \frac{1}{|\mathcal{T}|} \sum_{i=1}^{|\mathcal{T}|} \max(0, \delta - \cos(\mathcal{R}(p_i), \mathcal{R}(p_{i_+})) + \cos(\mathcal{R}(p_i), \mathcal{R}(p_{i_-}))) \quad (2)$$

where $\mathcal{R}(\cdot)$ denotes the representation model and δ is the margin. The intuition behind Equation 2 is that we want the positive pair (p_i, p_{i_+}) to be more similar to each other than their negative example p_{i_-}, by a margin of at least δ.

For a given cluster C_i containing $|C_i|$ publications, the cluster embedding is defined as $\mathbf{c}_i = \frac{1}{|C_i|} \sum_{j=1}^{|C_i|} \mathcal{R}(p_j)$. We choose the cluster with the highest similarity with C_i as its target merging cluster, denoted as C_j, the similarity between these two clusters is measured by the cosine similarity between $\mathbf{c}_i$ and $\mathbf{c}_j$. Finally, C_i and C_j will be merged with some post-blocking strategies, we will discuss it in the following paragraph.

3.3 Post-Blocking Strategy

Based on the learned model $\mathcal{R}(\cdot)$ and cluster embeddings, this step is proposed to determine the final partition. To avoid undesirable mergence caused by only measuring intra-cluster semantic similarity, we introduce Post-Blocking Strategy to take the statistical characteristics of two clusters into consideration. In our design, publication with the largest number of coauthors in C_i is selected as the anchor publication p_i^*, and p_j^* for C_j can be selected similarly. Then the anchor-to-anchor rule is deployed as follows:

- **Anchor-to-Anchor:** if $S_n(p_i^*) \cap S_n(p_j^*) = \{\alpha\}$ and $S_{a\setminus\alpha}(p_i^*) \cap S_{a\setminus\alpha}(p_j^*) = \emptyset$, then $\mathcal{M}(C_i, C_j) = 0$. For an anchor publication p_i^*, $S_{a\setminus\alpha}(p_i^*)$ denote the set of affiliations except the current author α.

The anchor-to-anchor rule can be interpreted as that, if there is no intersection between the name sets or the affiliation sets of p_i^* and p_j^* except the current author name α and its affiliation, we do not think C_i and C_j belong to the same author. To illustrate this process intuitively, we describe an example in Figure 1 (the third step). Although the similarity between {Pub-1, Pub-3} and {Pub-4} is the highest, the merge operation is still blocked as the anchor-to-anchor rule is violated.

4 Experiments

4.1 Dataset

We conduct our experiments on a recently widely used public benchmark dataset AMiner introduced in [24][3], which is sampled from a well-labeled academic database. The labeling process of the dataset is based on the publication lists on authors' homepages and the affiliations, e-mails in web databases (*e.g.* Scopus, ACM Digital Library). The training set contains publications of 500 author names, and the test set has 100 author names. For each publication, there are five fields as follows: *title*, *keywords*, *venue*, *author name and corresponding affiliation*. In this paper, we only use title, author name and affiliation to develop our solution. Compared with existing benchmarks for name disambiguation, AMiner is significantly larger (in terms of the number of documents) and more challenging (since each candidate set contains much more clusters) [24].

4.2 Experiment Settings

Following popular choices, we tune our model using five-fold cross validation. For the pre-merging strategy (PMS), we set λ_1 to 2 and λ_2 to 0.5 experimentally. Beyond that, CBOW model [13] with $k = 100$ is employed to learn initial word representations on the training set of AMiner. SNN model is trained using Stochastic Gradient Descent (SGD) algorithm with the initial learning rate of 0.1 and the weight decay of 0.9, the batch size is 50 and the margin is 0.3. At convolutional layer, the number of filter maps is 100 and the window size is 3. Dropout with $p = 0.3$ is used after the input layer.

4.3 Comparison Methods

Following Zhang et al. [24], we compare our model against 5 different methods:

- **Basic Rules** [24]**:** It constructs linkage graphs by connecting two publications when their co-authors, affiliations or venues are strictly equal. Results are obtained by simply partitioning the graph into connected components.
- **Fan et al.** [3]**:** For each name, it constructs a graph by collapsing all the co-authors with identical names to one node. The final results are generated by affinity propagation algorithm and the distance between two nodes is measured based on the number of valid paths.
- **Louppe et al.** [12]**:** It trains a pairwise distance function based on carefully designed similarity features, and uses semi-supervised Hierarchical Agglomerative Clustering (HAC) algorithm to determine clusters.
- **Zhang and Al Hasan** [23]**:** It constructs graphs for each author name based on co-author and document similarity. Embeddings are learned for each name and the final results are also obtained by HAC.

[3] https://static.aminer.cn/misc/na-data-kdd18.zip

Table 1. Results of author name disambiguation on the AMiner benchmark dataset. † marks results reported in [24].

Model	Precision	Recall	F1 Score
Basic Rules [24]†	44.94	89.30	53.42
Fan et al. [3]†	81.62	40.43	50.23
Louppe et al. [12]†	57.09	**77.22**	63.10
Zhang and Al Hasan [23]†	70.63	59.53	62.81
Zhang et al. [24]†	77.96	63.03	67.79
PMS	**81.86**	55.61	66.23
PMS+SNN	73.90	61.97	67.41
PMS+SNN+PBS (PNP)	76.92	64.54	**70.19**

– **Zhang et al.** [24]**:** It introduces a representation learning framework by leveraging both global supervision and local contexts, and also uses HAC as clustering method, which is the latest approach on the dataset[4]. Besides, it deploys the recurrent neural networks to estimate the number of cluster.

Our method is indicated by **PNP**. In order to analyze the contribution of each component, we present results at each of the three stages as described in Section 3.

4.4 Results

Table 1 shows the performances of different methods on the AMiner dataset. Following previous settings [24], we utilize pairwise Precision, Recall, and F1-score to evaluate all methods. Meanwhile, a macro averaged score of each metric is calculated according to all test names.

Louppe et al. [12] use some manual features to learn a pairwise similarity function and achieve competitive performance. Similarly, we hold the view that in the era of deep learning, the feature engineering methods may not be certainly worse than some exquisite neural networks, especially in this task. Moreover, some features can be further abstracted into rules and be used universally in the whole author name disambiguation problem, rather than limited to this dataset, we will discuss this phenomenon in Section 5.2. However, The results of Basic Rules are disappointing and contrary to ours. Since there is no implementation details, we speculate that it might be because the merging rules are too loose. By incorporating both rules and neural networks to model co-authorships, affiliations and titles explicitly, our PNP model outperforms all baselines in terms of F1-score (+3.54% over Zhang et al. [24], +11.75% over Zhang and Al Hasan [23], +11.23% over Louppe et al. [12] and +39.74% over Fan et al. [3] relatively).

In the bottom half part of Table 1, some incremental results of our method are presented. Specifically, PMS outperforms most baselines, which indicates the

[4] https://github.com/neozhangthe1/disambiguation/

Table 2. Runtime and trainable parameter number of different models.

Model	Runtime		Trainable
	Training	Testing	Parameters Number
Zhang el al. [24]	>24h	∼573s	3,024,193
PMS	-	∼31s	0
SNN	∼2h	-	30,100
PBS	-	∼88s	0
PMS+SNN+PBS (PNP)	∼2h	∼119s	30,100

effectiveness of heuristic rules. PMS+SNN with optimal reject threshold (0.8) yields better performance than PMS (+1.78% in terms of F1-score), which suggests the advantage of SNN. PNP outperforms PMS+SNN by +4.12% in terms of F1-score and +4.09% in terms of precision which verifies the incorporation of PBS can greatly enhance the performance. Overall, we attribute these success to the comprehensive consideration of rules and semantics based on the inherent characteristics of the author name disambiguation problem.

5 Analyses

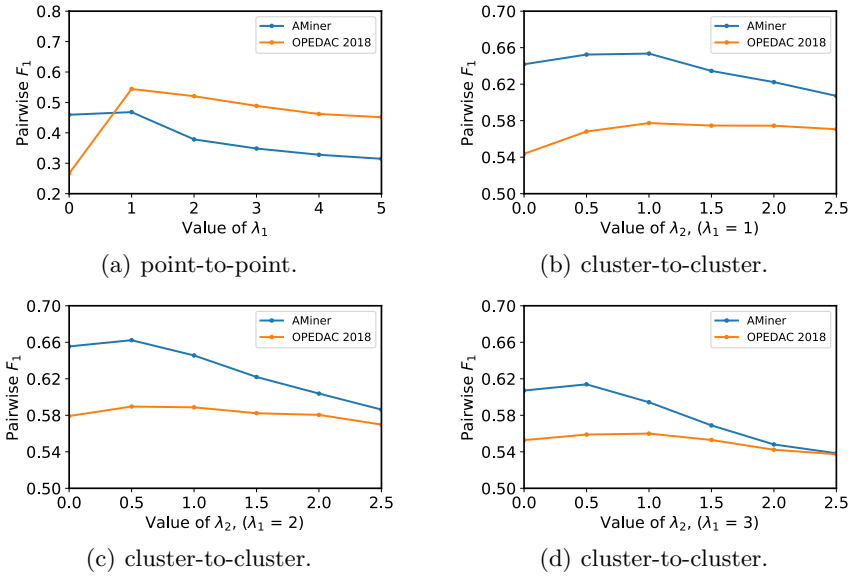
5.1 Efficiency Analysis

We study the runtime and model size (except word embeddings) of our method as well as the state-of-the-art model [24] using official implementation. For the sake of fairness, we run them on the same GPU server.

From Table 2, we find that Zhang et al. [24] is indeed computationally expensive, which is caused by the complex operations in modeling the local linkage with graph auto-encoder and estimating the number of clusters. Instead, our PNP model is quite simpler and faster because it mainly relies on the heuristic rules to model co-authors rather than embed the local co-authorship into representations. Beyond that, our proposed model removes the need to know or estimate cluster size beforehand, which is unrealistic or time-consuming. Generally, our approach is almost 5 times faster than the state-of-the-art model in test time and has a significant advantage in the model size, which means that training our model requires much fewer computation resources and less time.

5.2 Rule Sensitivity Analysis

As aforementioned, we hold that the optimal parameters (λ_1 and λ_2) can be applicable to the general author name disambiguation task, rather than limited to the AMiner dataset. To validate our opinion, we employ the Open Academic Data Challenge 2018 (OPEDAC 2018) dataset to analyze the parameter sensitivity of rules. OPEDAC 2018 consists of 80,050 publications of 50 authors.



(a) point-to-point. (b) cluster-to-cluster.

(c) cluster-to-cluster. (d) cluster-to-cluster.

Fig. 3. The effects of λ_1 and λ_2 to the performance when using different datasets. The subfigure (a) depicts results when only the point-to-point rules are used, while remaining subgraphs show the results of the entire pre-merging strategy.

Different from AMiner, many publications in OPEDAC 2018 contain hundreds of authors, which means that it is more difficult and closer to reality.

By varying the value of λ_1 across $\{0, 1, 2, 3, 4, 5\}$, we repeat the experiments and report results in Figure 3(a). As observed, when λ_1 increases, the F1-score first increases and then decreases, the best performances of both datasets are achieved when $\lambda_1 = 1$. It's intuitive, because a person usually has a fixed partner, such as a mentor or leader. Furthermore, as shown in Figure 3(b), 3(c) and 3(d), when fixing the value of λ_1 and varying λ_2, the two datasets have the similar trends and achieve the peak at almost the same value of λ_2, which is a strong evidence for our claim that such rules conform to the natural law and the hyper-parameters of rules are relatively insensitive to datasets. We hypothesize that this phenomenon is due to the particularity of problem, which is that friends and affiliations of a person are usually relatively fixed.

We also reproduce the state-of-the-art model [24] on the OPEDAC 2018 dataset and achieved the F1-score of 50.4%, and our PNP model outperforms it by a substantial margin (+15.4%), which suggests the generalizability of our model[5]. It is worthy to mention that the results of OPEDAC 2018 in Figure 3 should not be compared with other competitors in the leaderboard. The reason is that OPEDAC 2018 suffers from the problem of noise in the author list. Actually,

[5] Note that we do not report the comparison results on this dataset in Section 5.2 because we cannot reproduce other baseline methods without official implementation.

when combined with other denoising strategy, our PNP method finally ranks top 3% without any ensemble tricks in the competition.

5.3 Error Analysis

We analyze some of the errors made by our model on the AMiner dataset, and find that the most common error is the incorrect mergence when publications have the same short and incomplete affiliations (*e.g.* Department of Computer Science). In other words, there might be two different people with the same name who happen to work in the department of computer science, but if they do not belong to the same school, things will become more tricky.

For this purpose, we perform a supplemental experiment to explore the upper bound of precision, we merge two publications if and only if $S_n(p_i)$=$S_n(p_j)$ & $S_a(p_i)$=$S_a(p_j)$, which means that the name set and the affiliation set of two publications are exactly the same. Experimental results show that the precision is just about 95%. When facing the remaining 5%, even humans have no certain confidence to deal with them correctly. In this case, after removing these indistinguishable samples, our pre-merging strategy attains 86% precision, which is quite acceptable for the unsupervised heuristic rules.

6 Related Work

In many applications, author name disambiguation (AND) has been regarded as a challenging problem, which can date back at least few decades. With the growth of scientific literature, it becomes more and more difficult and urgent to solve this problem [4, 17, 20]. Based on the different scenarios, the author name disambiguation problem can be divided into two subtasks: author name disambiguation from scratch (ANDS) [23, 24] and incremental author name disambiguation (IAND) [8, 9], the former is generally a clustering problem, while the latter is a classification problem.

In this paper, we focus on the ANDS scenario, which is more challenging and practical than IAND. On the whole, state-of-the-art solutions for the task can be divided into two categories: feature-based and graph-based. Feature-based methods leverage pairwise distance function to measure documents. Huang et al. [6] first uses blocking technique to group candidate documents with similar names together and employs DBSCAN to cluster documents. Louppe el al. [12] uses a classifier to learn pairwise similarity and performs semi-supervised hierarchical clustering to generate results. Graph-based methods utilize graph topology and aggregate information from neighbors. Fan et al. [3] builds document graph for each name by co-authorship, and uses carefully-designed similarity function and affinity propagation algorithm to generate clustering results. Tang el al. [19] employs Hidden Markov Random Fields to model node and edge features in a unified probabilistic framework. Zhang and Al Hasan [23] learns graph embedding from three constructed graphs based on document similarity and co-authorship. Moreover, Zhang et al. [24] combines the advantages of above two methods by

learning a global embedding using supervised metric learning and refining the embedding using local linkage structures. In this push towards complexity, we do not believe that all researchers have adequately explored baseline methods, and thus it is unclear how much various fussy techniques actually help.

7 Conclusion

In this paper, we take heuristic rules that come from real-world observations into consideration and propose a strong baseline for the author name disambiguation problem. The proposed model contains a pre-merging strategy, simple neural networks and a post-blocking strategy, which do not need any extra knowledge about cluster size. Experimental results verify the advantage of our method over state-of-the-art methods, and demonstrate the proposed model is highly efficient and rules can be extended to other datasets, in which many conclusions are consistent with some sociological phenomena. Beyond that, we further explore the upper bound of disambiguation precision and analyze the possible reasons, which will be leaved as our future work. To conclude, we offer all data mining researchers a point of reflection like some previous work [14]: The most important thing is to consider baselines that do not involve complex architectures, simple methods might lead to unexpected performances. The source code of this paper can be obtained from https://github.com/xxx.

References

1. Backes, T.: The impact of name-matching and blocking on author disambiguation. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM). pp. 803–812 (2018)
2. Caron, E., van Eck, N.J.: Large scale author name disambiguation using rule-based scoring and clustering. In: Proceedings of the International Conference on Science and Technology Indicators (STI). pp. 79–86 (2014)
3. Fan, X., Wang, J., Pu, X., Zhou, L., Lv, B.: On graph-based name disambiguation. Journal of Data and Information Quality **2**(2) (2011)
4. Ferreira, A.A., Gonçalves, M.A., Laender, A.H.: A brief survey of automatic methods for author name disambiguation. Acm Sigmod Record **41**(2), 15–26 (2012)
5. Han, D., Liu, S., Hu, Y., Wang, B., Sun, Y.: Elm-based name disambiguation in bibliography. World Wide Web **18**(2), 253–263 (2015)
6. Huang, J., Ertekin, S., Giles, C.L.: Efficient name disambiguation for large-scale databases. In: Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD). pp. 536–544 (2006)
7. Hussain, I., Asghar, S.: A survey of author name disambiguation techniques: 2010–2016. The Knowledge Engineering Review **32** (2017)
8. Hussain, I., Asghar, S.: Incremental author name disambiguation using author profile models and self-citations. Turkish Journal of Electrical Engineering & Computer Sciences **27**(5), 3665–3681 (2019)

9. Kim, K., Rohatgi, S., Giles, C.L.: Hybrid deep pairwise classification for author name disambiguation. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM). pp. 2369–2372 (2019)
10. Levin, M., Krawczyk, S., Bethard, S., Jurafsky, D.: Citation-based bootstrapping for large-scale author disambiguation. Journal of the American Society for Information Science and Technology **63**(5), 1030–1047 (2012)
11. Liu, J., Lei, K.H., Liu, J.Y., Wang, C., Han, J.: Ranking-based name matching for author disambiguation in bibliographic data. In: Proceedings of the 2013 KDD Cup 2013 Workshop
12. Louppe, G., Al-Natsheh, H.T., Susik, M., Maguire, E.J.: Ethnicity sensitive author disambiguation using semi-supervised learning. In: Proceedings of the 7th International Conference on Knowledge Engineering and Semantic Web (KESW). pp. 272–287 (2016)
13. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proceedings of the 27th Annual Conference on Neural Information Processing Systems (NeruIPS). pp. 3111–3119 (2013)
14. Mohammed, S., Shi, P., Lin, J.: Strong baselines for simple question answering over knowledge graphs with and without neural networks. In: Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL). pp. 291–296 (2018)
15. Niu, F., Ré, C., Doan, A., Shavlik, J.: Tuffy: Scaling up statistical inference in markov logic networks using an rdbms. Proceedings of the Very Large Data Bases Endowment (VLDB) pp. 373–384 (2011)
16. Sculley, D., Snoek, J., Wiltschko, A., Rahimi, A.: Winner's curse? on pace, progress, and empirical rigor. In: Workshop on 6th The International Conference on Learning Representations (ICLR) (2018)
17. Shen, Q., Wu, T., Yang, H., Wu, Y., Qu, H., Cui, W.: Nameclarifier: A visual analytics system for author name disambiguation. IEEE transactions on visualization and computer graphics **23**(1), 141–150 (2016)
18. Smalheiser, N.R., Torvik, V.I.: Author name disambiguation. Annual review of information science and technology **43**(1), 1–43 (2009)
19. Tang, J., Fong, A.C., Wang, B., Zhang, J.: A unified probabilistic framework for name disambiguation in digital library. IEEE Transactions on Knowledge and Data Engineering **24**(6), 975–987 (2012)
20. Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z.: Arnetminer: extraction and mining of academic social networks. In: Proceedings of the 14th ACM International Conference on Knowledge Discovery & Data Mining (KDD). pp. 990–998 (2008)
21. Veloso, A., Ferreira, A.A., Gonçalves, M.A., Laender, A.H., Meira Jr, W.: Cost-effective on-demand associative author name disambiguation. Information Processing and Management: an International Journal **48**(4), 680–697 (2012)
22. Yoshida, M., Ikeda, M., Ono, S., Sato, I., Nakagawa, H.: Person name disambiguation by bootstrapping. In: Proceedings of the 23rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR). pp. 10–17 (2010)
23. Zhang, B., Al Hasan, M.: Name disambiguation in anonymized graphs using network embedding. In: Proceedings of the 26th ACM International Conference on Information and Knowledge Management (CIKM). pp. 1239–1248 (2017)
24. Zhang, Y., Zhang, F., Yao, P., Tang, J.: Name disambiguation in aminer: Clustering, maintenance, and human in the loop. In: Proceedings of the 24th ACM

International Conference on Knowledge Discovery & Data Mining (KDD). pp. 1002–1011 (2018)