SCIENCE CHINA
Information Sciences

• MOOP •

April 2021, Vol. 64 144101:1–144101:3
https://doi.org/10.1007/s11432-019-9884-y

Name disambiguation in AMiner

Jing ZHANG[1*] & Jie TANG[2]

[1]*Information School, Renmin University of China, Beijing* 100087, *China;*
[2]*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084, *China*

Received 16 March 2019/Accepted 23 April 2019/Published online 18 June 2020

Citation Zhang J, Tang J. Name disambiguation in AMiner. Sci China Inf Sci, 2021, 64(4): 144101, https://doi.org/10.1007/s11432-019-9884-y

Name disambiguation, aiming at disambiguating who is who, is one of the fundamental problems of the online academic network platforms such as Google scholar, microsoft academic and AMiner. This study takes AMiner[1)], a free online academic search and mining system [1], as the example to explain how we deal with the name ambiguity problem under three different scenarios. AMiner has already extracted 1.3×10^8 researchers' profiles from the Web and integrated with 2×10^8 papers from heterogeneous publication databases, with a growth rate of over 500000 per month. From the beginning when the system is built to the running and updating phases, we need to pay continuous attention on the problem of name disambiguation. In the following parts, we discuss the problem on three scenarios during the whole life cycle of AMiner, i.e., name disambiguation when the system is built from scratch (full ND), name disambiguation when persons' profiles are continuously updated (continuous ND) and error detection upon existing persons' profiles (error detection). Figure 1(a) illustrates an example of the disambiguating results for the researchers named "Jing Zhang" in AMiner and Figure 1(b)–(d) explains the problem of name disambiguation under three scenarios.

Full ND. At the beginning when we build AMiner, we collect a large number of published papers which need to be partitioned into groups, where each group represents the papers that are published by a same person. To achieve the goal, we formalize the problem as a clustering problem, and propose variant methodologies to solve it. Specifically, the problem is defined as follows.

Problem 1 (Full ND)**.** Given a be a name reference, and $\mathcal{D}^a = \{p_1^a, p_2^a, \ldots, p_N^a\}$ be a set of N papers associated with the author name a, we target at finding a function f to partition $\mathcal{D}^a$ into a set of disjoint clusters, i.e.,

$$f : \mathcal{D}^a \to \mathcal{C}^a, \text{ where } \mathcal{C}^a = \{c_1^a, c_2^a, \ldots, c_K^a\}, \qquad (1)$$

such that each cluster only contains papers belonging to the same person, i.e., $\mathbb{I}(p_i^a) = \mathbb{I}(p_j^a), \forall (p_i^a, p_j^a) \in c_k^a \times c_k^a$, and different clusters contain papers belonging to different persons, i.e., $\mathbb{I}(p_i^a) \neq \mathbb{I}(p_j^a), \forall (p_i^a, p_j^a) \in c_k^a \times c_{k'}^a, k \neq k'$, where $\mathbb{I}(p_i^a)$ denotes the person identity (corresponding real-world person) of paper p_i^a.

To solve the above defined clustering problem, we have tried the traditional feature-based Markov random field model [2] and also the current embedding-based hierarchical clustering algorithm [3]. Specifically, for the feature-based method, we define both the local features for each paper $p_i^a \in \mathcal{D}^a$ and the correlation features between two papers (p_i^a, p_j^a). The local features are defined as the similarities between a paper and its assigned cluster centroid according to papers' attributes such as title, venue, year, abstract, authors, and references. The correlation features are defined as the similarities between two papers according to their relationships such as the co-author, co-venue and citation relationships. Then we build an unsupervised Markov random field model to incorporate the local features and correlation features to iteratively assign papers to the closest new centroid and update the centroid and feature weights.

However, the above feature-based method is not easy to capture the semantic similarities of titles and abstracts. Besides, human-defined features may not be comprehensive to capture the similarities of papers. To avoid these limitations, we revisit the problem by proposing an embedding-based model plus a hierarchical clustering algorithm [3]. We train a global embedding and a local embedding for each paper. Specifically, based on several labeled data indicating whether two papers belong to a same person or not, we train a global triplet loss-based model to generate a global embedding for each paper. Based on the totally unlabeled papers $\mathcal{D}^a$ associated with each author name a, we build a graph on $\mathcal{D}^a$ by adding an edge between two papers in $\mathcal{D}^a$ if their similarity is above a threshold, and then train a graph auto-encoder to generate a local embedding for each paper in $\mathcal{D}^a$. Finally, the global and local embeddings are concatenated as the input of the hierarchical clustering algorithm to partition the papers in $\mathcal{D}^a$ for each name reference a.

To solve the problem of full ND, how to determine the number of the clusters K is a critical challenge. In [2], we use a pre-defined measurement such as Bayesian information criterion to determine whether a paper cluster should be split into two clusters or not. However, this method needs

[*] Corresponding author (email: zhang-jing@ruc.edu.cn)
[1)] http://aminer.org.

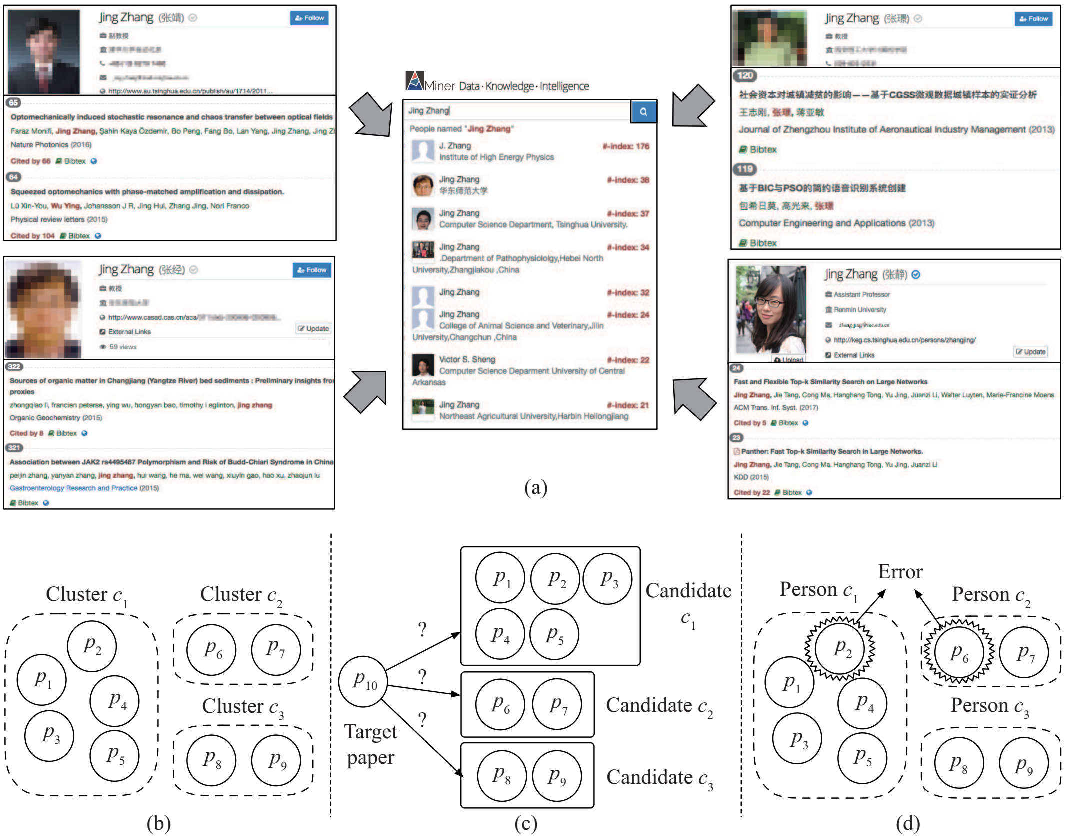

Figure 1 (Color online) (a) An example of name disambiguating results for the researchers named "Jing Zhang" in AMiner. Name disambiguation under three scenarios: (b) full ND; (c) continuous ND; (d) error detection.

iterative trial of possible splits, which is inefficient on large datasets. Thus, in [3], we treat a set of papers as input, and train a RNN model to directly predict the number of clusters K.

Continuous ND. Despite considerable research conducted in the problem of name disambiguation, a critical issue that has been largely ignored is how to perform the disambiguation in real-time. For example, there are about 500000 new arriving papers per month in AMiner, thus an important task is to assign the new papers to different researchers in the system. To achieve the above goal, we formalize the continuous name disambiguation problem as a ranking problem.

Problem 2 (Continuous ND)**.** Given a target paper p and the corresponding candidate persons $\mathcal{C}$, for each candidate person $c \in \mathcal{C}$, c is denoted as c^+ if c is the right author of p and is denoted as c^- if c is not the author of p. The target is to learn a scoring function f to calculate the similarity between p and c^+ or c^-:

$$\begin{aligned} &f : (p, c) \to \mathbb{R}, \\ &\text{s.t.} \quad f(p, c^+) > f(p, c^-), \ \forall c^- \in \mathcal{C}, \end{aligned} \tag{2}$$

where the target paper p is always associated with a name reference a. The candidate persons $\mathcal{C}$ to p can be generated by many ways, where one simplest way is to extract the candidate persons with the same name reference a.

To measure the similarity between a target paper and a candidate person, traditional unsupervised methods such as Jaccards Coefficient and cosine similarity can easily capture the exact matches between tokens. However, they suffer from the sparsity of the token-based representations. For example, the similarity is zero if two representations do not contain any same tokens, even if they are semantically similar. On the other hand, recently, some representation-based deep learning models can successfully capture the soft/semantic similarities, as they embed the high-dimensional sparse representations into low-dimensional dense representations. However, this model may suffer from the problem of semantic drift, as global representing of a paper or a person may dilute the effect of the exact same tokens in them by other different tokens. Thus, to capture both the exact and the soft matches, we adopt the interaction-based models [4], which are widely used in information retrieval. Through calculating the similarities between the embeddings of each pairs of tokens in the target paper and the candidate person, both the exact same relationships and the semantic same relationships between tokens can be modeled.

Error detection. No matter how accurate the disambiguation methods are, the errors of full NA and continuous NA cannot be avoided. Thus, we perform an error detection function to detect the wrongly assigned papers to persons.

Problem 3 (Error detection)**.** Given a person c and the set of N_c papers $\mathcal{D}^c = \{p_1, p_2, \ldots, p_{N_c}\}$ assigned to c, we target at learning a function:

$$f : p_k \to y_k, \ \text{where } y_k \in \{0, 1\} \tag{3}$$

to detect whether each $p_k \in \mathcal{D}^c$ is wrongly assigned to c $(y_k = 0)$ or not $(y_k = 1)$.

To address the problem, we construct a multi-relation graph $G^c = (V^c, E^c)$ upon the existing assigned papers $\mathcal{D}^c$

to each person c, where each node $v \in V^c$ is a paper and each edge $e \in E^c$ represents a relationship between two papers. We define 3 kinds of relationships including the co-author, co-venue and citation relationships, where the co-author relationships only consider the same author names except the name of the person c to be disambiguated. Then for each node in the graph, we extract its ego network composed by the node itself, all its neighbors and all the relationships between the nodes. We define 4 kinds of patterns for each ego network, including the number of neighbors, the number of relationships, total weight of all the relationships and the principal eigenvalue of the weighted adjacency matrix. Then we extract features following these four patterns for each relationship type and obtain $3 \times 4 = 12$ features for each node. The traditional outlier detection methods using all these features can be used to detect the outlier papers [5].

Summary. The study introduced the problem of name disambiguation under three different scenarios. Although the developed technologies to solve the problems are ready for online use, there is still room for improvement. For instance, the three functions are actually not dependent. It is possible to design a strategy to boost each function by the other functions. In addition to the three algorithms, we also developed "merge", "add" and"remove" functions to allow the users to manually merge person profiles, add news papers to persons or remove the wrongly assigned papers respectively. Furthermore, the collected user feedbacks can be leveraged to boost the performance of the name disambiguation algorithms.

Supporting information Videos and other supplemental documents. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Tang J, Zhang J, Yao L M, et al. ArnetMiner: extraction and mining of academic social networks. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, 2008. 990–998

2 Tang J, Fong A C M, Wang B, et al. A unified probabilistic framework for name disambiguation in digital library. IEEE Trans Knowl Data Eng, 2012, 24: 975–987

3 Zhang Y, Zhang F, Yao P, et al. Name disambiguation in AMiner: clustering, maintenance, and human in the loop. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018. 1002–1011

4 Hu B, Lu Z, Li H, et al. Convolutional neural network architectures for matching natural language sentences. In: Proceedings of the 27th Advances in Neural Information Processing Systems, 2014. 2042–2050

5 Leman A, Mary M, Christos F, et al. OddBall: spotting anomalies in weighted graphs. In: Proceedings of the 2010 Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2010. 410–421