Neural Computing and Applications (2023) 35:13093–13108
https://doi.org/10.1007/s00521-020-05684-y

S.I. : DEEP SOCIAL COMPUTING

A supervised and distributed framework for cold-start author disambiguation in large-scale publications

Yibo Chen[3] · Zhiyi Jiang[4] · Jianliang Gao[4] · Hongliang Du[4] · Liping Gao[1] · Zhao Li[2]

Received: 16 July 2020 / Accepted: 28 December 2020 / Published online: 5 March 2021

Abstract

Names make up a large portion of queries in search engines, while the name ambiguity problem brings negative effect to the service quality of search engines. In digital academic systems, this problem refers to a large number of publications containing ambiguous author names. Name ambiguity derives from many people sharing identical names, or names may be abbreviated. Although some methods have been proposed in the decade, this problem is still not completely solved and there are many subproblems needing to be studied. Due to lack of information, it is a nontrivial task to distinguish ambiguous authors accurately relying on limited internal information only. In this paper, we focus on the cold-start disambiguation task with homonymous author names, i.e., distinguishing publications written by authors with identical names. We present a supervised framework named DND (abbreviation for *Distributed Framework for Name Disambiguation*) to solve the author disambiguation problem efficiently. DND utilizes accessible information and trains a robust function to measure similarities between publications, and then determines whether they belong to the same author. In traditional clustering-based approaches for author disambiguation, the number of clusters which is the amount of authors sharing the same name is hard to predict in advance, while DND transforms the clustering task to a linkage prediction task to avoid specifying the number of clusters. We validate the effectiveness of DND on two real-world datasets. The experimental results indicate that DND achieves a competitive performance compared with the baselines.

Keywords Author disambiguation · Supervised · Distributed · Cold-start

✉ Liping Gao
gliping@hhtc.edu.cn

✉ Zhao Li
lizhao.lz@alibaba-inc.com

Yibo Chen
chenyibo8224@gmail.com

Zhiyi Jiang
zyjiang@csu.edu.cn

Jianliang Gao
gaojianliang@csu.edu.cn

Hongliang Du
duhongliang@csu.edu.cn

[1] Huaihua University, Huaihua, China

[2] Alibaba Group, Hangzhou, China

[3] Hunan Key Laboratory for Internet of Things in Electricity, Changsha, China

[4] Central South University, Changsha, China

1 Introduction

There are more than 7 billion people in the world. It is common that different people share the same name. There are even 109 people in the USA named Harry Potter![1] This brings a tremendous challenge to the name-related tasks. This similar scenario occurs in scientific publication management application more often, which is called the author ambiguity problem that a large number of publications contain common names in their author lists. These ambiguous names belong to the same or different individuals in the real world. Due to the author ambiguity problem, the performance of scientific and technical data retrieval is reduced. In the era of big data, researchers need to spend lots of time screening out useful literature for their researches from the massive data. When searching for references in Google Scholar, the results are numerous.

[1] http://howmanyofme.com, accessed July 1, 2020.

Filtering them by custom conditions may omit important results. When researchers lack relevant knowledge about what they search for, identifying the quality of papers by the citation number is usually the first choice, while this may lead to incomplete results. High quality of author disambiguation can benefit researchers and academic communities in many aspects, e.g., online scientific systems can establish profiles for the authoritative authors in various fields according to their institutions and the amount of their publications through the author disambiguation process. When researchers search for documents based on author names, accurate results related to different real-world authors can help them quickly find experts and scholars in a certain field, as well as reproduce and follow up their work.

Ambiguity is not existed only in online academic systems, it is a common phenomenon in people's daily life. For instance, some words may express different meanings in diverse contexts. Researchers specifically proposed solutions to this phenomenon. [1] proposed a model based on a multilayer perceptron and long short-term memory deep neural networks to solve the word sense disambiguation for Punjabi language. In visual fields, [2] proposed an unsupervised learning approach to model the distribution of pointing gestures using a growing-when-required (GWR) network to learn in cases of ambiguities resulting from close object proximity. Author disambiguation can be regarded as a branch of entity resolution from an extensive view. It has drawn researchers' attention for a long time. There are many applications on this problem, e.g., entity alignment in knowledge graphs [3], aligning proteins in protein–protein interaction networks [4], identifying users in different social networks [5]. Despite many approaches have been proposed to solve the problem, there are still many issues remaining to study. Most extant methods may be specifically designed for a dataset and the performance becomes unpredictable when scaling up to large-scale data.

Author disambiguation is divided into cold-start disambiguation and incremental disambiguation. Cold-start disambiguation can be understood as clustering all publications containing ambiguous author names from scratch. The target of this task is to split publications into different clusters in which each cluster represents a real-world person. Incremental disambiguation can be understood as assign publications newly added to the academic system to their corresponding author profiles. If a publication does not belong to any of the existing author profiles, a new profile for the author in this publication should be created. Incremental disambiguation is regarded as the downstream task of cold-start disambiguation. The establishment of correct profiles is essential to this task, which depends entirely on whether the existing publications are allocated into correct clusters in the first place. The main challenge of cold-start author disambiguation is the increasing literature and authors, which reduces the efficiency and accuracy of author disambiguation. An author's research field and institution may change, which leads to variations of the author's collaborators in different periods. These factors can cause errors and reduce the accuracy of disambiguation. Besides, if an author only publishes a few papers, it will be difficult to extract useful information from a small number of papers to distinguish the author from others, which will reduce the precision of the disambiguation process. If an author publishes a lot of papers, and the papers cover different topics, existing methods tend to divide them into different clusters, which reduces the recall of the disambiguation process.

Online scholar systems such as DBLP,[2] Microsoft Academic,[3] AMiner,[4] etc., have applied author disambiguation algorithms in their databases. DBLP provides a disambiguation page for search requests by author names. For example, if searching for the name "Wei Wang" in DBLP, users will get a pre-disambiguation list containing over 200 "authors" named "Wei Wang." Although DBLP has made a preliminary distinction of publications related to ambiguous author, its performance still has a lot of room for improvement. While the disambiguation strategy of Google Scholar is to require users to register and create their own profiles, which is accurate but inefficient. These academic databases have a fast update of data, including previewed and published papers from international journals and conferences, which reflect the forefront of global academic research scientifically. However, the data scale is quite large, their author disambiguation process does not cover all the ambiguous records, and the precision of the disambiguation results can still be improved. Obviously, distinguishing who is who in abundant publications is a challenging task. In this paper, we focus on the cold-start disambiguation task with homonymous names, i.e., distinguishing publications written by authors with identical names and present a supervised framework named DND (abbreviation for *DistributedFrameworkforNameDisambiguation*), to solve the author ambiguity problem systematically and efficiently. The main contributions of this paper are summarized as follows.

- We propose a distributed framework for author disambiguation in the academic literature. Its scalability is based on the implementation of Spark.[5] With the superiority on distributed computing of Spark, DND

[2] https://dblp.uni-trier.de.

[3] https://academic.microsoft.com/.

[4] https://www.aminer.cn.

[5] http://spark.apache.org/.

calculates the similarity between any two publications simultaneously. Its accuracy stems from our careful consideration and the utilization of different features in the publication records. Various feature similarity scores are weighted through the supervised learning.

- Differing from general clustering-based author disambiguation methods, our method avoids specifying the number of clusters by transforming the clustering task into a binomial classification task for publication pairs. Compared with those methods involving external data, DND depends on limited internal data to disambiguate authors accurately.
- Compared with the state-of-the-art method which is the backend of AMiner [6], experiment results on two real-world large datasets demonstrate that our method can disambiguate authors with a small number of publications whose related papers normally exist as isolated nodes in citation networks.

The rest of this paper is organized as follows. The motivation of our work is presented in Sect. 2. Section 3 introduces related work. Section 4 illustrates the definition of the problem studied in this paper through formulations, then follows the detailed demonstration of our method. Section 5 reports the experiments and results. Lastly, we conclude and discuss the future work in Sect. 6.

2 Motivation

With the continuous increment of scientific and technological publications, author ambiguity problem has become more complicated and difficult to solve. Because of the presence of homophonic characters in Chinese, diverse Chinese characters may be written the same in English, Chinese names are more likely to cause ambiguity than names in other regions. As more Chinese researchers get their work published in international conferences and journals, the author ambiguity problem becomes more common.

It is necessary to establish profiles for authors who have published a large number of papers, and newly added publications containing the names of these authors should be compared with their profiles first, which improves the efficiency of incremental author disambiguation. The obstacle for establishing profiles is the cold-start problem. Plenty of methods have been proposed in the past decade. AMiner [6] used representation learning to incorporate global and local information from the context. It also presents an end-to-end cluster number estimation strategy to enhance the effect of agglomerative clustering. [7] constructed three local graphs based on co-authorship and document similarity. It leverages topological information from networks in order to map each document into a low dimensional vector space and generate the final disambiguation result by agglomerative clustering. Typical approaches rely on information such as author affiliations, email addresses, co-authorship, research fields, publications topics to distinguish ambiguous authors. Some datasets for author disambiguation are built by crawling relevant meta-information of publications from web pages. It is common that some information in the metadata of publications is missing. For example, some authors' affiliations are unknown, abstracts and venues of some papers are unavailable. There is currently no perfect method for processing data lacking necessary information. Performing crawling for the missing data consumes additional costs, so a few approaches claim to use only co-author names for disambiguation [8], but this wastes other information which has been proved to be useful for improving the accuracy. Therefore, leveraging co-authorship as the main evidence and taking other information such as affiliation and venue can optimize the process of disambiguation. In this paper, we propose a supervised framework and implement it in a distributed way to make it extensible. Our method extracts features from accessible information which is the meta-information of a publication including title, abstract, author's name, affiliation, venue, etc., to measure the similarities between publications. The detail of our approach is presented in Sect. 4.

3 Related work

Author disambiguation for academic publications has gained continuous attention from research communities for years. Author disambiguation is regarded as a subproblem of entity disambiguation associated with semantic search and question and answering [9, 10]. Many related research works have been proposed, e.g., [11] proposed a rule-based algorithm based on word similarity or editing distance to improve existing techniques of institution author disambiguation (IND). [12] proposed a method to automatically generate labeled data using information features from publication records. [13] presented using simulations to study the effect of the quality of datasets produced by different author disambiguation processes on various bibliometric analysis. [14] proposed a supervised method for author disambiguation of Chinese patent inventors.

Author disambiguation in scientific literature becomes increasingly difficult as the number of publications and ambiguous author names keeps growing. Most previous works formulate author disambiguation as a statistical learning problem. However, most methods neglect the diversity of data sources. For example, when crawling data from academic system, some data are parsed incorrectly or some information is lost. Several proposed methods disambiguate authors only rely on the overlapping of co-author names between publications or the similarity between authors in the co-authorship graph. Such methods will bring errors to the disambiguation results for publications with only one author and isolated nodes. With the rapid development of machine learning technology in the field of artificial intelligence [15–20], many machine learning-based methods have been proposed to replace rule-based methods for the author ambiguity problem. In this paper, we roughly cataloged existing methods as machine learning-based (supervised and unsupervised) and graph-based. It should be noted that some works combine different types of methods, but we still classify them into certain categories according to their main parts.

3.1 Supervised methods

In the supervised approaches, the mainstream strategy of author disambiguation is to learn a similarity function to measure the probability that two publications belong to the same individual and train a classifier to generate results. [21] implemented a dynamic approach for similarity calculation based on all available data fields which creatively included differences in author contribution and publication year difference. [22] demonstrated the utility of word embedding-based semantic similarity methods for author disambiguation. [23] constructed high-qualitative training data from lists of rare names and evidence for the reliability of generated labels. [24] proposed an approach to author disambiguation from short text that integrates two models: entity co-occurrence and topic modeling. [25, 26] leveraged supervised learning method to learn a pairwise distance function between documents based on their feature vectors. [27] focused on the named entity disambiguation method based on context similarity. They analyzed the description of ambiguous entities and used context information as a prior probability to construct a classifier model to predict the class of ambiguous entities. According to the knowledge graph, the context of the candidate words and the context of the ambiguous words are mapped to the same low dimensional vector space. LOAD [28] exploited a supervised framework to train the similarity functions between publications, and a clustering algorithm is further applied to generate clusters. NameClarifier [29] quantified and visualized the similarities between ambiguous names in digital libraries. The similarities are calculated by co-authorship, venues, and temporal information. [30] used a classifier to learn pairwise similarity and performed semi-supervised hierarchical clustering to generate results. Moreover, [31] used a Dirichlet process prior to a Normal × Normal × Inverse Wishart data model which enables the identification of new ambiguous entities who have no record in the training data. [32] proposed an algorithm for pairwise disambiguation of author names based on random forests algorithm. [33] presented a supervised disambiguation method based on SVM and Naive Bayes.

3.2 Unsupervised methods

In the unsupervised approaches, clustering is basically the natural solution [25, 34–36]. [37] used semantic and relational information of papers jointly to obtain representation of papers. Semantic embedding is trained through a Word2Vec [38] model and relational embedding is generated from a heterogeneous network constructed by diverse relations between papers. Then, the joint embedding will be optimized through a variational auto-encoder. Finally, hierarchical agglomerative clustering (HAC) is applied to get the results. [39] proposed a hybrid framework using internal information such as co-authorship and applying web pages as a source of additional information. This framework utilized hierarchical clustering method with re-clustering mechanism to deeply process the publication records which cannot be found in any web pages and group them together if they refer to one person. [40] disambiguated author names for Chinese documents without external data based on semantic fingerprint. This method generated fingerprints of all publications based on their text, co-author names, and institutions. Results are generated by comparing fingerprint similarities between publications. As for clustering tasks, there are two main challenges needing to be addressed. The first challenge is how to quantify the similarity. The second challenge is how to determine the number of clusters. Most existing researches mainly focus on the former one, while ignoring the second by assuming the number of clusters is known beforehand. Several existing approaches claim to use clustering methods such as DBSCAN to avoid specifying K. However, several density-based hyper-parameters are still needed to be pre-specified. For determining the number of authors sharing the same name, AMiner [6] proposed an end-to-end model that directly estimated the number of persons (clusters) using a recurrent neural network. [41] used a variation in X-means [42] algorithm to iteratively estimate the optimal K by measuring clustering quality based on Bayesian information criterion (BIC). While on large-scale datasets, the BIC-based methods are inclined to merge clusters together, which results in low accuracy. On the whole, relevant approaches make efforts in two aspects to achieve potential enhancement. The first is to measure similarities between publications using different models and the second is to choose applicable clustering strategies to group publications into clusters. [43] offered an unsupervised Dempster–Shafer theory (DST) based on HAC algorithm. [44] proposed an enhanced vector space model for ambiguous authors and employed hierarchical clustering to get results. [45] designed a system consisting of similarity estimation and agglomerative clustering for ambiguous names. [26] proposed a two-stage clustering method to cluster documents by using only strong features in the first stage and revised them by using weak features in the second stage. [46] used blocking [25] technique to group candidate documents with similar names together. It learned the distance measurement between publications by the support vector machine (SVM) and employed DBSCAN to cluster publications.

3.3 Graph-based methods

In recent years, many researchers focus on graph-related problems [47–49] or convert the original data into graphs and apply graph-based models to solve the original problems [50, 51]. With the development of graph neural network in recent years [52, 53], it is possible to apply deep learning techniques on many problems relevant to unstructured data [47, 49, 54]. Graph-based methods have been proposed continuously to deal with author disambiguation since work relevant to graph started to attract the attention of researchers. Many graph-based approaches [55–58] are capable of utilizing graph topology and aggregating information from neighbor nodes. [48] proposed an unsupervised author disambiguation framework. They first constructed a publication heterogeneous network for each ambiguous name. Then, they used a heterogeneous graph convolutional network embedding method that encoded both graph structure and node attribute information to learn publication representations. The results were generated through an efficient graph enhanced clustering method which did not require the number of clusters. [59] constructed a graph model by using the co-author relations and resolved name ambiguities by graph operations such as vertex (or node) splitting and merging based on the co-authorship. AMiner [6] proposed a representation learning framework. It learned the global embedding of documents by supervised metric learning and refined the embedding through local linkage structures which projected each entity into a low dimensional latent common space for quantifying the similarity. They also involved human work into disambiguation to improve the accuracy. [60] used graph structural clustering and proposed similarity measure to resolve ambiguous authors. [61] made use of a probabilistic Markov random fields framework to solve the author disambiguation problem of the National Natural Science Foundation of China fund. [7] solved this problem by learning graph embedding from three constructed graphs based on document similarity and co-authorship. They leveraged relational data in the form of anonymized graphs and used a representation learning model to embed each document in a low dimensional vector space where author disambiguation was performed by a hierarchical agglomerative clustering algorithm. [62] executed author disambiguation task from timestamped link information obtained from a collaboration network. [63] provided the first probabilistic model to link the named entities in Web text with a heterogeneous information network. [64] introduced a pairwise factor graph (PFG) model called ADANA for author disambiguation. The model is flexible and has ability to incorporate various features. [41] employed hidden Markov random fields to model node and edge features in a unified probabilistic framework. They defined a disambiguation objective function for the problem and proposed a two-step parameter estimation algorithm. GHOST [8] built a document graph for each ambiguous name by co-authorship only, while excluding all other attributes such as email, venue, publication title, and author affiliation. GHOST [8] employed a valid path selection method to compute similarity between nodes and utilized affinity propagation clustering algorithm to get the disambiguation results.

In comparison with the canonical methods which learn the embedding of publications and generate results by clustering algorithms, our method does not require separate training processes for different names and it only needs to compare the new publications with the existing publications. For example, AMiner [6] needs to build local networks separately for diverse names based on the relevant publications. The connection between publications depends on whether the sum of the weights of the identical features between publications is greater than a specified threshold. Then, it trains on the local networks to optimize the global representations of publications. Retraining is required when new publications are added to the database, because the structure of the network changes. Compared with the graph-based methods which construct networks based on one type of relation between publications, our method fully considers the similarities of multiple features between publications such as affiliation and title. For example, GHOST [8] only utilizes co-authorship to construct networks. It may mistakenly regard isolated nodes in the network as different authors. Our method makes a more reasonable judgment than GHOST [8] on the publications lacking information of co-authors. In general, compared with general clustering-based methods, our method does not need to set the number of clusters of each name in advance. Compared with embedding-based methods, our method does not require separate data for each name. At the same time, our method has the ability to migrate to large-scale data.

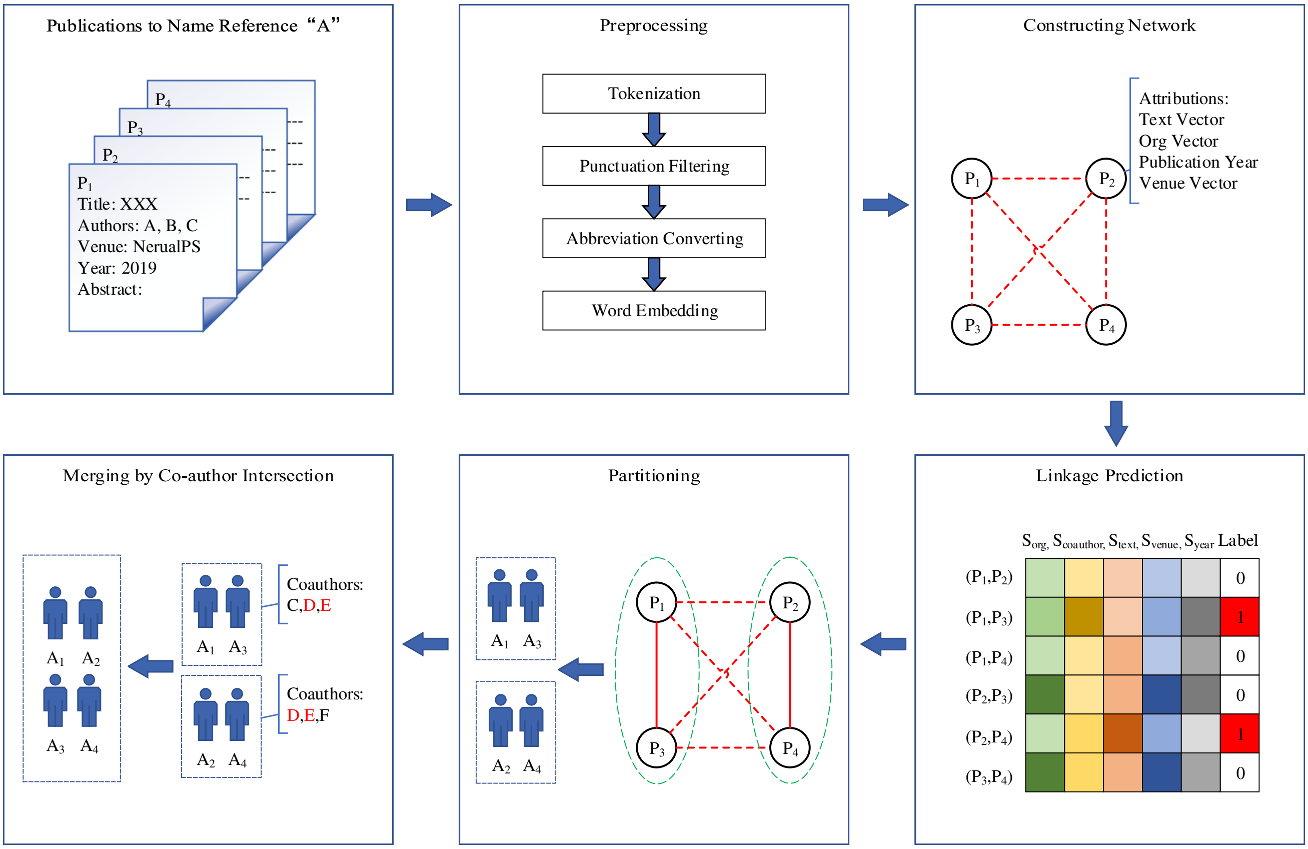


Fig. 1 An overview of the DND framework for author disambiguation. DND first completes preprocessing and extracts features from the raw data, then it constructs a fully connected publication network where vertices represent publications and “dashed line” denotes “ambiguous” relation between two authors in a publication pair. Then, DND calculates similarities between publications and predicts the class of “ambiguous” edges, which is a binomial classification task. In this stage, “solid line” denotes two authors are recognized as the same person by DND. Finally, DND merges initial partitions by a rule-based algorithm to get the disambiguation results

4 Methodology

We propose a supervised framework named DND and implement it in a distributed way for solving the author ambiguity problem in large-scale publication datasets. The objective of DND is to predict whether any two publications belong to the same author. In this section, we briefly describe the problem formulation in the first place, and then discuss the design and implementation of DND in detail. Finally, we analyze the shortcoming of DND. DND consists of four main parts, including feature extraction, constructing network, linkage prediction, and group merging. An overview of DND is shown in Fig. 1.

4.1 Problem formulation

We describe the task of author disambiguation as follows. Let a be an ambiguous name, and $P^a = \{p_1^a, p_2^a, \ldots, p_N^a\}$ be a set of N publications related to a. Each publication includes a set of attributes $p_i^a.X = \{x_1, x_2, \ldots, x_n\}$ including title, abstract, co-author names, publication year, venue, keywords, etc. We use $R(p_i^a)$ to denote the real-world individual of p_i^a. So $R(p_i^a) = R(p_j^a)$ means the authors of p_i^a and p_j^a are the same person. The purpose of our framework is to learn a function f to determine whether p_i^a and p_j^a are authored by the same author, i.e, $f(p_i^a, p_j^a) \in \{0, 1\}$, which is regarded as a linkage prediction task between two publications. Given this, we define the problem of author disambiguation as follows. We construct a publication network G^a for a name reference a, in which vertices represent publications and whether p_i^a and p_j^a are connected depends on the output of function $f(p_i^a, p_j^a)$. Vertices connected by edges in the network G^a form multiple connected components C^a, where $C^a = \{C_1^a, C_2^a, \ldots, C_k^a\}$. Each component only contains publications of the same person, i.e., $R(p_i^a) = R(p_j^a), \forall p_i^a \in C_m^a, \forall p_j^a \in C_m^a, 1 \le m \le k,$ and diverse components contain publications of different people, i.e., $R(p_i^a) \neq R(p_j^a), \forall p_i^a \in C_m^a, \forall p_j^a \in C_n^a, m \neq n.$ Above contents can be described by Eq. 1.

$$\begin{cases} (p_i^a, p_j^a, 1) \Rightarrow p_i^a \in C_m^a, p_j^a \in C_m^a, R(p_i^a) = R(p_j^a) \\ (p_i^a, p_j^a, 0) \Rightarrow p_i^a \in C_m^a, p_j^a \in C_n^a, R(p_i^a) \neq R(p_j^a) \end{cases} \quad (1)$$

All the pairs whose predicted class equals 1 construct many connected components in the publication network of the name reference a. Each component can be regarded as a cluster which only contains publications of the same identity.

4.2 Feature extraction

Data preprocessing is a pivotal prerequisite of feature extraction. It is of service to remove noisy information to facilitate the feature extraction from the plain text, reduce the vocabulary for lowering the computational cost of the disambiguation process, and improve the robustness of learned similarity function. The data preprocessing strategies we adopt are as follows.

- Because illegal characters existed in titles and abstracts of papers and punctuation that has no effect on semantics, we simply delete these kinds of characters.
- For the Asian names, the surname is in the first place and followed by the given name, which is opposite to the names in Western countries. To many Asian names, surname and given name are reversed due to journal requirements or personal writing errors, e.g., Wei Wang can be seen as Wang Wei in some publication records, which is a wrong way of writing in English but is valid in Chinese. We correct the order of all names according to the correspondence given in the dataset and make them conform to the writing habits of English. Besides, all names are converted to lowercase. Dots and dashes in names are replaced by underlines, e.g., both "S. Yang" and "S-Yang" are processed as "s_yang."
- Some affiliations contain abbreviations of particularly common words such as "department, school, university." These words appear in abbreviated form very often, i.e., "dept., sch., univ.". We revert these abbreviated words to their full form.

After data preprocessing is completed, we extract features of publications from the raw data. For a given publication p_i authored by n authors, we define r_i with $1 < j < n$ as the author's reference. The features we extract from p_j and r_i are shown in Table 1. We transform all the string-type attributes except co-author names to vector by training a Word2Vec [38] model on them.

Table 1 Features of each publication

Feature	Description
$p_j.text$	Vector of title and abstract
$r_i.org$	Vector of affiliation
$p_j.coauthors$	Array of co-authors' names
$p_j.year$	Year of publication
$p_j.venue$	Vector of venue

4.3 Constructing network

The framework constructs a full connected publication network based on the extracted features, in which vertices represent publications and edges denote that two publications contain the same ambiguous author name. We define the network as $G^a = (V, E)$. V is the vertex set, and E is the edge set. Each vertex represents a publication, and an edge denotes "ambiguous" relation between two authors in a publication pair related to the name reference a. Multiple similarities between vertices are calculated and updated to edges simultaneously.

4.4 Linkage prediction

Similarity function learning. The various feature similarities are calculated in different ways based on the feature types for pairs of publication records. All the similarity scores are combined into a vector, and logistic regression is applied to weigh diverse scores.

A feature similarity vector is composed of S_{text}, S_{org}, S_{year}, S_{coauthor}, S_{venue}. Let $\mathbf{X} = [x_1, x_2, \ldots, x_n]$ denotes the feature matrix with n representing the number of pairs. Given a point x_i from $\mathbf{X}$ for binomial classification, the model makes predictions by applying the logistic function in Eq. 2:

$$f(z) = \frac{1}{1 + e^{-z}} \quad (2)$$

where $z = w^T x_i$. If $f(w^T x_i) > threshold$, w^T is the feature coefficients. The positive output is marked by 1, and the negative output is marked by 0. The raw output of the logistic regression model $f(z)$ has a probabilistic interpretation, which is given by $P(y = 1|x_i : w) = f(w^T x_i)$. It is the probability that the pair of nodes belong to the same author.

The loss function added regularization to prevent overfitting is defined as Eq. 3.

$$L(w) = -\frac{1}{n}\sum_{i=1}^{n}[y^i \ln f(w^T x_i) + (1 - y^i) \ln (1 - f(w^T x_i))] + \frac{\lambda}{2n}\sum_{j=1}^{m} w_j^2 \quad (3)$$

In our approach, we calculate all kinds of similarity scores between publications in a distributed way, including S_{text}, S_{org}, S_{coauthor}, S_{year} and S_{venue}. The description of similarity scores is shown in Table 2. As shown in Eq. 4, S_{text}, S_{org}, S_{venue} are calculated by cosine similarity measurement.

Considering that authors tend to concentrate on a fixed research field and publish at several focused conferences/ journals intensively in a specific period, we define a function shown in Eq. 5 to calculate S_{year}, where γ is a hyper-parameter and we set it to 10 empirically. According to the formulation of S_{year} , the bigger the publication year difference, the smaller the S_{year} is, which conforms to the assertion that an author appears more likely in a specific period. In other words, S_{year} is inversely related to the publication year interval of two publications. S_{venue} is the venue similarity. We consider the venue similarity based on a hypothesis that an author tends to publish publications on journals or conferences in similar fields. Subjects of publications cannot be inferred only from venues, so we summarize the text (title and abstract) of all publications published on different venues and establish a mapping from venues to vectors of text related to venues. (i.e., the text is transformed to a vector as the representation of the venue through a Word2Vec [38] model.) We use S_t to denote S_{text}, S_{org}, S_{venue}. The calculation equations for each similarity score are shown from Eqs. 4 to 6.

$$S_t(V_i, V_j) = \frac{V_i.t^T V_j.t}{\|V_i.t\| \|V_j.t\|}, \quad \text{where } \|V_i.t\| = \sqrt{V_i.t^T V_i.t} \text{ and } t \in \{\text{text, org, venue}\} \tag{4}$$

$$S_{\text{year}}(V_i, V_j) = \exp\left(-\frac{|V_i.\text{year} - V_j.\text{year}|}{\gamma}\right) \tag{5}$$

$$S_{\text{coauthor}}(V_i, V_j) = \text{jaccard}(V_i.\text{coauthors}, V_j.\text{coauthors}) \tag{6}$$

Similarity scores shown in Table 2 on every edge form a feature vector as the input of the logistic regression model to predict the class of the edge.

Binomial classification. For each pair of publications, the predicted class being 1 means they belong to the same author, while the predicted class being 0 means they belong to the different authors.

Additionally, the classification algorithm can be replaced by other feasible algorithms, e.g., *Support Vector Machine* and *Random Forest*. Because some publications lack the author's organization, publication venue, and other information, the feature similarity vectors of these publications will contain zero values. We can use the kernel-based support vector machine algorithm to further explore the association between different features and improve the accuracy of classification for similarity vectors without zero values. This will be our future work.

Table 2 Similarities between publications

Similarity	Description
S_{text}	Cosine similarity of text vectors
S_{org}	Cosine similarity of author's organization vectors
S_{year}	Similarity of publication years
S_{coauthor}	Jaccard similarity of co-authors
S_{venue}	Cosine similarity of venue vectors

4.5 Group merging

The precision of initial results generated from the former steps is desirable, while the recall is low, which indicates the purity of clusters is high but some publications belong to the same person are split by mistake. So we designed some rules to merge clusters. The purpose is to make the difference between clusters as large as possible, and the difference between elements in the cluster as small as possible. The main rule is to merge clusters according to the number of identical author names between any two clusters. If the number of co-occurring authors in two clusters equals or exceeds the hyper-parameter k, these two clusters will be merged. Generally, when the number of papers increases, we also need to adjust the value of k. In our experiments, the k is set to 3 empirically. For publication records missing some information, we flexibly use remaining accessible information to group them accurately. For example, regarding the publication records that lack information about the authors'

Algorithm 1 Merging clusters by the overlap of co-authors in pairs of publications

```
Input:
    Components of $G_a$ constructed by DND, where $a$ is a name.
    Hyper-parameter $k$ denotes the minimum number of identical co-
    authors required for merging.
Output:
    Merged components. Each component denotes an author.
numComponents ← length(Components)
componentToAuthors ← []
for i, component ∈ enumerate(Components) do
    coauthors ← {}
    for all publication ∈ component do
        coauthors ← (authors ∈ publication)
    end for
    componentToAuthors[i] ← coauthors
end for
nodesNeedMerge ← {}
for i ∈ range(0, numComponents − 1) do
    for j ∈ range(i, numComponents) do
        intersection ← componentToAuthors[i] ∩
        componentToAuthors[j]
        if length(intersection) ≥ k then
            nodesNeedMerge ← (i, j)
        end if
    end for
end for
mergedGraph ← mergeNodes(G, nodesNeedMerge)
mergedComponents ← getComponents(mergedGraph)
return mergedComponents
```

affiliations and publication venues, DND can still complete the disambiguation process by measuring the co-author similarity and text similarity between them.

4.6 Discussion

Essentially, the objective of DND is to determine whether two publications are authored by the same person based on multiple feature similarities. We assume that an author usually does researches in a fixed institution with a small number of consistent authors for a period of time and publishes papers in journals and conferences in similar research fields. But there are extreme situations in real life, i.e., a researcher's affiliation and research field may change. The establishments of author profiles and additional human constraints are needed for such situation. DND struggled when dealing with such cases, and we take solving the dilemma as our future work.

5 Experiment

All code and data used in this work are publicly available.[6] In this section, we evaluate the performance of our method and the baselines on two real-world publication datasets which are significantly larger (in terms of the number of publication records) and more challenging (each name is related to much more authors) than others. Extensive experimental results manifest that our method can effectively and accurately complete author disambiguation in large-scale datasets in comparison with the baselines. In this section, we first give a brief introduction of the datasets and describe the detail of our baselines. Then, we discuss the experiment settings including evaluation metrics and the process of training the classifier. Finally, we verify the robustness and validity of the learned similarity function through experiments and discuss the scalability of our model by analyzing the experiment results.

5.1 Baselines

We compare DND with the state-of-the-art method [6] and two rule-based methods to systematically evaluate the performance of DND. Additionally, we combine our method with [6] and conduct experiments. We also mask some information in the datasets randomly to inspect the ability of different methods to identify authors with incomplete information of publication records.

AMiner[6]: There are four main stages in the framework proposed by [6]. The first stage is extracting all different types of information as features of an author in a publication, such as title, abstract, affiliation, venue, keywords, then it learns the global embedding of publications by training a Word2Vec model. In the second stage, instead of using contrastive loss to enforce positive pairs to a single point, AMiner [6] used triplet loss to make documents with the same identity to reside a manifold. The embedding function trained in this stage can further refine the global embedding. A triplet is extracted for the publications of each author reference. A triplet consists of an anchor node, a positive node, and a negative node. The anchor node and positive node belong to the same author, and the anchor node and negative node belong to different authors. Then, it uses a graph auto-encoder (GAE) to refine the embedding from both global and local context. They define the linkage between two publications when the sum of IDF value of the common features shared by them is over a threshold. The input of GAE includes a nodes embedding matrix and an adjacent matrix. The GAE consists of a node encoder and an edge decoder. The node encoder produces a refined embedding of the original node embedding, and the edge decoder predicts edges according to the generated embedding. The objective of GAE is to minimize the reconstruction error between the predicted $\tilde{A}$ and the original adjacency matrix A. Finally, it utilizes hierarchical agglomerative clustering to get the disambiguation results.

Rule-based methods: We conduct two kinds of rule-based methods as the baselines. The first method is clustering publications that depends on the number of common co-authors between different clusters, which is marked by *Rule (co-author)*. As for the second method, we construct local linkage graphs by connecting two documents when their affiliations are strictly matched, which is marked by *Rule (org)*. The clustering is obtained by simply partitioning the graph into connected components. The rule-based methods take publication records set P_a where a is a name as the input. The first step of Rule (org) method is taking each publication as an initial cluster. The second step is merging clusters with the same affiliation about author a. Clusters that do not have the same affiliation about author a as other clusters still remain separated. For each cluster whose publications lacking organizations of author a, calculate the number of identical co-authors between it and other clusters and merge it with the cluster which has the largest number of identical co-authors compared with itself. The affiliations of author a in the final generated clusters should be different. As for the Rule (co-author) method, take one publication of name a as a cluster in the beginning. Then, perform an iterative operation that counts the number of identical co-authors between each publication and existing clusters for all publications. Lately, add the publication to the cluster which has the largest number of identical co-authors

[6] https://github.com/xxx/xxx.

compared with itself. If there is no same co-author between the publication and existing clusters, take the publication as a new cluster. At last, each cluster is regarded as a real-world person.

5.2 Datasets

We assess our method on two publicly available datasets. The Dataset-1 is the same as [6] used. The Dataset-2 is from a author disambiguation competition held by biendata.[7] Notably, although there are incorrect labels to authors in the datasets, the experiment still reflects the effect of the methods since we use the same data for training and testing. There are also some publication records in the datasets missing some information, such as affiliation, venue, and abstract. The missing ratio of different types of information is shown in Table 3. Thus, the evaluation results are significantly different for the two datasets.

Dataset-1. The Dataset-1 consists of 203,078 scientific publications from 1,121,831 authors. We take the data of 200 names for training and the data of 100 names for testing.

*Dataset-2.*The Dataset-2 consists of 286,749 scientific publications from 5,398,890 authors. We take the data of 100 names for training and the data of 50 names for testing.

The statistical information of the datasets is shown in Table 4. Obviously, the averages of authors and papers are quite large, which indicates the author ambiguity problems in the two datasets are acute and challenging.

Table 3 Information missing ratio

	Org (%)	Venue (%)	Abstract (%)	Year (%)
Dataset-1	0.88	0.00	1.39	0.00
Dataset-2	31.71	3.99	36.30	0.25

Table 4 Statistics of the datasets

	#Authors/name	#Papers/name
Dataset-1	63.99	351.29
Dataset-2	100.46	940.10

5.3 Evaluation settings

Evaluation metrics. To more accurately assess the effectiveness of our proposed framework, we use two common evaluation indicators to evaluate our experimental results. Besides Pairwise Macro F1 used by most aforementioned works, we also evaluate our experiment results by Pairwise Macro F1. Here, we briefly introduce the difference and calculation process of these two indicators.

In the multi-class classification task, the final precision and recall of Macro F1 are computed based on the mean precision and recall of all classes, and the Macro F1 score is computed based on these two values.

As for Pairwise Micro F1, the final precision and recall are computed based on the total correct predicted pairs (TP), wrong predicted pairs (FP), and not found pairs (FN) of all classes. The Pairwise Micro F1 can accurately evaluate the model effect when the number of samples of different categories varies greatly. Because the numbers of publications related to diverse author names are greatly different, using the Pairwise Micro F1 for evaluation can better reflect the effect of our model.

$$\text{PairwisePrecision} = \frac{\#\text{PairsPredictedCorrectly}}{\#\text{PairsPredicted}} \tag{7}$$

$$\text{PairwiseRecall} = \frac{\#\text{PairsPredictedCorrectly}}{\#\text{Pairs}} \tag{8}$$

$$\text{Pairwise}F_1 = \frac{2 \times \text{PairwisePrecision} \times \text{PairwiseRecall}}{\text{PairwisePrecision} + \text{PairwiseRecall}} \tag{9}$$

Training the classifier. Every piece of the training data is composed of a label and a feature vector where each dimension is a kind of similarity score. The training data are generated from the publication pairs related to the sampled author names. These pairs can be divided into positive and negative pairs. Positive means that two publications belong to the same author, and negative means two publications belong to different authors. Negative pairs take a large proportion in the training data and many feature vectors are composed of lower similarity scores, while some negative pairs contain vectors composed of high similarity scores. The classifier should be able to cope with the interference caused by such pairs, so we utilize K-means clustering to divide the training data into two groups and sample data from the group in which each example consists of feature vector with higher value.

Because a portion of publication records lack some information, feature vector related to these publication records will have zero values in corresponding dimensions. If the predicted labels of these pairs are 1, it will interfere with the model training process, so we filter out such pairs. Besides, since the number of negative pairs is generally greater than the number of positive pairs, we randomly sample from remaining negative pairs to avoid overfitting.

[7] https://www.biendata.com/competition/aminer2019/.

For hyper-parameter tuning, we apply *TrainValidationSplit* to optimize *RegParam* (parameter for regularization, in range [0.01, 0.1]).

5.4 Results

AMiner [6] proposed a cluster number estimation method based on RNN. For simpleness, we skip the process of cluster number estimation and set the number of clusters to the real number of authors. Notably, DND transforms the clustering task to a linkage prediction task which is a binomial classification task, so our proposed method does not require to specify the cluster number previously.

As shown in Tables 5 and 6, the performance of our method is close to or better than other baselines. Obviously, our approach has achieved a higher level of balance. In Table 5, indicators of [6] drop slightly compared with those shown in their paper. The reason is that we do not filter authors with less than five publications while [6] did in their experiments. As Tables 7 and 8 show, DND achieves a better performance on disambiguating names related to a large number of publications and authors ($\#Publications \geq 300, \#Authors \geq 50$). However, the effect of our method for disambiguating the authors of the abbreviated names may be reduced, because the ambiguity of such names is more complicated. An abbreviated name may correspond to different names in reality, which undoubtedly increases the difficulty of the problem. For example, “D. Johnson” might be the abbreviation of “Daniel Johnson” or “David Johnson.” On the whole, our method (DND) outperforms the baselines in terms of Macro F1 score (+42.60% over AMiner [6], + 5.4% over Rule (co-author) and + 153.88% over Rule (org) relatively on the Dataset-1. + 8.53% over AMiner [6], + 105.71% over Rule (org) relatively on Dataset-2) and Micro F1 score (+53.55% over AMiner [6], + 15.11% over Rule (co-author) and + 181.75% over Rule (org) relatively on the Dataset-1. + 29.93% over AMiner [6], + 42.78% over Rule (co-author) and + 140.24% over Rule (org) relatively on the Dataset-2). We also conducted an ablation study to evaluate the effect of group merging. The result that DND outperforms DND w/o merging in terms of Macro-F1 (+ 8.53% on Dataset-1, + 33.75% on Dataset-2) shows that group merging of DND takes positive effect on the process of disambiguation.

Notably, although the Rule (co-author) obtained the highest macro-pR, macro-pF1, and micro-pR on the Dataset-2, it got the lowest macro-pP and micro-pP among all the methods. Since the name ambiguity problem of the Dataset-2 is more complicated than that of the Dataset-1, i.e., the average number of authors corresponding to each name is significantly more than that of the Dataset-1, identifying ambiguous authors only depending on co-author co-occurrence may group a large number of publications written by different authors into one cluster mistakenly. However, DND obtained the highest pairwise precision on the Dataset-2 (both macro and micro) by considering multiple features of publications to avoid the limitation of leveraging only co-author co-occurrence to identify ambiguous authors.

Masking information. We also tested the performance of our method for identifying ambiguous authors with information missing. We randomly selected papers in the Dataset-1 and blocked one or two types of information at a time according to a fixed ratio. The results indicate that our method still achieves competitive performance when a large portion of affiliation and text (title and abstract) information is missing. The experimental results also prove that our method can use affiliation, text, and other information to improve the accuracy. The results are shown in Table 9. Under the masking information setting, the F1 score of all methods declined, but the performance of DND is still maintained at a higher level than that of AMiner [6]. Compared with the experimental setting of masking information, the precision of DND w/o merging under the setting of no masking dropped a little, while the recall and F1 score were higher. This is because DND will utilize remaining accessible information when some information is missing, which may lead to an improvement of precision. For example, when lacking organization information of authors, DND depends on co-author similarity and text similarity between publications more to disambiguate authors. Publications with similar co-authors are likely to belong to the same author, but publications with low co-author similarity may not necessarily belong to the same author. This may increase the precision but reduce the recall, resulting in a significant decrease in the F1 score.

Evaluation of the similarity function. In order to further evaluate the accuracy of the similarity measure function obtained by the logistic regression model, we project the probability of predicted class equaling 1 for each publication pair into a similarity matrix. We take the similarity matrix as the input of affinity propagation (AP) algorithm. The results in Table 10 show that the F1 score and recall obtained by AP clustering are close to DND, but the precision is significantly lower than DND, indicating that the feature weights obtained by training are effective.

Embedding analysis. In order to further evaluate the utility of DND, we project the embedding of documents generated by Word2Vec into a two-dimensional Euclidean space which can be easily visualized. Figure 2 shows the t-SNE plot of the embedding of a publication set where each point is a publication record. In Fig. 2a, the color of a point denotes the corresponding ground truth cluster, while in Fig. 2b, c, the color denotes the cluster predicted by author disambiguation approaches.

Table 5 Experiment results on the dataset-1

Method	Macro-pP	Macro-pR	Macro-pF1	Micro-pP	Micro-pR	Micro-pF1
AMiner [6]	77.52	32.49	45.79	83.77	31.39	45.67
Rule (co-author)	68.35	**56.65**	61.95	63.06	58.92	60.92
Rule (org)	**81.47**	15.27	25.72	**83.99**	14.61	24.89
DND w/o merging	80.88	53.27	64.24	80.35	62.31	**70.19**
DND	78.22	56.04	**65.30**	75.64	**65.37**	70.13

The optimal value of each column is bolded

Table 6 Experiment results on the dataset-2

Method	Macro-pP	Macro-pR	Macro-pF1	Micro-pP	Micro-pR	Micro-pF1
AMiner [6]	72.34	47.08	57.03	72.01	15.29	25.22
Rule (co-author)	58.70	**66.15**	**62.21**	14.90	**49.96**	22.95
Rule (org)	83.74	18.34	30.09	79.21	7.46	13.64
DND w/o merging	**88.69**	31.30	46.28	**82.17**	16.01	26.81
DND	79.96	50.50	61.90	47.17	25.11	**32.77**

The optimal value of each column is bolded

Table 7 Experiment results of five sampled names on dataset-1

Name	#Pubs	#Authors	DND			AMiner [6]		
			Macro-pP	Macro-pR	Macro-pF1	Macro-pP	Macro-pR	Macro-pF1
Dandan Zhang	347	130	76.72	48.56	59.47	61.37	23.30	33.78
Yang Shen	700	157	80.48	43.27	56.28	68.04	20.62	31.65
Jing Luo	682	174	68.26	42.97	52.74	56.69	18.52	27.92
Lei Song	879	159	55.09	86.13	67.20	61.23	33.27	43.12
Jie Jiang	689	115	70.67	51.39	59.51	61.73	27.56	38.11

Table 8 Experiment results of five sampled names on dataset-2

Name	#Pubs	#Authors	DND			AMiner [6]		
			Macro-pP	Macro-pR	Macro-pF1	Macro-pP	Macro-pR	Macro-pF1
Di Wang	1425	112	99.34	56.94	72.39	60.95	48.90	54.26
Bin Yao	1340	83	87.60	57.20	69.21	83.96	31.25	45.55
Hong Li	3481	283	78.56	42.33	55.01	53.42	23.19	32.34
Bo Shen	1797	57	47.85	81.27	60.23	84.48	46.90	60.32
Feng Gao	2663	92	69.22	65.80	67.46	67.33	46.97	55.34

Table 9 Experimental results on the dataset-1 of masking information

Mask Conf.	DND w/o merging			DND			AMiner [6]		
	Macro-pP	Macro-pR	Macro-pF1	Macro-pP	Macro-pR	Macro-pF1	Macro-pP	Macro-pR	Macro-pF1
80% org	83.07	48.33	61.11	80.09	52.31	63.29	76.03	31.49	44.54
80% text	83.45	46.39	59.63	80.38	51.32	62.64	76.49	30.70	43.82
No masking	80.88	53.27	64.24	78.22	56.04	65.30	77.52	32.49	45.79

Table 10 Evaluation of learned feature weights

Method	ma-pP	ma-pR	ma-pF1
DND	78.22	56.04	65.30
Affinity propagation w/o merging	69.00	57.72	62.86

Table 11 Results of combining DND with Zhang et al

Method	Dataset-1		Dataset-2	
	ma-pF1	mi-pF1	ma-pF1	mi-pF1
Zhang et al.	45.79	45.67	**57.03**	25.22
Zhang et al. w/ DND	**59.63**	**76.84**	48.48	**25.80**

The optimal value of each column is bolded

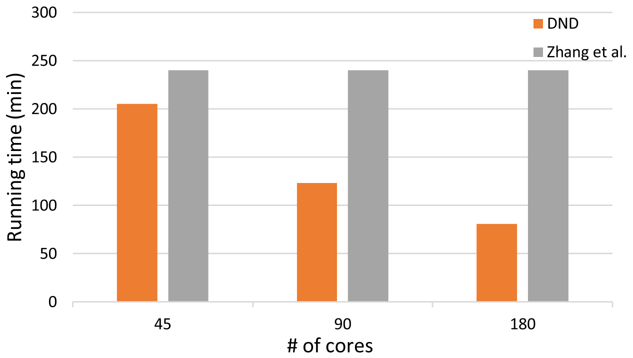
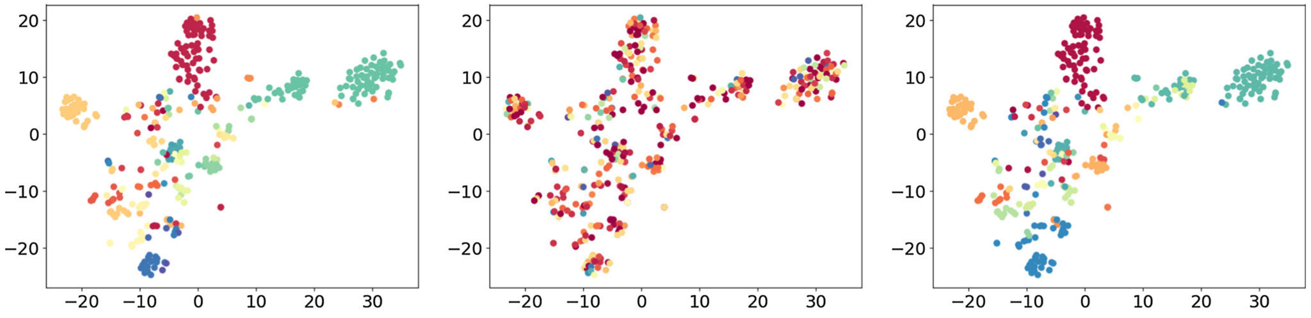


Fig. 3 The running time of DND in different quantities of cores. The running time of DND decreases significantly with the increase of cores in the cluster, and it is less than the running time of the program of AMiner [6] which only runs on a single machine

Combining DND with Zhang et al. AMiner [6] first extracted the relevant attributes of records and added category tags for them, then trained a Word2Vec [38] model to obtain the global vector representations of publication records. Secondly, they sample triples which are composed of an anchor node, a positive node, and a negative node from the training data. Each node represents a record. Positive and negative are defined by whether they belong to the same author as the anchor node does. The purpose of triplet training is to find a boundary to make the embedding of papers to a real person as close as possible, and the embedding of papers to different people as far as possible. Lately, they use a graph auto-encoder to further optimize the learned representation of papers and finally get disambiguation result by agglomerative clustering. The input network of GAE is constructed according to the feature similarity defined by the sum of IDF values of common features in two publications. That is, when the summing IDF value of the common feature between two publications exceeds a threshold, they will be connected by an edge. We replace the edges with our predicted edges and remove the triplets training stage. The results displayed in Table 11 show that our strategy makes the performance of AMiner [6] significantly improve on Dataset-1.

5.5 Scalability

In this section, we explain why our model is suitable for solving the name ambiguity in large-scale scientific publication datasets. We implement our framework on Spark and Python, respectively. The Spark cluster we built for experiments consists of five nodes. Each node has a Intel(R) Xeon(R) CPU E5-2650 v3 @ 2.30GHz and 128G RAM. With the advantage of distributed computation, the program of our method on Spark has a better performance on effectiveness than that of Python. In our model, the publication network for each specified name is constructed by Spark GraphX which is recognized as an effective distributed graph calculation tool. Multiple similarities between nodes are calculated by *MapTriplets* function which takes effect on every edge independently and synchronously, then we utilize a logistic regression model to



(a) Ground Truth **(b)** AMiner [6] (F1: 68.53%) **(c)** DND (F1: 77.40%)

Fig. 2 t-SNE visualization of the embedding space on a publication set associated with a name reference. Each color represents an individual ground-truth cluster in (**a**), while each color in (**b**), (**c**) represents a predicted cluster predict the labels of edges. By filtering edges with label equaling 0, nodes connected by edges with label equaling 1 are composed of many connected components and each component represent an author. Lastly, a rule-based group merging algorithm is applied to fine-tune the disambiguation result. The similarity calculation and binomial classification are conducted in parallel. The trend of running time on different quantity of cores is described in Fig. 3. Since the program of [6] runs on a single machine, it is time-consuming for large datasets. It takes over 200 minutes for [6] to finish the whole disambiguation process on one machine in the Spark cluster. Compared with its running time, our model is more time-saving and has better performance when running on a distributed cluster, which can be seen from Fig. 3. It only takes 81 minutes for DND to disambiguate all the sampled ambiguous names in the Dataset-1 when our approach runs on a cluster containing five workers which have 180 cores, which is significantly faster than AMiner [6]. All in all, our model can be applied to large-scale scientific publication datasets benefiting from distributed computation technology, which is an advantage compared with other methods.

6 Conclusion

In this paper, we aim at the cold-start author disambiguation problem and focus on overcoming the shortcomings of existing methods for author disambiguation, i.e., they are not effective to disambiguate authors who only publish a few papers. Besides, they do not take full advantage of all available features in the publication records to disambiguate authors. We propose a framework which extracts multiple features in publication records and learns a robust similarity function to measure similarities between publications. We implement this framework through Spark, so it has the advantage of distributed computing. Our framework is capable of handling the author ambiguity problem in large-scale datasets efficiently. We compare our method with several baselines, and the results demonstrate that our framework is competitive in accuracy and scalability. We plan to involve more human constraints and exploit implicit information to improve the disambiguation process as the future work.

Acknowledgements This work was supported in part by National Natural Science Foundation of China under grant no. 61873288, in part by CERNET Innovation Project (NGII20190517); in part by Technology Projects, Hunan Key Laboratory for Internet of Things in Electricity (2019TP1016); in part by the Fundamental Research Funds for the Central Universities of Central South University (2020zzts594).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest regarding the contents of present article.

References

1. pal Singh V, Kumar P (2020) Word sense disambiguation for Punjabi language using deep learning techniques. Neural Comput Appl 32:2963–2973
2. Jirak D, Biertimpel D, Kerzel M, Wermter S (2020) Solving visual object ambiguities when pointing: an unsupervised learning approach. Neural Comput Appl. https://doi.org/10.1007/s00521-020-05109-w
3. Wang Z, Zhang J, Feng J, Chen Z (2014) Knowledge graph embedding by translating on hyperplanes. In: Proceedings of AAAI conference on artificial intelligence, pp 1112–1119
4. Gao J, Tian L, Lv T, Wang J, Song B, Hu X (2019) Protein2vec: aligning multiple ppi networks with representation learning. IEEE/ACM Trans Comput Biol Bioinform 19(3):571–578
5. Zhang J, Philip SY (2015) Multiple anonymized social networks alignment. In: Proceedings of IEEE international conference on data mining. IEEE, pp 599–608
6. Zhang Y, Zhang F, Yao P, Tang J (2018) Name disambiguation in aminer: clustering, maintenance, and human in the loop. In: Proceedings of ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 1002–1011
7. Zhang B, Al Hasan M (2017) Name disambiguation in anonymized graphs using network embedding. In: Proceedings of ACM international conference on information and knowledge management. ACM, pp 1239–1248
8. Fan X, Wang J, Pu X, Zhou L, Lv B (2011) On graph-based name disambiguation. J Data Inf Qual (JDIQ) 2(2):10
9. Shen J, Xiao J, He X, Shang J, Sinha S, Han J (2018) Entity set search of scientific literature: an unsupervised ranking approach. In: Proceedings of ACM SIGIR conference on research and development in information retrieval. ACM, pp 565–574
10. Zwicklbauer S, Seifert C, Granitzer M (2016) Robust and collective entity disambiguation through semantic embeddings. In: Proceedings of ACM SIGIR conference on research and development in information retrieval. ACM, pp 425–434
11. Huang S, Yang B, Yan S, Rousseau R (2014) Institution name disambiguation for research assessment. Scientometrics 99(3):823–838
12. Kim J, Kim J, Owen-Smith J (2019) Generating automatically labeled data for author name disambiguation: an iterative clustering method. Scientometrics 118(1):253–280
13. Schulz J (2016) Using Monte Carlo simulations to assess the impact of author name disambiguation quality on different bibliometric analyses. Scientometrics 107(3):1283–1298
14. Yin D, Motohashi K, Dang J (2020) Large-scale name disambiguation of Chinese patent inventors (1985–2016). Scientometrics 122(2):765–790
15. Krizhevsky A, Sutskever I, Hinton G E (2012) Imagenet classification with deep convolutional neural networks. In: Proceedings of international conference on neural information processing systems. Curran Associates Inc., pp 1097–1105
16. Devlin J, Chang M-W, Lee K, Toutanova K (2018) Bert: pretraining of deep bidirectional transformers for language understanding. arXiv1810.04805, pp 1–14
17. Singh M, Kumar R, Chana I (2020) Improving neural machine translation for low-resource Indian languages using rule-based feature extraction. Neural Comput Appl. https://doi.org/10.1007/s00521-020-04990-9

18. Teles G, Rodrigues JJPC, Saleem K, Kozlov S, Rabêlo RAL (2020) Machine learning and decision support system on credit scoring. Neural Comput Appl 32:9809–9826
19. Hou R, Kong Y, Cai B, Liu H (2020) Unstructured big data analysis algorithm and simulation of internet of things based on machine learning. Neural Comput Appl 32:5399–5407
20. Zhang Y, Wu J, Zhou C, Cai Z (2017) Instance cloned extreme learning machine. Pattern Recognit 68:52–65
21. Gurney T, Horlings E, Van Den Besselaar P (2012) Author disambiguation using multi-aspect similarity indicators. Scientometrics 91(2):435–449
22. Müller M-C (2018) On the contribution of word-level semantics to practical author name disambiguation. In: Proceedings of ACM/IEEE joint conference on digital libraries, pp 367–368
23. Yin D, Motohashi K (2018) Inventor name disambiguation with gradient boosting decision tree and inventor mobility in China (1985–2016). Technical report, Research Institute of Economy, Trade and Industry
24. Ju Y, Adams B, Janowicz K, Hu Y, Yan B, McKenzie G (2016)Things and strings: improving place name disambiguation from short texts by combining entity co-occurrence with topic modeling. In: Proceedings of European knowledge acquisition workshop. Springer, pp 353–367
25. Steorts RC, Ventura SL, Sadinle M, Fienberg SE (2014) A comparison of blocking methods for record linkage. In: Proceedings of international conference on privacy in statistical databases. Springer, pp 253–268
26. Yoshida M, Ikeda M, Ono S, Sato I, Nakagawa H (2010) Person name disambiguation by bootstrapping. In: Proceedings of ACM SIGIR international conference on research and development in information retrieval. ACM, pp 10–17
27. Zhang K, Zhu Y, Gao W, Xing Y, Zhou J (2018) An approach for named entity disambiguation with knowledge graph. In: Proceedings of international conference on audio, language and image processing. IEEE, pp 138–143
28. Qian Y, Hu Y, Cui J, Zheng Q, Nie Z (2011) Combining machine learning and human judgment in author disambiguation. In: Proceedings of ACM international conference on information and knowledge management. ACM, pp 1241–1246
29. Shen Q, Wu T, Yang H, Wu Y, Qu H, Cui W (2016) Nameclarifier: a visual analytics system for author name disambiguation. IEEE Trans Vis Comput Graph 23(1):141–150
30. Louppe G, Al-Natsheh HT, Susik M, Maguire EJ (2016) Ethnicity sensitive author disambiguation using semi-supervised learning. In: Proceedings of international conference on knowledge engineering and the semantic web. Springer, pp 272–287
31. Zhang B, Dundar M, Al Hasan M (2016) Bayesian non-exhaustive classification a case study: Online name disambiguation using temporal record streams. In: Proceedings of ACM international on conference on information and knowledge management. ACM, pp 1341–1350
32. Treeratpituk P, Giles CL (2009) Disambiguating authors in academic publications using random forests. In: Proceedings of ACM/IEEE joint conference on digital libraries. ACM, pp 39–48
33. Han H, Giles L, Zha H, Li C, Tsioutsiouliklis K (2004) Two supervised learning approaches for name disambiguation in author citations. In: Proceedings of ACM/IEEE joint conference on digital libraries. IEEE, pp 296–305
34. Pooja KM, Mondal S, Chandra J (2018) An unsupervised heuristic based approach for author name disambiguation. In: Proceedings of international conference on communication systems and networks. IEEE, pp 540–542
35. Kim J (2018) Evaluating author name disambiguation for digital libraries: a case of DBLP. Scientometrics 116(3):1867–1886
36. Zhu J, Wu X, Xueqin Lin, Huang C, Fung GPC, Tang Y (2018) A novel multiple layers name disambiguation framework for digital libraries using dynamic clustering. Scientometrics 114(3):781–794
37. Xiong B, Bao P, Wu Y (2020) Learning semantic and relationship joint embedding for author name disambiguation. Neural Comput Appl. https://doi.org/10.1007/s00521-020-05088-y
38. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Proceedings of international conference on neural information processing systems. Curran Associates Inc., pp 3111–3119
39. Zhu J, Yang Y, Xie Q, Wang L, Hassan S-U (2014) Robust hybrid name disambiguation framework for large databases. Scientometrics 98(3):2255–2274
40. Han H, Yao C, Fu Y, Yu Y, Zhang Y, Xu S (2017) Semantic fingerprints-based author name disambiguation in chinese documents. Scientometrics 111(3):1879–1896
41. Tang J, Fong ACM, Wang B, Zhang J (2011) A unified probabilistic framework for name disambiguation in digital library. IEEE Trans Knowl Data Eng 24(6):975–987
42. Pelleg D, Moore AW et al (2000) X-means: extending k-means with efficient estimation of the number of clusters. In: Proceedings of international conference on machine learning, vol 1, pp 727–734
43. Wu H, Li B, Pei Y, He J (2014a) Unsupervised author disambiguation using Dempster–Shafer theory. Scientometrics 101(3):1955–1972
44. Arif T, Ali R, Asger M (2014) Author name disambiguation using vector space model and hybrid similarity measures. In: Proceedings of international conference on contemporary computing. IEEE, pp 135–140
45. Liu W, Doğan RI, Kim S, Comeau DC, Kim W, Yeganova L, Lu Z, Wilbur WJ (2014) Author name disambiguation for pubmed. J Assoc Inf Sci Technol 65(4):765–781
46. Huang J, Ertekin S, Giles CL (2006) Efficient name disambiguation for large-scale databases. In: Proceedings of European conference on principles of data mining and knowledge discovery. Springer, pp 536–544
47. Wu J, Pan S, Zhu X, Zhang C, Wu X (2016) Positive and unlabeled multi-graph learning. IEEE Trans Cybern 47(4):818–829
48. Qiao Z, Du Y, Fu Y, Wang P, Zhou Y (2019) Unsupervised author disambiguation using heterogeneous graph convolutional network embedding. In: 2019 IEEE international conference on big data (Big Data), pp 910–919
49. Li Z, Sun Y, Zhu J, Tang S, Zhang C, Ma H (2020) Improve relation extraction with dual attention-guided graph convolutional networks. Neural Comput Appl. https://doi.org/10.1007/s00521-020-05087-z
50. Wu J, Pan S, Zhu X, Cai Z (2014b) Boosting for multi-graph classification. IEEE Trans Cybern 45(3):416–429
51. Wu J, Zhu X, Zhang C, Philip SY (2014) Bag constrained structure pattern mining for multi-graph classification. IEEE Trans Knowl Data Eng 26(10):2382–2396
52. Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. In: Proceedings of international conference on learning representations, pp 1–14
53. Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y (2018) Graph attention networks. In: Proceedings of international conference on learning representations, pp 1–12
54. Huang W, Qu Q, Yang M (2020) Interactive knowledge-enhanced attention network for answer selection. Neural Comput Appl. https://doi.org/10.1007/s00521-019-04630-x
55. Rozenshtein P, Bonchi F, Gionis A, Sozio M, Tatti N (2020) Finding events in temporal networks: segmentation meets densest subgraph discovery. Knowl Inf Syst 62:1611–1639
56. Chen Z, Chen F, Lai R, Zhang X, Lu C-T (2018) Rational neural networks for approximating jump discontinuities of graph

convolution operator. In: Proceedings of IEEE international conference on data mining. IEEE, pp 406–415

57. Yang C, Feng Y, Li P, Shi Y, Han J (2018) Meta-graph based hin spectral embedding: methods, analyses, and insights. In: Proceedings of IEEE international conference on data mining. IEEE, pp 657–666
58. Hermansson L, Kerola T, Johansson F, Jethava V, Dubhashi D (2013) Entity disambiguation in anonymized graphs using graph kernels. In: Proceedings of ACM international conference on information and knowledge management. ACM, pp 1037–1046
59. Shin D, Kim T, Choi J, Kim J (2014) Author name disambiguation using a graph model with node splitting and merging based on bibliographic information. Scientometrics 100(1):15–50
60. Hussain I, Asghar S (2018) Author name disambiguation by exploiting graph structural clustering and hybrid similarity. Arab J Sci Eng 43(12):7421–7437
61. Si HJ, Tong W, Kausar S (2018) A conditional random field model for name disambiguation in national natural science foundation of china fund. J Algorithms Comput Technol 12(2):91–100
62. Saha TK, Zhang B, Al Hasan M (2015) Name disambiguation from link data in a collaboration graph using temporal and topological features. Soc Netw Anal Min 5(1):11
63. Shen W, Han J, Wang J (2014) A probabilistic model for linking named entities in web text with heterogeneous information networks. In: Proceedings of ACM SIGMOD international conference on management of data. ACM, pp 1199–1210
64. Wang X, Tang J, Cheng H, Philip SY (2011) Adana: active name disambiguation. In: Proceedings of international conference on data mining. IEEE, pp 794–803