Multiple Features Driven Author Name Disambiguation

Qian Zhou[1], Wei Chen[1,*], Weiqing Wang[2], Jiajie Xu[1], Lei Zhao[1,*]
[1]*Institute of Artificial Intelligence, School of Computer Science and Technology, Soochow University, Suzhou, China*
[2]*Faculty of Information Technology, Monash University, Melbourne, Australia*
Email: qzhou0@stu.suda.edu.cn, {robertchen, xujj, zhaol}@suda.edu.cn, Teresa.Wang@monash.edu

***Abstract*—Author Name Disambiguation (AND) has received more attention recently, accompanied by the increase of academic publications. To tackle the AND problem, existing studies have proposed many approaches based on different types of information, such as raw document feature (e.g., co-author, title, and keywords), fusion feature (e.g., a hybrid publication embedding based on raw document feature), local structural information (e.g., a publication's neighborhood information on a graph), and global structural information (e.g., the interactive information between a node and others on a graph). However, there has been no work taking all the above-mentioned information into account for the AND problem so far. To fill the gap, we propose a novel framework namely MFAND (Multiple Features Driven Author Name Disambiguation). Specifically, we first employ the raw document and fusion feature to construct six similarity graphs for each author name to be disambiguated. Next, the global and local structural information extracted from these graphs is fed into a novel encoder called R3JG, which integrates and reconstructs the above-mentioned four types of information associated with an author, with the goal of learning the latent information to enhance the generalization ability of the MFAND. Then, the integrated and reconstructed information is fed into a binary classification model for disambiguation. Note that, several pruning strategies are applied before the information extraction to remove noise effectively. Finally, our proposed framework is investigated on two real-world datasets, and the experimental results show that MFAND performs better than all state-of-the-art methods.**

***Index Terms*—author name disambiguation, multiple features, binary classification, pruning strategy**

I. Introduction

We have witnessed the unprecedented growth of academic digital records in the past decade [1] [2]. The latest estimation presents that there are more than 271 million publications, 133 million scholars, and 754 million citations on Aminer [3]. These numbers are much larger than those on Google Scholar database, Digital Bibliography & Library Project (DBLP), and Microsoft Academic [4] [5]. In these databases, many publications meet the AND problem which is referred to as object distinction [6] and name identification [7]. The AND problem can cause inconvenience in data mining communities and academic information retrieval [8] [9] [10]. For instance, an online search in DBLP for "Michael Jordan" may retrieve professors coming from UC Berkeley, Germany Helmut Schmidt University, Glasgow Caledonian University, and other academic affiliations simultaneously. This phenomenon has been a common problem for many online digital libraries. Making the matter worse, the growth of publications and researchers shows an unprecedented increase in recent years, thus the AND problem has been a pressing task [11].

Addressing the above-mentioned problem means splitting a set of publications into different clusters, and each cluster represents publications published by a unique person [12] [13] [14]. Many existing studies have made great contributions in this domain, and they roughly use the following four types of information, such as raw document feature (e.g., co-author, title, and keywords), fusion feature (e.g., a hybrid publication embedding based on raw document feature), local structural information (e.g., a publication's neighborhood information on a graph), and global structural information (e.g., the interactive information between a node and others on a graph). For instance, some methods employ the fusion feature to generate the context embedding and extract global structural information [12] [13], or disambiguate author names based on this feature directly [15] [16] [14]. [17] [9] employ raw document feature to construct similarity graphs, and extract local structural information based on these graphs to model high-order connections that can capture publications' neighborhood information.

The former approaches [15] [16] [12] [13] developed based on fusion feature and global structural information cannot capture a publication's neighborhood information, due to the lack of local structure information. The latter approaches [17] [9] that only use raw document feature and local structural information have met the problem: *missing of raw document feature*, which represents the phenomenon that some publications only have a part of raw document features such as titles and authors, and other features (e.g., venue and keywords) are missing. For example, keywords and abstract are often missing for some older publications in Aminer [3]. This problem brings a great challenge for measuring similarity between publications with raw document feature, thus the local structural information, which is extracted from neighbors on the similarity graph constructed based on raw document feature, cannot characterize an author precisely. Although the raw document feature, fusion feature, and local structural information are taken into account in [9], the global structure information, which can be used to further precisely measure the similarity between two publications from the global perspective, is out of consideration.

The limitations of the existing methods are that they merely use a part of the information, and have not solved the

*Corresponding author

AND problem effectively due to the problem *missing of raw document feature*. To overcome the drawbacks of existing studies, we propose a unified framework namely MFAND, where all types of information are taken into account, and a novel generating strategy is proposed to extract local structure information from fusion feature and extract global structural information from raw document feature. Our framework can solve the problem *missing of raw document feature* to a certain extent and disambiguate author name more precisely. Specifically, extracting local structure information from the fusion feature leads to a better representation of a publication, since the fusion feature denotes the interaction information between raw document features in one publication and quantifies the similarity of pairwise publications robustly [13]. For example, if a publication does not have the raw document feature co-author, the similarity between the publication and other publications based on co-author can not be quantified. And the publication will be an isolated node in the graph $G_{co-author}$ constructed based on co-author. On the contrary, we consider the fusion feature as it contains more information even one raw document feature is missing, and this feature can be used to estimate the similarity of pairwise publications robustly. Additionally, the novel encoder R3JG involved in MFAND can integrate and reconstruct the above-mentioned four types of information to enhance the generalization ability of MFAND by learning the latent information.

Specifically, the proposed framework MFAND consists of the following three components. (1) Construction of raw document feature graph: we construct five similarity graphs based on five raw document features (i.e., co-author, affiliation, venue, title, and keywords). Each node of a graph denotes a publication. Each edge's weight of the graph is the similarity of pairwise publications calculated based one raw document feature. For instance, we construct a similarity graph $G_{co-author}$, where each edge's weight is the similarity of pairwise publications calculated based the raw document feature co-author. We calculate the similarity between one publication and all other publications written by the author with the same name during the construction, which allows our method to measure the similarity from the global perspective. (2) Construction of fusion feature graph: each node of the graph denotes a publication and each edge is the similarity of pairwise publications calculated based the fusion feature(i.e., generated in the form of context that uses co-author, affiliation, venue, title, and keywords to concatenate simply). Following this construction, we apply the random walk on this fusion feature graph to generate walks that represent the local structural information of nodes. The local structural information is represented by walks that can be considered as nodes' neighborhood information, and we show the details in Section III-C. (3) Construction of the R3JG encoder and triplets decoder: an encoder namely R3JG is designed to integrate and reconstruct the above information to enhance the generalization ability of MFAND by learning the latent information. The R3JG consists of three JACNN (**Joint multiple features information with addition and concatenation of graph convolutional network**) layers with a residual block. The JACNN, which is designed for reconstructing and integrating all information, is comprised of two modules: *Concatenation* and *Addition*, where *Concatenation* is used to reconstruct the raw document feature and global structural information, and *Addition* is designed to rebuild fusion feature and local structural information. In the triplets decoder, the nodes' embeddings and the edges' embeddings constitute the triplets and they are fed into Multilayer Perceptron (MLP) to disambiguate whether two publications belong to the unique author. Additionally, we use different pruning strategies on different feature similarity graphs to remove noise effectively and details are introduced in Section III-B and Section III-C.

Our main contributions are outlined as follows:

- We propose a unified framework called MFAND, where multiple features information are taken into account, with the goal of addressing the AND problem more effectively.
- We design a novel generating strategy that extracts local structure information from fusion feature, and extracts global structural information from raw document feature to solve the problem *missing of raw document feature* to a certain extent. Our framework consists of a novel module namely R3JG that integrates and reconstructs the multiple features information to enhance the generalization ability of MFAND by learning the latent information, and a triplets decoder that formulates the AND problem as a binary classification task. Moreover, when raw document and fusion feature graphs are constructed, different pruning strategies are designed to remove noise effectively.
- We evaluate the framework namely MFAND on two real-world datasets. The experimental results verify the advantages of our method over state-of-the-art methods. Our codes are publicly available on github[1].

This paper is structured as follows. Section II shows the definition of the AND problem and similarity graph. Section III discusses the solution of dealing with the AND problem. Section IV shows the performance evaluation results and the comparative results. Section V presents the related work. In Section VI, conclusions and future work are discussed.

II. Preliminaries

A. *Problem Definition*

Formally, given an author name a, $P^a = \{p_1^a, p_2^a, ..., p_N^a\}$ is a set of N publications associated with the author name a. And $p_i^a = \{x_1, x_2, ..., x_K\} \in P^a$ is a publication that contains a set of raw document features and x_i is a raw document feature, such as co-author, affiliation, title, keywords, or venue. The fusion feature is generated in the form of context that uses all of these raw document features to concatenate simply. We use $\Psi(p_i^a, p_j^a)$ to describe whether the p_i^a and p_j^a have the same identity (corresponding real-world person) [12] or not. If p_i^a and p_j^a have the same identity, we have $\Psi(p_i^a, p_j^a) = 1$. Otherwise, we have $\Psi(p_i^a, p_j^a) = 0$. In addition, we omit the superscript a in the following description if there is no

[1]https://github.com/wx-qzhou/MFAND.

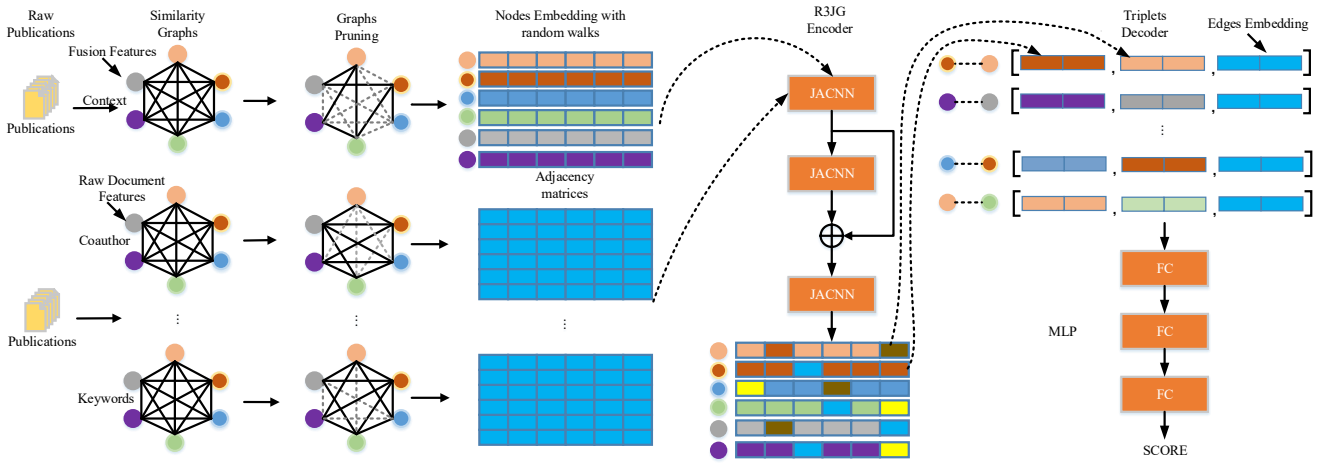

Fig. 1. Overview of the proposed framework.

ambiguity, e.g., $P^a \rightarrow P$, $p_i^a \rightarrow p_i$. Given this, we define the problem of author name disambiguation as follows.

Author Name Disambiguation. Given a publication set $P = \{p_1, p_2, ..., p_N\}$ associated with author name a, the task of author name disambiguation is to find a function Θ to partition P into a series of disjoint clusters based on multiple features information, such as raw document feature, fusion feature, local structural, and global structural information, i.e.,

$$\Theta(P) \rightarrow C = \{c_1, c_2, ..., c_k\}, \tag{1}$$

where C is the set of disjoint clusters, c_k is the cluster that only contains publications of the same identity, i.e., $\forall(p_i, p_j) \in c_k \times c_k, \Psi(p_i, p_j) = 1$, and different clusters contain publications of different identities, i.e., $\forall(p_i, p_j) \in c_k \times c_{k'}, k \neq k', \Psi(p_i, p_j) = 0$.

B. *Similarity Graph*

Given a publication set $P = \{p_1, p_2, ..., p_N\}$ associated with author name a, we construct a similarity graph $G_x = (D, E, x, S_x, w)$, where D and E are the set of nodes and edges in the graph respectively. In detail, $D_i \in D$ is a node and represents the publication p_i. $E_{ij} \in E$ is an edge and represents that two publications p_i and p_j have a certain degree of similarity. The feature x is a raw document feature (e.g., co-author or affiliation) or the fusion feature. S_x is the similarity function to quantify the similarity of pairwise publications based on feature x. The weight w_{ij} is the similarity of pairwise publications p_i and p_j calculated by S_x, and if pairwise publications p_i and p_j do not have the feature x or the intersecting set based on the feature x, the weight w_{ij} is zero. The similarity graph is an undirected weighted graph. Therefore, we have $E_{ij} = w_{ij} = E_{ji} = w_{ji}$. And each node has a self-loop, which is an edge that connects a vertex to itself. We show an example of similarity graph in Fig. 2. Note that, we construct six similarity graphs, i.e., $G_{co-author}$, $G_{affiliation}$, G_{title}, $G_{keywords}$, G_{venue} and G_{fusion} based on five common raw document features and the fusion feature.

III. Proposed Framework

In this section, we introduce the proposed framework MFAND in detail as shown in Fig. 1. We first describe how to estimate the similarity of pairwise publications by IDF (Inverse Document Frequency) based on raw document and fusion feature, which is a fundamental part to construct the similarity graph. Then, we construct five similarity graphs based on five raw document features (i.e., co-author, affiliation, venue, title, and keywords), and each similarity graph is used to extract the global structural information. This can make our method measure the similarity between two publications from the global perspective. Next, modeling the similarity graph based on the fusion feature is represented, and this similarity graph is applied to extract the local structural information and can represent the neighborhood information of one node better. Finally, we show how to perform the encoder and decoder to integrate multiple features information for disambiguation.

A. *Similarity of Pairwise Publications*

Given a publication set $P = \{p_1, p_2, ..., p_N\}$ associated with author name a, $Z^x = \{z_1^x, z_2^x, ..., z_{\kappa}^x\}$ is the set of the words based on feature x in these publications, where each word z_i^x in Z^x is not repetitive and x can be a raw document feature (e.g., co-author or affiliation) or the fusion feature. $Y^x = \{y_1^x, y_2^x, ..., y_{\kappa}^x\}$ is the set of the weight of the words in Z^x, where y_i^x calculated by IDF is the weight of the word z_i^x. And [12] has proved that IDF is a good choice to embed the words. $\tilde{Y}^x = \{\tilde{y}_1^x, \tilde{y}_2^x, ..., \tilde{y}_{\kappa}^x\}$ is the normalized representation of Y^x. Given two publications p_i and p_j, $Z_i^x \subset Z^x$ and $Z_j^x \subset Z^x$ are the word sets of p_i and p_j based on feature x respectively, where each word is not repetitive. The sum of normalized weight of the words in $Z_i^x \cap Z_j^x$ is utilized to denote the pairwise similarity between two publications, and we formalize it as follows.

$$Sim(p_i, p_j)^x = \sum_{z_m^x \in (Z_i^x \cap Z_j^x)} \tilde{y}_m^x, \tag{2}$$

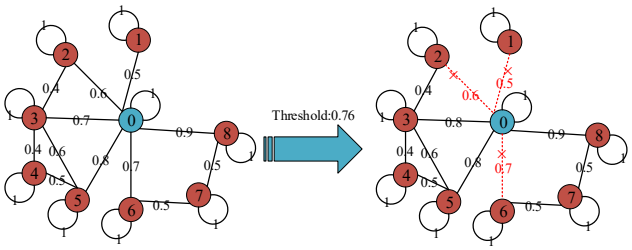

Fig. 2. An example of the similarity graph and the graph pruning.

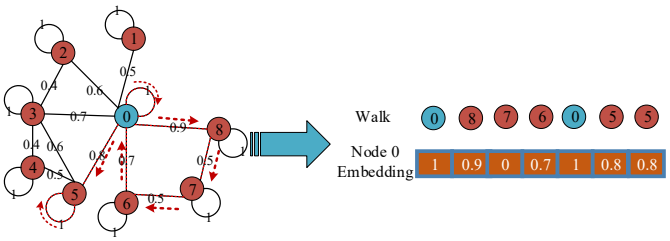

Fig. 3. An example of embeddings of nodes.

where $\tilde{y}_m^x$ is the normalized weight of the word z_m^x. Note that, $Sim(p_i, p_j)^x$ is set to zero on condition that $Z_i^x \cap Z_j^x = \emptyset$.

B. Construction of Raw Document Feature Graph

Based on the similarity of pairwise publications, we employ five raw document features (i.e., co-author, affiliation, venue, title, and keywords) to build five similarity graphs. Five similarity graphs are used to extract the global structural information, which makes our method measure the similarity between two publications from the global perspective.

Raw Document Feature Graph: Let $P = \{p_1, p_2, ..., p_N\}$ be a set of publications written by authors with name a, we employ a $N \times N$ adjacent matrix $\hat{M}^x$ to denote the raw document feature graph $G_x = (D, E)$, where x is one of five raw document features. $\hat{M}^x{}_{ij}$ is the similarity of pairwise publications p_i and p_j, and denotes the weight of edge E_{ij} between D_i and D_j.

If the similarity of pairwise publications is low, it represents that the edges are weak between pairwise publications and these edges are considered as noise. We adopt a similar pruning strategy like [18] to prune these weak edges for each raw document feature graph.

Raw Document Feature Graph Pruning [18]: The $N \times N$ adjacent matrix $\hat{M}^x$ denotes the raw document feature graph G_x, where x is one of five raw document features. The pruned adjacent matrix M^x is defined as:

$$\tilde{M}_{ij}^x = \begin{cases} 0, & \text{if } \hat{M}_{ij}^x < \frac{\sum_{m=0}^{N-1} \hat{M}_{im}^x}{N}, \\ \hat{M}_{ij}^x, & \text{else}, \end{cases} \tag{3}$$

$$M_{ij}^x = \frac{\tilde{M}_{ij}^x + \tilde{M}_{ji}^x}{2}. \tag{4}$$

This pruning strategy filters some weak edges by setting a threshold which is the mean value of each row and column. Specifically, the elements of $\hat{M}^x$, which are less than the threshold, can be set to zero. We give an example in Fig. 2 for the process. The pruned adjacent matrix M^x is the edges' embeddings and denotes the global structural information.

C. Construction of Fusion Feature Graph

Apart from the raw document feature graphs, we also build a fusion feature graph based on the fusion feature and the similarity of pairwise publications. The fusion feature is generated in the form of context that uses the five raw document features (i.e., co-author, affiliation, venue, title, and keywords) to concatenate simply. The fusion feature graph aims at extracting the local structural information. In order to represent the local structural information precisely, we must hold information that is more relevant to the current node on intuition. Hence, the following pruning strategy is adopted for the fusion feature graph to filter some weak edges.

Fusion Feature Graph Pruning: For a given $N \times N$ adjacent matrix $\hat{M}^f$, the element of $\hat{M}^f$ is the similarity of pairwise publications based on fusion feature. The process of calculation is shown in Section III-A. f is the fusion feature, which uses the five raw document features (i.e., co-author, affiliation, venue, title, and keywords) to concatenate simply. The pruned adjacent matrix M^f is defined as:

$$\theta = \frac{\alpha \times \frac{\sum_{m=0}^{N-1} \hat{M}^f{}_{im}}{N} + \max(\hat{M}^f{}_i)}{\alpha + 1}, \tag{5}$$

$$\tilde{M}^f{}_{ij} = \begin{cases} 0, & \text{if } \hat{M}^f{}_{ij} < \theta, \\ \hat{M}^f{}_{ij}, & \text{else}, \end{cases} \tag{6}$$

$$M_{ij}^f = \frac{\tilde{M}^f{}_{ij} + \tilde{M}^f{}_{ji}}{2}, \tag{7}$$

where θ is a pruning threshold for each row. α is a parameter that balances the mean and maximum to get the better pruning results.

Fusion Feature Graph: For a given pruned $N \times N$ adjacent matrix M^f, a fusion feature graph $G_f = (D, E)$ is built by the triples, i.e., if M_{ij}^f is nonzero, the triple (D_i, D_j, E_{ij}) is built, where D_i and D_j are the nodes of graph and represent the documents d_i and d_j, E_{ij} is the weight of edge between D_i and D_j, and E_{ij} equals to M_{ij}^f.

After constructing the fusion feature graph, the random walk [19], which can capture the neighborhood information of nodes, is used to construct the walks on the fusion feature graph. And the nodes' embeddings are generated by refining these walks and represent the local structural information. An example of this process is shown in Fig. 3.

Embeddings of Nodes: We denote the embeddings of nodes $D = \{D_i\}$ as $V = \{V_i\}$. For a given fusion feature graph $G_f = (D, E)$, we first utilize random walk to generate some walks $\Omega_i = \{\omega_j^i\}$ for each node D_i, and one walk ω_j^i consists of some nodes around the node D_i, i.e., $\omega_j^i = \{D_{i1}, D_{i2}, ...\}$. For the node D_i and its random one walk $\omega_j^i = \{D_{i1}, D_{i2}, ...\}$, this walk is redefined by using the weight of the edges between

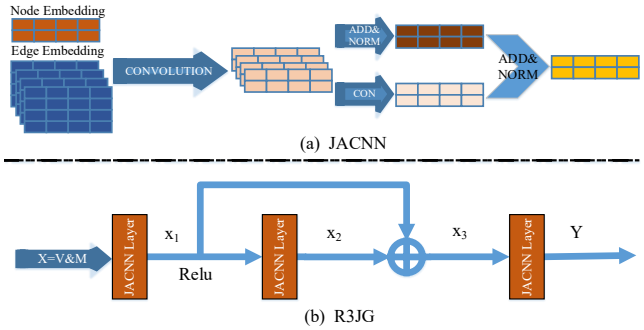

Fig. 4. The architecture of JACNN and R3JG module.

D_i and the nodes in this walk. Then, this redefined walk is employed to embed the node D_i, i.e., $V_i = \{v_k\}$, $v_k = E_{ik}$.

D. *Supervised Graph Neural Network model*

In addition to the generation of the above four types of information, another core part of our framework is how to integrate multiple features information for disambiguation. To achieve the goal, we design an encoder that can integrate and reconstruct the above four types of information to enhance the generalization ability of MFAND by learning the latent information. We design the decoder that makes the AND problem be a binary classification between each pairwise publications for disambiguation.

Normalization: Firstly, row normalization is used to normalize each $N \times N$ pruned raw document feature adjacency matrix M^x, where x is one of five raw document features, i.e.,

$$M_{ij}^x = \frac{M_{ij}^x}{\sum_{m=0}^{N-1} M_{im}^x}. \tag{8}$$

Then, a set of the normalized pruned raw document feature adjacency matrices are denoted by $\mathcal{M} = \{M^x\}$.

Encoder: The core part of the encoder is the **Joint multiple features information with addition and concatenation of graph convolutional network** (JACNN), which is comprised of two components: **Addition** and **Concatenation**. The encoder is shown in Fig. 4 and formalized as follows.

The JACNN can integrate and reconstruct the above four types of information to enhance the generalization ability of MFAND. Specifically, we first employ the convolutional operation to integrate the four types of information. Then, we use **Addition** and **Concatenation** to enhance the generalization ability of MFAND by reconstructing the four types of information. The Concatenation operation is considered to reconstruct the raw document feature and global structural information. The Addition operation is considered to reconstruct the fusion feature and local structural information. Next, we feed the all reconstructed information into the Addition and Normalization layer to integrate. We formalize them as below.

$$V_{Con}(V^k) = Con_x(M^x V^{k-1}) W_{Con}^k, \tag{9}$$

$$V_{Add}(V^k) = Norm(Add_x(M^x V^{k-1})) W_{Add}^k, \tag{10}$$

$$V^k = \sigma(Norm(V_{Con}(V^k) + V_{Add}(V^k))), \tag{11}$$

where x represents one of five raw document features, Con denotes the concatenation operation, Add denotes the addition operation, $Norm$ is the normalization operation. σ is an activation function, and RELU is used. V^k denotes the nodes' embeddings matrix of the k^{th} layer. W_{Con}^k and W_{Add}^k are the parameter of the Concatenation and Addition operation in the k^{th} layer.

Since [20] introduces a deep residual learning framework to address the degradation problem, we introduce a residual block in the encoder to reduce the loss of information during the training process. The nodes' embeddings V and $\mathcal{M}$ are fed into an encoder namely R3JG which consists of three JACNN layers with a residual block. We formalize them as below,

$$V^{k+1} = \sigma(Norm(V_{Con}(V^k + V^{k-1}) + V_{Add}(V^k + V^{k-1}))). \tag{12}$$

Decoder: We employ the embeddings of pairwise nodes V_i and V_j with corresponding edges' embeddings $Con_x(M_{ij}^x)$ to make a triplet T_m. Next, the generated triplets are fed into a MLP classifier to disambiguate. We formalize it as follows.

$$T_m = Con[V_i, V_j, Con_x(M_{ij}^x)], \tag{13}$$

$$\mathcal{P}_r = \text{MLP}(T), \tag{14}$$

where T is the concatenated triplets matrix and $m \in \{0, N^2\}$. $\mathcal{P}_r$ denotes the N^2 vector of predicted probabilities and is used to distinguish whether pairwise publications belong to a unique author or not.

For the convenience of calculation, we define a graph $G_c = (D, E)$ as the ground truth for the given document set $P = \{p_i\}$ associated with each author name and its annotation result $C = \{c_1, c_2, ..., c_k\}$ [16] [18]. From C, we first generate positive edge set E_p and negative edge set E_n. i.e., $E_p = \{E_{ij} = 1, \forall(p_i, p_j) \in c_k \times c_k, c_k \in C\}, E_n = \{E_{ij} = 0, \forall(p_i, p_j) \in c_k \times c_{k'}, k \neq k'\}$, and G_c consists of $E_p \cup E_n = E$ and $D_i \in D$ that denotes the publication p_i. Then, the $N \times N$ adjacent matrix of the ground truth graph G_c is compressed into N^2 vector Q as labels. Note that, the elements of the adjacent matrix of the ground truth graph G_c are $E_{ij} \in E$, where E_{ij} can be 0 or 1.

E. *Training*

The model is trained by minimizing the negative log-likelihood loss function. Given an author name a, the loss function of the model is defined as:

$$L = -\frac{1}{N} \sum_{m=0}^{N-1} (q^{(m)} log(p_r^{(m)}) + (1 - q^{(m)}) log(1 - p_r^{(m)})), \tag{15}$$

where $p_r \in \mathcal{P}_r$ is the vector of predicted probabilities and $q \in Q$ is the binary label vector.

We optimize the objective with the Adam optimization algorithm [21] simultaneously. The process for the author name disambiguation is summarized in Algorithm 1.

TABLE I
THE DETAILED RESULTS ON OAG-WHOISWHO

	Size	*MFAND*			*GANAND*			*AMiner*			*Beard*			*AGAND*		
Name	**-**	***Prec***	***Rec***	***F1***	***Prec***	***Rec***	***F1***	***Prec***	***Rec***	***F1***	***Prec***	***Rec***	***F1***	***Prec***	***Rec***	***F1***
Akio_Kobayashi	229	81.91	79.87	80.88	68.98	53.12	60.02	72.89	67.96	70.34	85.94	73.63	79.31	94.77	90.69	**92.68**
Yutaka Shimada	307	97.96	94.35	**96.12**	74.32	72.5	73.4	76	62.89	68.83	90.47	85.54	87.94	97.09	40.53	57.19
Xiaoming_Xie	479	88.18	90.51	89.33	74.3	57.84	65.04	87.77	93.58	**90.59**	84.87	74.58	79.4	90.17	17.48	29.28
Suqin Liu	518	92.76	99.68	**96.09**	84.87	49.59	62.6	95.19	60.54	74	89.88	55.75	68.82	80.91	17.9	29.32
Wensheng_Yang	540	96.6	83.17	**89.39**	79.27	54.42	64.54	95.99	56.84	71.4	86.88	86.22	86.55	76.16	6.49	11.97
Junyi Li	611	99.92	92.85	**96.26**	73.64	26.15	38.59	98.17	85.65	91.48	85.41	99.2	91.79	94.41	19.57	32.42
Feng Deng	703	99.29	99.11	**99.2**	74.62	48.86	59.05	99.2	50.48	66.91	97.84	77.29	86.36	82.28	11.54	20.24
Dan Wu	887	95.93	71.84	**82.15**	62.83	22.75	33.4	71.97	37.61	49.4	90.4	49.71	64.15	81.23	48.88	61.03
Xiaodong He	895	92.67	90.74	**91.69**	74.33	57.62	64.92	94.66	56.82	71.01	86.43	59.41	70.42	81.22	19.22	31.09
Xiaohua Liu	1335	98.09	98.4	**98.25**	68.02	35	46.22	95.3	91.3	93.26	98.51	69.38	81.42	82.87	13.58	23.34
Weimin Liu	1485	76.32	96.17	**85.1**	72.31	41.29	52.56	98.33	44.41	61.18	77.43	86.08	81.53	91.36	13.3	23.22
Min Yang	2245	68.96	68.06	**68.51**	66.14	18.76	29.22	56.08	19.91	29.39	76.26	32.14	45.22	74.41	24.12	36.43
Li Yang	3200	74.78	47.57	**58.15**	68.28	35.52	46.73	63.1	17.78	27.74	66.94	32.38	43.65	70.81	35.07	46.9
Avg.	-	**89.49**	**85.56**	**87.01**	72.45	44.11	54.83	84.97	57.37	68.49	85.94	67.79	74.35	84.44	27.57	41.56

Algorithm 1: The proposed framework: MFAND

Data: Publication set $P^a = \{p_1^a, p_2^a, ..., p_N^a\}$
Result: $\Theta(P^a) \rightarrow C^a = \{c_1^a, c_2^a, ..., c_k^a\}$

```
Evaluate similarity of pairwise publications according to Eq. (2);
Utilize the pruning strategy according to Eq. (3) and (4) to construct raw document feature graphs based on the five raw document features, and generate the edges' embeddings set M = {M^x};
Utilize the pruning strategy according to Eq. (5), (6) and (7) to construct the fusion feature graph;
Perform random walk on the fusion feature graph and generate the nodes' embeddings V = {V_i};
while model not converge do
    Feed M and V into the module R3JG with JACNN to integrate and reconstruct four types of information acorrding to Eq. (9), (10), (11) and (12);
    Put the integrated and reconstructed information into the triplets decoder to disambiguate, according to Eq. (13) and (14);
    Update all parameters according to Eq. (15) and Adam optimization algorithm;
end
```

Output: Use $\Psi(p_i^a, p_j^a)$ to distinguish whether p_i^a and p_j^a belong to a unique author or not, then achieve $\Theta(P^a) \rightarrow C^a$.

IV. EXPERIMENT

A. *Datasets*

We use two real-world AND datasets to evaluate our experiments: OAG-WhoisWho[2] and AMiner-AND[3]. The OAG-WhoisWho dataset contains $608,363$ documents and $57,138$ distinct authors with 642 equivocal author names [22]. In the OAG-WhoisWho dataset, we sample 320 author names and split them into 200, 60, 60 for the training, validating, and testing [18], which contains $341,457$ publications. Each publication has six features, such as co-author, affiliation, venue, year, title, and keywords. The AMiner-AND dataset is released by [12], which contains 500 author names as the training set and 100 author names as the testing set, and includes $203,078$ publications. Each publication has seven features, such as co-author, affiliation, venue, year, title, abstract, and keywords.

TABLE II
EXPERIMENT DATA STATISTICS FOR LOSING DIFFERENT FEATURES

	OAG-WhoisWho		***AMiner-AND***	
Features	***Num***	***Per***(%)	***Num***	***Per***(%)
co-author	295	0.09	-	-
affilication	112048	32.81	134	0.07
venue	25375	7.43	-	-
year	-	-	3	0.001
keywords	140522	41.15	49132	24.19

We analyze the datasets to illustrate the phenomenon: *missing of raw document feature*. From table II, we find that the publications, which lose the co-author, the affiliation, the venue, and the keywords, are 0.09%, 32.81%, 7.43%, and 41.15% of the OAG-WhoisWho dataset respectively. The publications missing the affiliation, the year, and the keywords are 0.07%, 0.001%, and 24.19% of the AMiner-AND dataset respectively. No publications lose the title and year on the OAG-WhoisWho dataset, and no publications lose the co-author, title, and venue on the AMiner-AND dataset. As the OAG-WhoisWho dataset does not have the abstract, this feature is not considered.

TABLE III
DATA STATISTICS FOR LOSING DIFFERENT NUMBER OF FEATURES

	OAG-WhoisWho		AMiner-AND	
Number	***Num***	***Per***(%)	***Num***	***Per***(%)
1	110566	32.38	49155	24.2
2	63038	18.46	57	0.03
3	13866	4.06	-	-

[2]https://www.aminer.cn/billboard/whoiswho.
[3]https://www.aminer.cn/na-data.

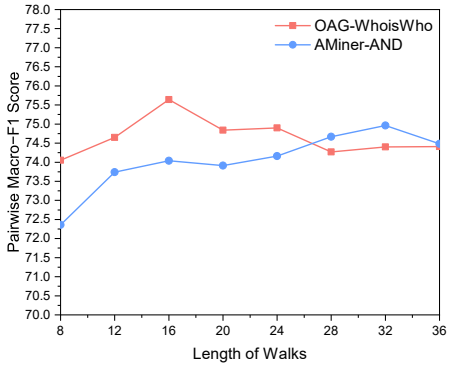

Fig. 5. The effect of different length of random walks of nodes for the result of the AND problem.

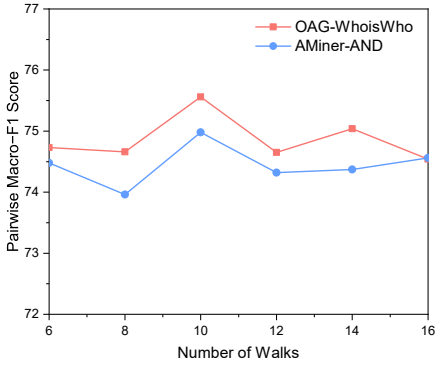

Fig. 6. The effect of different number of random walks of nodes for the result of the AND problem.

Observed from table III, there are $110,566$ publications losing one raw document feature, $63,038$ losing two raw document features, and $13,866$ losing three raw document features, which are 32.38%, 18.46%, and 4.06% of the OAG-WhoisWho dataset respectively. There are $49,155$ publications losing one raw document feature and 57 losing two raw document features, which are 24.2% and 0.03% of the AMiner-AND dataset respectively. For the AMiner-AND dataset and OAG-WhoisWho dataset, the maximum amount of the raw document features of publications, which are missing, is three. Therefore, we show that the statistical results are the number of publications that lose one, two, and three raw document features.

From these statistics, the problem of the *missing of raw document feature* is very serious, and the publications that lose the raw document feature are 54.9% and 24.23% of the OAG-WhoisWho dataset and AMiner-AND dataset respectively. Hence, it is necessary to make full use of and integrate multiple features information in the raw publications in a proper way.

B. Baselines

To validate the performance of our proposed approach, we compare our method with four state-of-art author name disambiguation methods.

Beard [16]: This method trains a distance function to measure the similarity between each pair of papers by a set of well-designed similarity features, including author names, titles, institute names, etc. A semi-supervised HAC algorithm is used to determine clusters.

AGAND [17]: This method builds three graphs based on document similarity and co-author relationship, and the triplets, which are employed to improve graph embedding, are sampled from these graphs. The final result is generated by agglomerative hierarchical clustering.

AMiner [12]: This method designs a supervised global stage to fine-tune the word2vec result, and designs an unsupervised local stage based on the first stage.

GANAND [9]: This method builds a generative adversarial framework. The discriminative module distinguishes whether two papers are from the same author, and the generative module selects possibly homogeneous papers from the heterogeneous information network.

TABLE IV
RESULTS OF AUTHOR NAME DISAMBIGUATION ON OAG-WHOISWHO

	Macro			*Micro*		
Model	***Prec***	***Rec***	***F1***	***Prec***	***Rec***	***F1***
AGAND	78.52	35.56	48.95	75.97	22.19	34.35
Beard	78.59	56.66	65.84	79.82	56.65	66.27
AMiner	78.39	60.69	68.41	80.83	21.05	33.4
GANAND	69.43	42.69	52.88	67.16	36.34	47.16
MFAND	74.91	76.39	**75.64**	71.95	75.58	**73.72**

C. Experiment Results

We evaluate our method by using pairwise precision, recall, and F_1 on each sampled author name and using micro-precision, micro-recall, micro-F_1, macro-precision, macro-recall, and macro-F_1 over the whole testing set. Table I shows the performance of different AND methods on sampled author names of different sizes, and the sampled author names are sampled from the OAG-WhoisWho dataset. Benefiting from the multiple features information and the module of R3JG, our method outperforms other state-of-the-art methods at most samples, and the average of F_1 of 13 samples outperforms others at least $+12.66\%$.

TABLE V
RESULTS OF AUTHOR NAME DISAMBIGUATION ON AMINER-AND

	Macro		
Model	***Prec***	***Rec***	***F1***
AGAND	70.63	59.53	62.81
Beard	57.09	77.22	63.1
AMiner	77.96	63.03	67.79
GANAND	82.23	67.23	72.92
MFAND	81.39	69.47	**74.96**

We verify the advantages of our method over state-of-the-art methods by experimental results on two real-world datasets. Table IV and Table V demonstrate that our method outperforms other baselines by at least $+7.23\%$ in macro-F_1 score and $+7.45\%$ in micro-F_1 score on the whole OAG-WhoisWho,

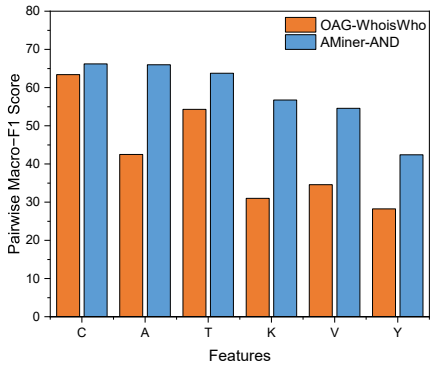

Fig. 7. The effect of different raw document feature for the result of the AND problem. C, A, T, V, K, and Y represent Co-author, Affiliation, Title, Venue, Keywords, and Year respectively.

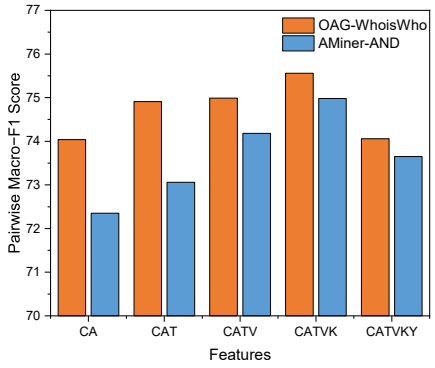

Fig. 8. The effect of selecting different raw document features by adding the features one by one for the result of the AND problem. C, A, T, V, K, and Y represent Co-author, Affiliation, Title, Venue, Keywords, and Year respectively.

and $+2.04\%$ in macro-F_1 score on the whole AMiner-AND. On OAG-WhoisWho, our method outperforms the baselines in terms of macro-F_1 ($+26.69\%$ over AGAND, $+9.8\%$ over Beard, $+7.23\%$ over AMiner, and $+22.76\%$ over GANAND relatively). And our method outperforms the baselines in terms of macro-F_1 ($+12.15\%$ over AGAND, $+11.86\%$ over Beard, $+7.17\%$ over AMiner, and $+2.04\%$ over GANAND relatively) on AMiner-AND. Observed from the experimental results, our framework has a better performance in F_1 score than all state-of-the-art methods since our framework pays more attention to the balance of precision and recall. And we find that the recall of our model is better than other methods from Table IV. We consider that the global structural information and the effective use of raw document feature (most methods do not have) can be the reason to achieve better recall. This proof is when we add the Concatenation module that reconstructs raw document feature and global structural information, the recall greatly boosts $+13.93\%$ on OAG-WhoisWho, as shown in Table VI. From Table IV, GANAND has the better precision than ours. Beard achieves the better recall than ours. This is because GANAND uses adversarial representation learning, which generates high-quality samples. And Beard proposes phonetic-based blocking strategies to increase recall [16].

TABLE VI
CONTRIBUTION OF EACH COMPONENT.

	OAG-WhoisWho			*AMiner-AND*		
Module	***Prec***	***Rec***	***F1***	***Prec***	***Rec***	***F1***
Addition	74.49	62.46	67.95	73.57	67.27	70.28
Concatenation	74.91	76.39	**75.64**	81.39	69.47	**74.96**

To evaluate the performance of each module, we also present our performance at different stages in Table VI. It can be seen that Addition, which reconstructs local structural information and fusion feature, achieves a good performance on AMiner-AND, while the result on OAG-WhoisWho is low in terms of macro-F_1. This is because OAG-WhoisWho meets serious *missing of raw document feature* problem. Compared with other models, we find that this module's result in macro-F_1 is better than most models such as AGAND and Beard, which do not use fusion feature. When we add the Concatenation module, which reconstructs the global structural information extracted from five raw document features (i.e., co-author, affiliation, venue, title, and keywords), into the encoder, we can achieve better results which are 75.64 and 74.96 in macro-F_1 on two datasets respectively. This is why Aminer, which has fusion feature and global structural information, achieves better results on OAG-WhoisWho and others do not. But Aminer does not consider the local structural information and make full use of the raw document feature, which makes Aminer not solve AND better. We take all the above-mentioned information into consideration and achieve better results.

D. *Parameters of Walk*

The node' embedding, which represents the fusion feature and the local structural information, is an important part of our proposed method. We examine how the nodes' neighborhood parameters (number of walks and walk length) affect the performance on OAG-WhoisWho and AMiner-AND. Observed from Fig. 5, when the length of walks reaching around 16 and 32, our proposed method achieves better results on OAG-WhoisWho and AMiner-AND respectively. Similarly, when the number of walks is set to 10, our proposed method can achieve better performance, observed from Fig. 6. This is because the noise is introduced when the length of walks and the number of walks are set to large values. Conversely, when the length of walks and the number of walks are set to small values, nodes will contain less useful information. Both these parameters have a relative impact on the performance of our proposed method, but the performance differences are not large.

E. *Selection of Raw Document Feature*

The edge' embedding, which represents the raw document feature and the global structural information, is another important aspect of our method. Each feature has an effect on the AND problem. We take the *missing of raw document feature*

into consideration to investigate the importance of each raw document feature used in our approach. As the co-author of an author is considered to be a strong discriminative feature by [23], [24], the Fig. 7 shows that co-author and affiliation are the strong discriminative feature. The title, venue, and keywords have some influence on the AND problem, and the year has the least influence. And, we examine how different raw document features affect the performance of our proposed approach by adding the features one by one and starting with co-author and affiliation. Observed from Fig. 8, when several raw document features, which are selected, are co-author, affiliation, title, venue, and keywords, our proposed approach can get better performance on two datasets. The year has a negative effect on our method, and the result is expected, because co-author, affiliation, title, venue, and keywords are highly related to the author, but the year is not. Generally, the authors with the same name may publish articles in the same year, and the authors, who have a unique identity, may publish articles in different years. Therefore, the year is not useful for our method.

V. Related Work

Author name disambiguation has been studied by various methods. Existing methods only exploit parts of the different types of information that consist of raw document feature, fusion feature, local structural, and global structural information. The state-of-the-art methods for author name disambiguation can be divided into two categories: Context-based and Graph-based.

Context-based: Context-based methods consider all raw document features to be the context, which is represented by the feature vectors, and leverages the supervised learning method to learn a pairwise function between publications based on these feature vectors. Then, these methods predict whether two publications written by people with the same name belong to unique people for disambiguation. Han et al. [25] utilize Naive Bayes and SVM to deal with the AND problem. Han et al. [15] employ TF-IDF and NTF to define similarity functions to calculate the similarity of the documents and employs k-way spectral clustering for disambiguation. Yoshida et al. [26] propose a two-stage clustering method to learn better feature representation via the first clustering step. Müller et al. [27] use a deep neural network to solve the AND problem. Kim et al. [13] introduce a hybrid method that extracts structure-aware features and global features, and gradient boosted trees (GBT) and DNN are introduced to experiment with the result of disambiguation respectively. Jhawar et al. [14] conduct experiments with two ensemble-based classification algorithms, namely, random forest and gradient boosted decision trees, on a publicly available corpus of manually disambiguated author names from PubMed.

Graph-based: Graph-based methods consider graphical models by utilizing graph topology and capturing the information of neighbors to deal with the AND problem. Fan et al. [28] construct a graph by collapsing all the co-authors with identical names to one single node, and the distance between two nodes is measured based on the number of valid paths. Tang et al. [29] solve the problem by employing Hidden Markov Random Fields (HMRF) to model node features and edge features in a unified probabilistic framework. Zhang et al. [17] construct three graphs based on document similarity and co-author relationship, and learns graph embedding from three graphs. Zhang et al. [12] design a supervised global stage to fine-tune the word2vec result and an unsupervised local stage based on the first stage. Wang et al. [9] propose a generative adversarial framework, and the discriminative module distinguishes whether two papers are from the same author, the generative module selects possibly homogeneous papers directly from the heterogeneous information network.

Though lots of work has been performed, to the best of our knowledge, there has been no work taking all above-mentioned information into account for the AND problem so far, and they ignore the phenomenon: *missing of raw document feature*. To overcome the drawbacks of existing studies, we propose a unified framework namely MFAND to reduce the effect of *missing of raw document feature* and deal with the AND problem sufficiently.

VI. Conclusion

In this paper, we propose a unified framework namely MFAND, where all types of information consisting of raw document feature, fusion feature, local structural, and global structural information are taken into account, with the goal of addressing the AND problem more effectively. We design a novel generating strategy that extracts local structure information from fusion feature and extracts global structural information from raw document feature precisely. The generating strategy can solve the problem *missing of raw document feature* to a certain extent and disambiguate author name more precisely. The R3JG module, which consists of three JACNN modules that is comprised of Addition and Concatenation, is an encoder used to integrate and reconstruct multiple features information to enhance the generalization ability of MFAND by learning the latent information. Meanwhile, we employ different pruning strategies for different feature graphs to remove noise effectively. Experimental results on two real-world datasets verify the advantages of our method over state-of-the-art methods. In the future, we can apply the proposed framework to recommender systems.

Acknowledgment

This work was supported by the Major Program of the Natural Science Foundation of Jiangsu Higher Education Institutions of China under Grant Nos. 19KJA610002 and 19KJB520050, and the National Natural Science Foundation of China under Grant No.61902270.

References

[1] S. Gupta, N. Duhan, and P. Bansal, "An approach for focused crawler to harvest digital academic documents in online digital libraries," *Int. J. Inf. Retr. Res.*, pp. 23–47, 2019.

[2] Y. Chikazawa, M. Katsurai, and I. Ohmukai, "Multilingual author matching across different academic databases: a case study on kaken, dblp, and pubmed," *Scientometrics*, pp. 2311–2327, 2021.

[3] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: extraction and mining of academic social networks," in *SIGKDD*, 2008, pp. 990–998.

[4] A. A. Ferreira, M. A. Gonçalves, and A. H. F. Laender, *Automatic Disambiguation of Author Names in Bibliographic Repositories*. Morgan & Claypool Publishers, 2020.

[5] J. Kim and J. Owen-Smith, "Model reuse in machine learning for author name disambiguation: An exploration of transfer learning," *IEEE Access*, pp. 188 378–188 389, 2020.

[6] X. Yin, J. Han, and P. S. Yu, "Object distinction: Distinguishing objects with identical names," in *ICDE*, 2007, pp. 1242–1246.

[7] X. Li, P. Morie, and D. Roth, "Identification and tracing of ambiguous names: Discriminative and generative approaches," in *AAAI*, 2004, pp. 419–424.

[8] K. M. Pooja, S. Mondal, and J. Chandra, "A graph combination with edge pruning-based approach for author name disambiguation," *J. Assoc. Inf. Sci. Technol.*, pp. 69–83, 2020.

[9] H. Wang, R. Wang, C. Wen, S. Li, Y. Jia, W. Zhang, and X. Wang, "Author name disambiguation on heterogeneous information network with adversarial representation learning," in *AAAI*, 2020, pp. 238–245.

[10] Y. Ma, Y. Wu, and C. Lu, "A graph-based author name disambiguation method and analysis via information theory," *Entropy*, p. 416, 2020.

[11] L. Zhang and Z. Ban, "Author name disambiguation based on rule and graph model," in *NLPCC*, 2020, pp. 617–628.

[12] Y. Zhang, F. Zhang, P. Yao, and J. Tang, "Name disambiguation in aminer: Clustering, maintenance, and human in the loop," in *KDD*, 2018, pp. 1002–1011.

[13] K. Kim, S. Rohatgi, and C. L. Giles, "Hybrid deep pairwise classification for author name disambiguation," in *CIKM*, 2019, pp. 2369–2372.

[14] K. Jhawar, D. K. Sanyal, S. Chattopadhyay, P. K. Bhowmick, and P. P. Das, "Author name disambiguation in pubmed using ensemble-based classification algorithms," in *JCDL*, 2020, pp. 469–470.

[15] H. Han, H. Zha, and C. L. Giles, "Name disambiguation in author citations using a k-way spectral clustering method," in *JCDL*, 2005, pp. 334–343.

[16] G. Louppe, H. T. Al-Natsheh, M. Susik, and E. J. Maguire, "Ethnicity sensitive author disambiguation using semi-supervised learning," in *KESW*, vol. 649, 2016, pp. 272–287.

[17] B. Zhang and M. A. Hasan, "Name disambiguation in anonymized graphs using network embedding," in *CIKM*, 2017, pp. 1239–1248.

[18] Z. Xiao, Y. Zhang, B. Chen, X. Liu, and J. Tang, "A framework for constructing a huge name disambiguation dataset: algorithms, visualization and human collaboration," *CoRR*, vol. abs/2007.02086, 2020.

[19] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: online learning of social representations," in *KDD*, 2014, pp. 701–710.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015, pp. 1–15.

[22] B. Chen, J. Zhang, J. Tang, L. Cai, Z. Wang, S. Zhao, H. Chen, and C. Li, "Conna: Addressing name disambiguation on the fly," *TKDE*, vol. 10.1109/TKDE.2020.3021256, 2020.

[23] R. G. Cota, A. A. Ferreira, C. Nascimento, M. A. Gonçalves, and A. H. F. Laender, "An unsupervised heuristic-based hierarchical method for name disambiguation in bibliographic citations," *J. Assoc. Inf. Sci. Technol.*, pp. 1853–1870, 2010.

[24] A. F. Santana, M. A. Gonçalves, A. H. F. Laender, and A. A. Ferreira, "On the combination of domain-specific heuristics for author name disambiguation: the nearest cluster method," *Int. J. Digit. Libr.*, pp. 229–246, 2015.

[25] H. Han, C. L. Giles, H. Zha, C. Li, and K. Tsioutsiouliklis, "Two supervised learning approaches for name disambiguation in author citations," in *JCDL*, 2004, pp. 296–305.

[26] M. Yoshida, M. Ikeda, S. Ono, I. Sato, and H. Nakagawa, "Person name disambiguation by bootstrapping," in *SIGIR*, 2010, pp. 10–17.

[27] M. Müller, "Semantic author name disambiguation with word embeddings," in *TPDL*, 2017, pp. 300–311.

[28] X. Fan, J. Wang, X. Pu, L. Zhou, and B. Lv, "On graph-based name disambiguation," *ACM J. Data Inf. Qual.*, pp. 10:1–10:23, 2011.

[29] J. Tang, A. C. M. Fong, B. Wang, and J. Zhang, "A unified probabilistic framework for name disambiguation in digital library," *IEEE Trans. Knowl. Data Eng.*, pp. 975–987, 2012.