Hindawi
Scientific Programming
Volume 2021, Article ID 5577692, 11 pages
https://doi.org/10.1155/2021/5577692

Research Article

Name Disambiguation Based on Graph Convolutional Network

Ya Chen,[1] **Hongliang Yuan**,[2] **Tingting Liu**,[3] **and Nan Ding**[2]

[1]*University of Science and Technology Beijing, Beijing 100083, China*
[2]*Beihang University, Beijing 100191, China*
[3]*China Association for Science and Technology, Beijing 100081, China*

Correspondence should be addressed to Hongliang Yuan; yuanhl@act.buaa.edu.cn

Received 13 January 2021; Revised 14 March 2021; Accepted 31 March 2021; Published 10 May 2021

Academic Editor: Pengwei Wang

Recently, massive online academic resources have provided convenience for scientific study and research. However, the author name ambiguity degrades the user experience in retrieving the literature bases. Extracting the features of papers and calculating the similarity for clustering constitute the mainstream of present name disambiguation approaches, which can be divided into two branches: clustering based on attribute features and clustering based on linkage information. They cannot however get high performance. In order to improve the efficiency of literature retrieval and provide technical support for the accurate construction of literature bases, a name disambiguation method based on Graph Convolutional Network (GCN) is proposed. The disambiguation model based on GCN designed in this paper combines both attribute features and linkage information. We first build paper-to-paper graphs, coauthor graphs, and paper-to-author graphs for each reference item of a name. The nodes in the graphs contain attribute features and the edges contain linkage features. The graphs are then fed to a specialized GCN and output a hybrid representation. Finally, we use the hierarchical clustering algorithm to divide the papers into disjoint clusters. Finally, we cluster the papers using a hierarchical algorithm. The experimental results show that the proposed model achieves average F1 value of 77.10% on three name disambiguation datasets. In order to let the model automatically select the appropriate number of convolution layers and adapt to the structure of different local graphs, we improve upon the prior GCN model by utilizing attention mechanism. Compared with the original GCN model, it increases the average precision and F1 value by 2.05% and 0.63%, respectively. What is more, we build a bilingual dataset, BAT, which contains various forms of academic achievements and will be an alternative in future research of name disambiguation.

1. Introduction

The development of the Internet allows users to conveniently and quickly obtain information from digital learning platforms. Most academic research activities use the Internet as a source to search and download academic resources in various databases. Under such circumstances, how to quickly and accurately filter the required content from the massive data becomes key to improving the user experience. In the literature database system, using person names as keywords is a common search method. Researchers who have been working in a certain field for years often search for relevant studies and reviews in this field. However, the ambiguity of the name itself not only affects the query of information on the Internet, but also affects the inquiry of literature in academic research. When a name is searched in a literature database, it will return a mixed presentation of all documents shared by this author name because most of the original databases use simple string-matching method. This may cause users to waste a lot of time browsing irrelevant content or to increase the input keywords in order to search for the documents they are satisfied with.

In real world, the ambiguity of names is manifested in two aspects. On the one hand, various types of name reference items exist, such as pseudonym and aliases. Besides, the forms of name are not fixed, such as full names and partial abbreviations. On the other hand, the same name reference item can refer to multiple entities in real world. We focus on the latter problem, which is also called author name disambiguation. With the rapid growth of academic

resources, the problem of authors with the same name not only affects the efficiency of academic research and brings inconvenience or even misleading to researchers, but also affects the construction and use of academic resource libraries. Therefore, author name disambiguation attracts more attention of the research community in recent years.

Author name disambiguation is beneficial to the accurate retrieval in the retrieval system. When the user enters the author name to be queried, we can first give the user a series of entity interfaces that share this name reference item. Each interface corresponds to an author entity in the real world. By discriminating the attributes of each entity, the user accesses the relevant literature of the entity he/she wants to query through the interface, which reduces the user's workload and improve the user's experience.

Author name disambiguation is helpful to improve the accuracy and completeness of the author character information. The name disambiguation technology can not only classify the documents with shared names, but also integrate the scattered information of each author entity, so that the characteristic information of the author entity can be continuously improved. This is one of the important steps in the construction of the author's personal home page.

Author name disambiguation is an important part of the construction of literature database. Both universities and scientific research institutions need to count and file the collected papers. One of the important tasks is to file according to individual authors and accurately construct their own literature database in order to evaluate the scientific research achievements and level of the unit. For example, DBLP (https://dblp.uni-trier.de/) is an English literature database system in the computer field. It collects the published scientific research achievements in international journals and conferences with the authors as the core and reflects the frontier direction of foreign academic research.

For the dataset we used, there is more information about the attributes of the paper, but little description information about the author. It is impossible to construct the list of target entities for linking disambiguation. Therefore, we use the disambiguation method based on clustering. The existing methods can be divided into three categories: clustering based on attribute features, clustering based on linkage structure, and hybrid method. Attribute features describe the characteristics of the object itself, such as the title, keywords, and organization. Methods based on attribute features usually focus on measuring the similarity between papers. Structural features describe the relationship between objects, such as whether they participate in writing and whether they have a coauthor relationship, while the linkage-based method pays more attention to the structural information of the graph constructed by the paper and the author. Considering the different emphasis of the two features, we think of an idea that the dataset can be abstracted into a graph, in which the attribute features can be regarded as the characteristics of the nodes in the graph, and the structural features can be quantified as the weights of the edges. In this way, all the information we obtained can be reflected in the form of graph data and can be processed by graph neural networks.

In order to effectively integrate the two levels of feature information, that is, the features of nodes and edges in the graph, we naturally think of the GCN. GCN is an excellent graph data processing model, which can continuously aggregate the node information to form a new node representation by using the edge weight. Therefore, we combine the attribute features with the structure information of the constructed graph, and the embedding representation we have learned has stronger distinguishing power. Finally, we use the clustering algorithm to complete the disambiguation task.

The main contributions of this paper include the following aspects:

We use hybrid features to disambiguate papers based on clustering. More specifically, the attribute features include the title, keywords, venue, name of collaborators, and organizations; the structural information includes the coauthor relationship between authors and the writing relationship between authors and papers. We construct three association graphs for each candidate set, in which nodes contain attribute features and edges contain structural information. The two levels of features are effectively integrated together using GCN, so that the embeddings of papers have strong distinguishing power.

Besides, we use the GCN based on attention mechanism (AGCN). The attention mechanism gives more weight to the areas related to the current task. AGCN can adaptively select different number of convolutional layers according to the structure of different association graphs and properly train and fit the graph data for all name reference items, so as to improve the performance of the disambiguation model.

We also build a dataset, BAT. This dataset collects different forms of achievements including papers, patents, and projects, and the diversification of data forms expands the scope of application of the disambiguation system. In addition, the achievements can be displayed in English or Chinese, and the disambiguation algorithm supports multilanguage processing, which is closer to the actual scenario. BAT dataset provides an experimental platform for exploring and studying various disambiguation models in the future.

2. Related Work

2.1. Author Name Disambiguation. Most disambiguation methods [1, 2] use the information of title, abstract, keywords, and coauthor relationship to extract features for disambiguation. The research on the author name disambiguation is divided into two categories. One is the simple and efficient disambiguation method based on the name [3]. It only uses the author's last name and initials to mine the information contained in the name itself, which is simple and accurate to implement. However, it is more suitable for western countries with the habit of using middle names. The other is mainly solved by advanced methods, including traditional machine learning methods, probability-based methods, and graph-based methods. We now present the details.

Due to the rich variety of machine learning models, we have so many choices to solve the problem of name disambiguation. The supervised methods use labeled sample to train the model and predict whether two papers belong to the same author. Han et al. [4] proposed a Naive Bayes model and a Support Vector Machine model, but the experimental results are quite unsatisfactory when the information is incomplete. The method proposed by Zhang [5] is to construct a Bayesian nonexhaustive classification framework. The data model is a priori. When a new fuzzy entity appears, the Dirichlet module is required to handle it. Besides, the online name disambiguation task is completed by the Gibbs sampler. In practice, manually labeling a large set of data is expensive, which limits the application of supervised methods.

The unsupervised methods extract the feature vectors from the data to calculate the similarity between two records, and disambiguate by the clustering algorithm. In 1998, Bagga and Baldwin [6] proposed a method that uses vector space model to deal with cross-text coreferential disambiguation. However, this method cannot deal with the problem of text field vacancy, so the performance is limited. Mann and Yarowsky [7] construct the feature space by extracting the basic attribute of the author, which achieves a higher accuracy of disambiguation, but the sparsity of the character feature can easily lead to a lower recall rate. Chen and Martin [8] use SoftTFIDF weighted semantic and syntactic features for clustering, which significantly improves the accuracy of disambiguation. Han et al. [9] use *K*-ways spectral clustering to solve the disambiguation problem. Compared with the supervised methods, the unsupervised methods do not need to label data, but the limitation is that the number of clusters is usually unpredictable.

When there is only a small amount of labeled data, semisupervised methods become a better choice. The distance-based semisupervised clustering method uses the similarity distance index to train the model, and the constraint-based semisupervised clustering method combines unsupervised methods with user feedback or guidance constraints to guide clustering [10]. Zhang et al. [11] combined the two previous approaches and constructed a probability model to deal with it. This method considers a variety of constraints; the EM algorithm is used to calculate the distance between two papers and allows users to improve the disambiguation results. The experimental results show that this method is better than the one based on hierarchical clustering.

The problem of author name disambiguation can also be solved by probability model. Tang et al. [12] use Markov random field to combine text attributes and linkage features for disambiguation. Song et al. [13] divided the name disambiguation task into two processes. They first extended the Probabilistic Latent Semantic Analysis (PLSA) and Latent Dirichlet Allocation (LDA), two hierarchical Bayesian text models, and established the topic distribution model. Then, they extracted the features of the distribution and used clustering algorithm to complete the disambiguation task.

In recent years, graph-based disambiguation methods have been applied. They use attribute features extracted from the paper or the external knowledge base to construct the graph [14], and then cluster the nodes in the graph. They also use the linkage information in the graph to mine close relationship between papers and collaborators [15, 16]. Fan et al. [17] propose a graph-based GHOST framework. They first construct the relationship graph between the paper and the author, confirm the effective path between the two points, and eliminate the invalid one. Then, they calculate the similarity matrix and use the affine propagation clustering algorithm to get paper clusters. Finally, they incorporate the users' feedback to improve the results. Network representation learning algorithm Diting [18] consists of three components: network construction, embedding representation, and clustering. Other works use the topological structure of the graph. Franzon et al. [19] proposed a two-step traversal disambiguation method, which uses topological similarity to evaluate candidate nodes. In addition, Peng et al. [20] constructed a heterogeneous network with different types of edges by extracting author name, venue, title, abstract, and other information and proposed to use generating antagonistic network (GAN) to get the embedding representation of the network, so as to cluster the papers.

2.2. Graph Convolution Network. Original Convolution Neural Network (CNN) can only deal with Euclidean structure data with regular spatial structure, but it struggles to deal with the graphs generated from recommendation system, social network, and molecular structure, in which each node has its own characteristic information and structure information. It is necessary to use Graph Convolution Network (GCN) to automatically learn and extract the spatial features of topological graph. The function of GCN is to aggregate the information of nodes into an edge and output a new node representation.

For the graph $G = (N, E)$, input the feature matrix X of all nodes and the adjacency matrix A, and output the embedding representation Z of the node. The computation procedure of GCN is shown in Figure 1 and can be expressed as the following equation:

$$H^{(l)} = \sigma\left(\widehat{A}H^{(l-1)}W^{(l)}\right), \tag{1}$$

where $H^{(l)}$ is the output of nodes in the *lth* layer and $H^{(0)} = X$, σ is the ReLU activation function, $\widehat{A} = D^{(1/2)}AD^{-(1/2)}$ is the symmetrically normalized adjacency matrix of graph G, D is the degree matrix of G, and $W^{(l)}$ is the weight parameters of the *lth* layer. The state of each node depends on other nodes in the graph. The closer the nodes are, the greater the influence is. Ideally, stacking N convolution layers can at most propagate N-hop node information, and the node in the center of the graph can completely aggregate the information of the whole graph.

3. Disambiguation Model Based on Graph Convolution Network

In this section, we first define the problem of name disambiguation. Then, we present the proposed name disambiguation model based on GCN. Specifically, the proposed model consists of two components: global representation

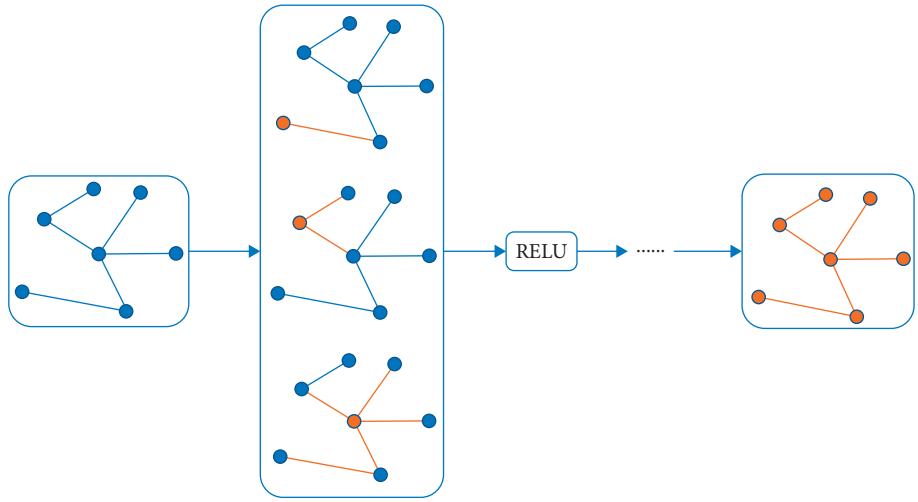

Figure 1: Diagram of the working principle of GCN.

and local representation. The global representation extracts features from the attribute information of the papers and authors [14], and the local representation extracts features from the linkage information in the graph [15].

3.1. Problem Formulation. Given an ambiguous name term, let $P = \{P_i, i = 1, 2, \ldots\}$ denote all the papers related to it and be called the candidate set for this item. Each paper is represented by a varied-length set of attributes including the title, keywords, venue, name of the collaborator, and organization. The real author of the paper is indicated by $A(P_i)$. The goal of the author name disambiguation is to find a mapping function Φ that clusters the papers in P, and each cluster corresponds to only one author entity in the real world. That is to say,

$$\Phi(P) = C, \quad C = \{C_i, i = 1, 2, \ldots\}, \tag{2}$$

$$\text{s.t.} \begin{cases} \forall(P_i, P_j) \in C_k \times C_k, & A(P_i) = A(P_j), \\ \forall(P_i, P_j) \in C_{k1} \times C_{k2}, k1 \neq k2, & A(P_i) \neq A(P_j). \end{cases} \tag{3}$$

3.2. Overall Model for Name Disambiguation. To sum up, this section proposes a disambiguation model based on GCN, which mainly includes two parts: global representation learning and local linkage learning. We first extract the attributes of papers, embed them into a unified vector space, and fine-tune them by triplet loss function. Then, we construct three association graphs in the candidate set, and the edges contain linkage information. In order to integrate the two parts effectively, we build two GCNs and use three kinds of loss functions for iterative training. The final output of Paper-GCN is the hybrid feature representation of papers. Finally, we use the HAC for clustering. The framework of the entire disambiguation model is shown in Figure 2.

3.3. Global Representation Learning. In order to effectively compute the similarity between different papers, it is necessary to convert the papers represented by strings into a vector representation. Here, we use the embedding approach to obtain the vector representation of a paper. The embedding process is divided into two steps. First, all the papers in the database are represented in a unified vector representation. Then, we use the labeled data to train a supervised model which updates the vector representations of the papers.

The core of this method is the modeling of the representation of context and the relationship between context and target words. We use the word embedding method to identify the information of papers; the specific steps are as follows:

Step 1: Extract attribute information of the papers, including the title, keywords, venue, name of collaborators, and organization; carry out word segmentation and cleaning; and then contact these attributes as a sequence, to construct the feature information $P_i = \{x_j, j = 1, 2, \ldots\}$ for each paper P_i. $P_i = \{x_j, j = 1, 2, \ldots\}$. x_j is word appearing in sequence.

Step 2: Take the whole dataset as the corpus and input the feature words of all papers to the Word2Vec model [21]. It embeds the word x_j into a low-dimensional vector representation $x_j\prime$.

Step 3: Calculate the IDF value of each word x_j by

$$\text{IDF}_j = \log\left(\frac{|X|}{|x_j| + 1}\right), \tag{4}$$

where $|x_j|$ indicates the frequency of occurrence of x_j in the corpus and $|X|$ indicates the size of the corpus.

Step 4: Multiply the embedding $x_j\prime$ of each word in a paper by the corresponding IDF value, and sum all the products to obtain the unified embedding P_i' of the paper.

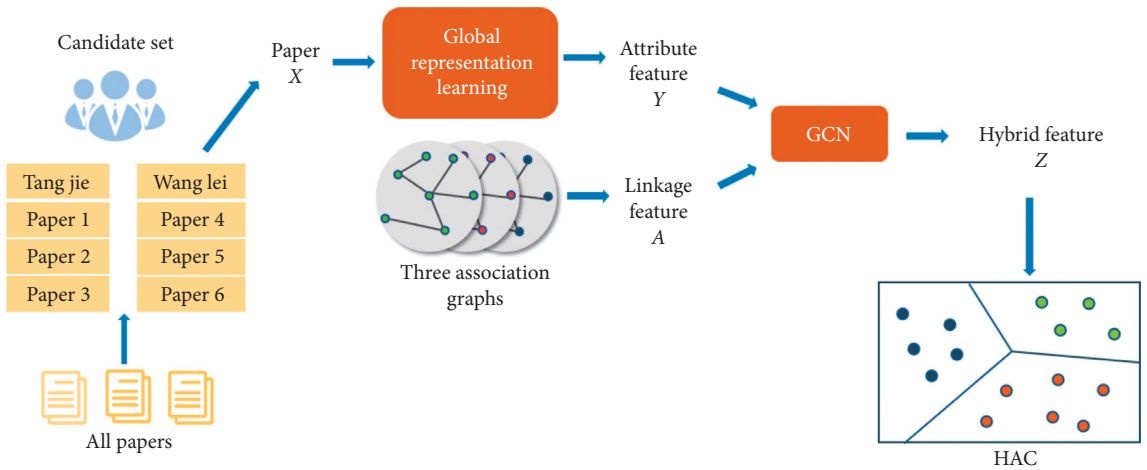

Figure 2: An overview of the author disambiguation framework.

$$P_i' = \sum_{x_j \in P_i} I\ DF_j \cdot x_j'. \tag{5}$$

The distinguishing ability of the unified embedding of papers is limited, so we use the labeled data to adjust the unified representation. Considering that, in the real world, the research field and interest of the same author entity are relatively fixed, in the candidate set corresponding to the same name reference item, we call the two records belonging to the same author entity as positive sample pair. Otherwise, they are called as negative sample pair. Our goal is to train a neural network to find a new mapping relation $f: p' \longrightarrow y$ and update the paper embeddings at the same time. Here, p' is the unified embedding of the paper and y is the global representation. We expect that the positive sample pairs keep closer, while the negative ones keep as far away as possible in the vector space. For any triple (p', p_+', p_-') in the dataset, the constraint condition is $|y, y_+| \ll |y, y_-|$, where $||$ indicates the Euclidean distance. We define the margin-based triplet loss function:

$$L = \sum \max\left(0,\ |y, y_+| - |y, y_-| + m\right), \tag{6}$$

where m is a hyperparameter indicating the margin. The structure of the network is shown in Figure 3.

3.4. Construction of Association Graph. The global representation considers the attributes of papers and authors; the close relations between papers and authors are however not taken into account. In order to make full use of these relations, we consider integrating the relation information to learn a more effective representation. In the candidate set of the same name reference item, we construct three kinds of graphs: paper-to-paper graph, coauthor graph, and paper-to-author graph. The weight $w$of edges in the graph represents the degree of relevance between papers and collaborators.

Paper-to-paper graph: All papers in the candidate set are represented as paper nodes. If the intersection of the attribute features in two papers exceeds a threshold ε after being weighted by the IDF, an edge will be constructed between the two nodes. The reason we take this procedure is that the more attributes two papers have in common, the more likely they are to be written by the same author.

Coauthor graph: This graph shows the cooperative relationship between authors. All the collaborators involved in the candidate set are represented as author nodes. The edge indicates that there is a cooperative relationship between two authors, and the weight of the edge indicates the number of cooperation times.

Paper-to-author graph: This graph shows the writing relationship within the candidate set. We take papers and authors as nodes and construct edge between the paper nodes and the author nodes.

3.5. Local Linkage Learning. In order to effectively integrate the attribute information of papers and authors and the structure information of association graphs, we build two networks, Paper-GCN and Author-GCN. The input of the networks is the features Y of the paper and the author node. The information carried by edges of graphs continuously aggregates the information of nodes, obtaining the new embedding Z of the paper and the author node in the same vector space. By optimizing the loss function, the embedding of nodes is adjusted to ensure that closely connected objects are also adjacent to each other in the embedding space. Finally, the clustering algorithm is utilized to solve the author name disambiguation problem.

We use the linkage information of edges in the graph to integrate the attribute information of nodes. Each candidate set is a relatively independent operation space, and different candidate sets do not affect each other. Hence, we call this module local linkage learning.

In the paper-to-paper graph, the more the common features of two papers are, the more likely they are to belong to the same author entity in the real world, so they should be kept close in the embedding space. That is, for any paper

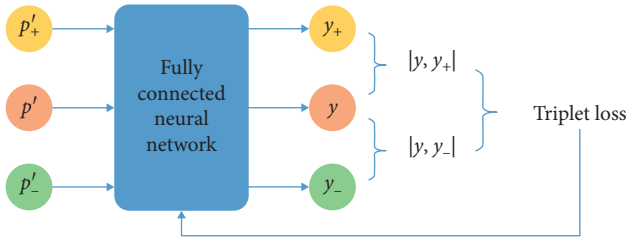

FIGURE 3: Neural network for fine-tuning the unified representation.

node i, the positive sample node j meets the condition $w_{ij} = 1$, and the negative sample node k meets the condition $w_{ik} = 0$, which should ensure $|Z_{pi}, Z_{pj}| \ll |Z_{pi}, Z_{pk}|$. The definition of paper-to-paper loss is

$$\text{Loss}_{pp} = \sum \max\left(0, \left|Z_{pi}, Z_{pj}\right| - \left|Z_{pi}, Z_{pk}\right| + m\right), \quad (7)$$

where $||$ indicates the Euclidean distance and m is a margin between positive and negative node pairs.

Similarly, in the coauthor graph, the more the times two authors cooperate, the more their research fields and interests are similar, or the more they work in similar institutions, so they should remain close in the embedding space. That is, when $w_{ij} > w_{ik}$, we should ensure $|Z_{ai}, Z_{aj}| \ll |Z_{ai}, Z_{ak}|$. The definition of author-to-author-loss is

$$\text{Loss}_{aa} = \sum \max\left(0, \left|Z_{ai}, Z_{aj}\right| - \left|Z_{ai}, Z_{ak}\right| + m\right). \quad (8)$$

In the paper-author graph, if there is a writing relationship between a paper node and an author node, they should remain close in the embedding space compared with those sample pairs without the relationship. For any paper node i, positive sample author node j, and negative sample author node k, when $w_{ij} = 1, w_{ik} = 0$, it should be ensured that $|Z_{pi}, Z_{aj}| \ll |Z_{pi}, Z_{ak}|$. The definition of paper-to-author loss is

$$\text{Loss}_{pa} = \sum \max\left(0, \left|Z_{pi}, Z_{aj}\right| - \left|Z_{pi}, Z_{ak}\right| + m\right). \quad (9)$$

The paper-author graph not only provides information about writing relationships, but also serves as a bridge between the Paper-GCN and the Author-GCN. Loss_{pa} is a function calculated by Z_p and Z_a, which constrains them to the same vector space and facilitates the representation and measurement of the distance between different types of nodes. In the process of optimizing Loss_{pa}, the parameter learning processes of the two GCNs influence each other until the whole disambiguation model converges.

As mentioned above, the structure of three loss functions is the same, and the distance of the positive sample pairs in the embedding space is much smaller than that of the negative sample pairs. When generating the triples, for each anchor point i, we randomly select the positive sample node j according to the weight of the edge connected to i in the graph. The greater the weight, the higher the possibility of being selected. When choosing a negative sample node k, the smaller the weight, the higher the possibility of being selected, and triplets should satisfy $w_{ij} > w_{ik}$.

Paper-GCN and Author-GCN are two parallel networks with the same training methods. The training process is shown in Figure 4, where the blue thin lines represent the input and output of data, and the red thick lines represent the back propagation of the weight parameters.

The inputs are the adjacency matrix A of the graph and the feature matrix Y of all nodes. An adjacency matrix is a symmetric matrix obtained from the undirected weighted graph. Y_{paper} is the global representation of papers. Attribute information about the author in the dataset is rarely described, at most having affiliated organization information. Thus, we use one hot coding to represent the attribute information Y_{author} of author nodes. The outputs are the embeddings Z_{paper} and Z_{author} of graph nodes. For each iteration, we do the following:

(1) We obtain embedding Z_p and Z_a from the Paper-GCN and the Author-GCN, respectively.

(2) We sample triples from the paper-paper graph, minimize Loss_{pp}, and update the weights of the Paper-GCN.

(3) We sample triples from the coauthor graph, minimize Loss_{aa}, and update the weights of the Author-GCN.

(4) We sample triples from the paper-author graph, minimize Loss_{pa}, and update the weights of both networks at the same time.

The GCN plays a key role in the whole mode. It accepts the feature matrix of nodes and the adjacency matrix of the graph as input, corresponding to attribute information and structure information, respectively. As shown in (1), GCN uses the information of edges to aggregate the information of nodes. To be more specific, under the guidance of the linkage information, the attribute information carried by nodes will be transmitted to other nodes from near and far. At the same time, it also receives the information from other nodes and constantly aggregates it with its own information. Each node in the graph is affected by the surrounding nodes and changes its own state. The closer the relationship is, the greater the influence is. The node embedding of the final output of GCN aggregates the features of the nodes and edges and the topological information of the whole graph, so it has stronger distinguishing power.

After training the Paper-GCN, the embedding of the paper nodes is a hybrid feature representation that combines attribute information and linkage information. We use the hierarchical agglomerative clustering (HAC) algorithm to cluster the papers in the candidate set.

3.6. Improved Model Based on Attention Mechanism. Using GCN for name disambiguation, we set the fixed number of convolutional layers to 2. In practice, this setting has some limitations.

First, the optimal number of convolutional layers is usually difficult to determine. If the number of layers is too

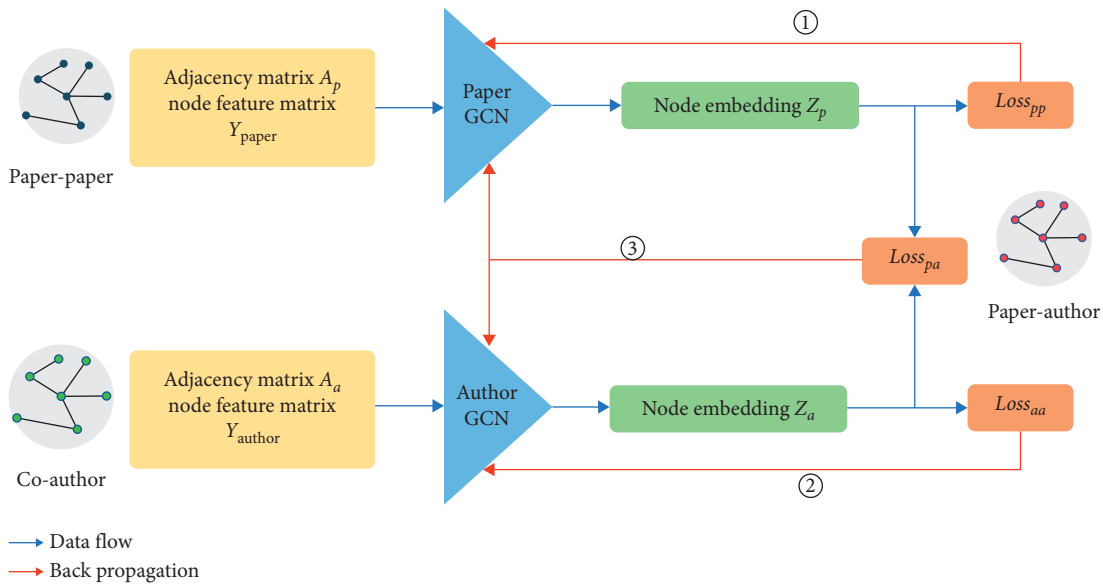

Figure 4: The training of Paper-GCN and Author-GCN.

small, they cannot fully learn the spatial features of the graphs. If there are too many layers, the N-hop neighbor nodes starting from a certain node may form a loop. When aggregating node information, it makes it more difficult to distinguish between distant nodes and nearby nodes. Theoretically, when the number of layers reaches a certain level, the state of the entire network presents a stable fixed point and achieves equilibrium. In practice, the optimal number of layers often needs many tests before it can be adopted.

Second, a fixed number of convolutional layers is not applicable to all graph data. By analyzing the data distribution in the datasets, we find that the nodes and edges in the graph constructed under different name references are in different orders of magnitude. If the topological information in the graph is extracted by the fixed number of convolutional layers, it is likely to result in overfitting for sparse graph data and underfitting for dense graph data. All these will affect the performance of the disambiguation task.

A series of articles show that 3 convolutional layers in GCN can accomplish most of the tasks. We set up 3 layers in GCN and use the embedding of nodes output by all convolutional layers $H = \{h_1, h_2, h_3\}$ to assign an attention coefficient to the output h_i of each layer:

$$a = \text{softmax}(W \cdot H), \tag{10}$$

where W is the parameter that needs to be learned with the training of GCN. The output of the whole network is the weighted summation of the output by each convolutional layer, so that the network can adaptively choose the best number of layers at a certain time.

$$Z = a^T \cdot H. \tag{11}$$

We call the disambiguation model based on attention mechanism as the AGCN model, and the specific structure is shown in Figure 5.

4. Experiment

4.1. Dataset. In order to effectively evaluate the method proposed in this paper, we select two public datasets based on AMiner (https://www.aminer.cn/) system. The AMiner-18 dataset comes from data (https://github.com/neozhangthe1/disambiguation/) published in the disambiguation paper [14] in 2018. The dataset contains a total of 600 name items, including 39781 real authors and 203078 papers. AMiner-12 dataset (https://www.aminer.cn/disambiguation) contains 109 name items, involving 7447 papers from 1546 real author entities. The format of the AMiner-18 data record is shown in Table 1, and the AMiner-12 data record is the same, except that the fields "org," "keywords," and "abstract" are empty.

We construct a bilingual disambiguation dataset, which is provided by the China Association for Science and Technology (https://www.actkg.com/). It brings together a wide range of data sources, covering a variety of research fields, including tens of millions of scientific and technological talents. The disambiguation dataset contains 2905 naming items, with total of 47273 real author entities, and provides 7 data subsets, including talent information set, paper information set, author-paper relation information set, patent information set, author-patent relation information set, project information set, and author-project relation information set. This dataset has the following characteristics:

Each dataset contains detailed attributes. However, due to different data sources, some attributes of some papers are incomplete. The disambiguation algorithm should consider the lack of data attributes.

In addition to common papers, we also collect the relevant achievement information of patents and projects, which enriches the diversity of data and expands the scope of application of the disambiguation system.

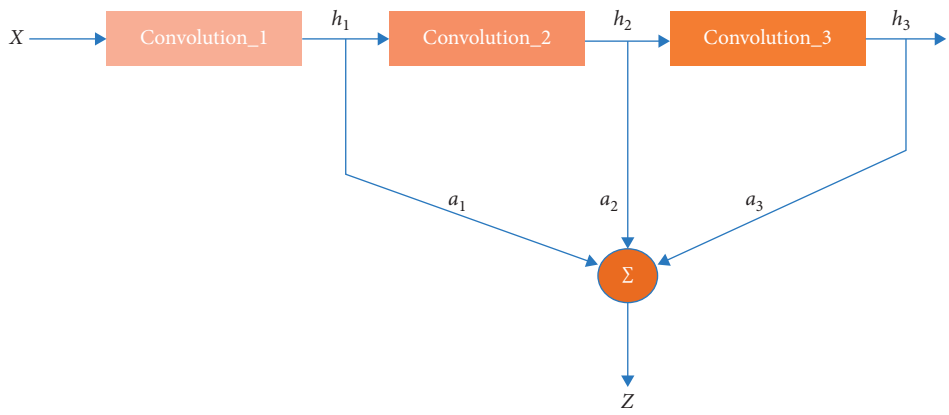

Figure 5: GCN based on attention mechanism (referred to as AGCN for short).

Table 1: The record format of the paper in AMiner-18.

```
"5b5433efe1cd8e4e150e0c7b":{
  "authors": [
    {"name": "Jun Shan," "org": "College of Chemical
    Engineering," "id": "5b5433eee1cd8e4e150c2be6"},
    . . .. . .],
  "title": "Bubble-film network structures of radiation synthesized
  terpolymer hydrogels,"
  "keywords": ["Bubble-Film Network Structure," "Hydrogel,"
  "Swelling Kinetics"],
  "venue": "Polymer Journal," "year": 1997,
  "abstract": "Three series of terpolymer poly(AM-NaA-NTBA)
  hydrogels... "
```

This dataset supports cross-language disambiguation algorithm. The achievements can be written in different languages such as English and Chinese, which is more in line with the actual scenarios.

In order to unify the format with other datasets, this paper only shows the disambiguation results of the paper datasets. The Chinese dataset BAT-CN contains 94 name items, involving 4128 papers of 307 real authors, while the English dataset BAT-EN contains 15 name entries, involving 1288 papers of 35 real authors. The format of BAT paper dataset is the same as AMiner-18.

4.2. Implementation and Parameter Settings. In the global representation learning, the title of the paper is regarded as a necessary attribute. The data missing in this field is regarded as invalid data and discarded. In addition, other attributes are not extracted if the fields are empty. Because we use IDF weighted summation of the extracted attributes to represent the feature vector of the paper, the vacancy of unnecessary fields does not affect the calculation of the feature vector. We sampled 500 name items from AMiner-18 and built a two-layer neural network for model training to fine-tune the unified representation in section global representation learning. The remaining 100 name items involve 35129 papers of 6399 real authors, together with AMiner-12 and BAT as test sets to participate in local link learning.

BAT datasets contain different forms of achievements, and the main difference of disambiguation tasks lies in the extraction of attribute features. For patents, we extract attributes such as patent title, all designers and their organizations, patent types, agencies, and keywords. For projects, we extract project names, all staff and their organizations, keywords, project sources, and project types. Due to the difference in writing formats between Chinese and English, the BAT-CN carries out word segmentation using the Jieba before extracting the attributes.

When building the paper-paper graph, considering the differences in the capacity of three datasets, we set the thresholds ε for IDF to 32 (AMiner-18), 20 (AMiner-12), and 10 (BAT), respectively. We set up two convolutional layers in the GCNs with 128 and 64 neurons, respectively. GCN is trained with Adagrad optimizer, and the learning rate is set at 0.01. In order to get credible results, we train the network multiple times to calculate the average performance metrics to avoid accidental errors.

4.3. Baseline Model. In order to verify the effectiveness of the name disambiguation method based on GCN, we select the common basic disambiguation methods as a comparison.

A simple clustering method (HAC). It only considers the similarity between papers. Extract the title, keywords, venue, and other attribute; use IDF weighted Word2Vec to get the embedding of each paper; and directly carry on the hierarchical clustering for papers.

Rule-based method. This method judges whether two papers belong to the same author entity according to artificially defined rules. For two papers involved in the same name reference item, if the number of common collaborators exceeds the threshold, an edge is constructed between the two article nodes. In the paper-paper graph, the paper nodes in the same connected component belong to the same author entity.

Graph autoencoder [14]. This method constructs a local linkage paper-paper graph for each candidate set. The common attributes of two papers are weighted by IDF to get their similarity. If the similarity exceeds the threshold, construct an edge between two nodes. The unsupervised self-encoder is used to learn the local linkage. We encode the global metric matrix and the

Table 2: The performance of the disambiguation method on individual datasets.

Datasets	AMiner-18			AMiner-12			BAT-CN			BAT-EN		
	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
GCN	65.59	**69.96**	**65.71**	73.94	**78.29**	**74.62**	84.91	**84.07**	**83.43**	79.66	**94.67**	84.62
HAC	70.66	46.28	54.40	67.88	58.38	60.88	67.72	60.95	63.26	86.31	86.79	**86.45**
Rule	**89.19**	37.06	47.31	**99.46**	52.43	64.23	**96.44**	35.31	46.94	**96.93**	49.40	60.54
Graph autoencoder	71.95	57.16	62.03	72.95	56.39	62.11	78.28	75.27	75.80	79.96	84.73	81.89

adjacency matrix of the graph, decode and output the predicted adjacency matrix, and minimize the cross-entropy reconstruction error of the two adjacency matrixes. The intermediate coding vectors are the embedding that integrates the global metric and local linkage information. Finally, hierarchical clustering is used to disambiguate the intermediate coding.

4.4. Results and Discussion. We use pairwise precision, recall, and F1 as the evaluation metrics of the model performance, and the formula is defined as follows:

$$\text{Precision } P = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{12}$$

$$\text{Recall } R = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{13}$$

$$F1 = \frac{2 \times P \times R}{P + R}. \tag{14}$$

We make statistics of sample pairs in the candidate clusters and calculate the value of TP, TP, and FN. The definition is as follows:

True Positive (TP). The sample pairs that actually belong to the same class and are predicted to be of the same class.

False Positive (TP). The sample pairs that actually belong to different classes but are predicted to be of the same class.

False Negative (FN). The sample pairs that actually belong to the same class but are predicted to be of different classes.

Table 2 shows the overall performance of different disambiguation methods on each dataset. As can be seen from the table, the precision of the rule-based disambiguation method approaches 100%, but the recall rate is less than 50%. It indicates that artificially defined rules cannot find the implied information between papers. The performance of simple clustering disambiguation method (HAC) is moderate, which shows that only using the attributes of the paper and comparing the similarity can also meet the basic disambiguation requirements. However, because it does not take into account the constraints in the real world and does not use other linkage information, the average F1 value is about 10% lower than that of the GCN method. The performance of the graph autoencoder is second only to our proposed GCN method. This method also uses both global and local features, but only uses the topological information of the paper-paper graph in the local features and does not consider whether there is participation in writing, whether there is a coauthor relationship, and other link information. The average F1 value is about 6% lower than that of the GCN method.

The disambiguation model based on GCN we proposed has good disambiguation performance. The specific analysis is shown in Table 3. For the first three rows, although the name items are all about 100, the actual capacity is very different. Because of the huge amount of data in AMiner-18, it is not difficult to know that the network structure of the association graphs is complex. On average, there are 60 clusters for each item, and each cluster contains only 5 to 6 papers, so it is difficult to divide it accurately, and the F1 value is the lowest (66.77%). The capacity of AMiner-12 and BAT datasets is smaller, the network structure of their association graphs is relatively simple, and the number of clusters of each name item is 5–10, so their disambiguation performance is better than that of AMiner-18. It is worth mentioning that because the record in AMiner-12 is empty on "org," "keywords," and "abstract" attributes, while the fields of the BAT are complete, its F1 value is about 9% higher than the former. This also shows that the more complete the attributes of the paper, the better the performance of disambiguation.

The disambiguation performance of AGCN has improved GCN to some extent, and the overall accuracy has been improved by 2.05%. Although the recall rate has decreased, the F1 value has increased by 0.63%. It can be found that, on the three datasets of AMiner-18, AMiner-12, and BAT-EN, the accuracy of GCN disambiguation is relatively low compared with the corresponding recall rate. While AGCN makes a trade-off between accuracy and recall rate, thus improving the F1 value. Therefore, the attention mechanism is helpful to improve the performance of the GCN-based author name disambiguation task.

4.5. Component Contribution Analysis

Only use global representation. The global representation learns the attribute information of papers and authors. If we do not use the linkage information in the graph, such as cooperation relationship and writing relationship, and cluster the global representation of papers directly, the result is shown in Table 4.

When using only the title, keywords, venue, name of coauthors, organizations, and other attributes, although the disambiguation performance index can reach more than 50%, compared with the hybrid features, there is still

Table 3: Performance of GCN-based disambiguation method on each dataset.

Datasets	Name reference items	Real author entities	Papers	GCN			AGCN		
				Pre	Rec	F1	Pre	Rec	F1
AMiner-18	100	6399	35129	65.59	**69.96**	65.71	**68.00**	67.74	**65.87**
AMiner-12	109	1546	7447	73.94	**78.29**	74.62	**77.76**	77.98	**76.66**
BAT-CN	94	307	4128	84.91	**84.07**	**83.43**	**85.45**	81.65	82.55
BAT-EN	15	35	1288	79.66	94.67	84.62	**81.10**	**95.38**	**85.85**
Average				76.03	**81.75**	77.10	**78.08**	80.69	**77.73**

a certain gap (the difference of F1 value is 3%–17%). Besides, the larger the capacity and the more complex the structure of the dataset are, the more obvious the difference will be. When only global features are used for disambiguation for the most complex dataset AMiner-18, three evaluation metrics are about 15% lower than our hybrid feature model, which shows the importance of linkage information in graphs, such as cooperation and writing relationships, to disambiguation.

Only use local representation. Local representation learns linkage information in the graphs. If we do not use the attribute information of papers and authors and only use the linkage information for disambiguation, we represent the features of nodes in Paper-GCN by one hot coding. The result is shown in Table 5.

It can be found from the table that only using the linkage information of the coauthor relationship between authors, the similar relationship between papers, and the writing relationship between authors and papers can also complete the disambiguation task. However, compared with our hybrid feature model, there is still a deficiency (the difference of F1 value is 4%–11%). Furthermore, when the capacity of the dataset is smaller and the network structure is simpler, the advantage of the linkage information in the disambiguation task is less obvious. For example, three evaluation metrics of the simplest dataset BAT-EN are about 11% lower than our hybrid feature model. This also shows that the paper's and the author's own attribute information play an important role in the author name disambiguation.

Table 4: Performance of the hybrid features vs. global features.

Datasets	Hybrid features			Global features		
	Pre	Rec	F1	Pre	Rec	F1
AMiner-18	65.59	69.96	65.71	51.51	53.87	51.03
AMiner-12	73.94	78.29	74.62	56.90	61.65	57.74
BAT-CN	84.91	84.07	83.43	66.12	78.66	70.37
BAT-EN	79.66	94.67	84.62	77.82	89.15	81.90

Table 5: Performance of the hybrid features vs. local features.

Datasets	Hybrid features			Local features		
	Prec	Rec	F1	Pre	Rec	F1
AMiner-18	65.59	69.96	65.71	58.99	62.80	61.28
AMiner-12	73.94	78.29	74.62	67.14	70.34	67.31
BAT-CN	84.91	84.07	83.43	74.26	83.77	77.86
BAT-EN	79.66	94.67	84.62	69.11	80.52	73.44

5. Conclusions

The rapid development of modern science and technology has provided great convenience for today's academic research activities, but the author name ambiguity in the literature database has become one of the urgent problems in the field of information retrieval. We carry out some research work on the authors of the same name in the literature database and propose a method to deal with the author name disambiguation by using GCN model, which uses the context information to improve the performance. This model can also benefit other disambiguation tasks, such as word sense disambiguation (WSD) [22], because context information is very indispensable for disambiguation task. We get some improvement, but there are still some related problems to be solved, which mainly involve the practical application of the method:

(1) For the actual disambiguation task, the data is not labeled, so the number of real class clusters is unpredictable. If we use clustering algorithm based on partition, how to estimate the real cluster number is a problem, if we use clustering algorithm based on density, although not need to specify the number of clusters, we also need to estimate the other parameters according to the distribution of data, and these will be our next research content.

(2) Academic papers on the Internet have been growing rapidly; in particular, for some hot fields in recent years, the publication of articles tends to increase exponentially. Faced with such a large dataset, we should not only ensure the accuracy of disambiguation algorithm, but also explore more efficient methods.

(3) We live in a digital world with a strong academic research atmosphere. Excellent articles will constantly emerge and the literature database will be dynamically updated. How to set the update strategy, solve the task of disambiguation with the same name in real time, and keep the consistency of the database are also the problems we should consider and solve.

Data Availability

1. Previously reported AMiner-18 data were used to support this study and are available at https://github.com/neozhangthe1/disambiguation. These prior datasets are cited at relevant places within the text. 2. Previously reported AMiner-12 data were used to support this study and are available at https://www.aminer.cn/disambiguation. These prior datasets are cited at relevant places within the text. 3.

The BAT-CN and BAT-EN data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities (FRF-TP-19-045A1).

References

[1] R. Bunescu and M. Pasca, "Using encyclopedic knowledge for named entity disambiguation," in *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy, April 2006.

[2] X. Han, L. Sun, J. Zhao et al., "Collective entity linking in web text: a graph-based method," in *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 765–774, Beijing China, July 2011.

[3] S. Milojevic, "Accuracy of simple, initials-based methods for author name disambiguation," *Journal of Informetrics*, vol. 7, no. 4, pp. 767–773, 2013.

[4] H. Han, L. Giles, H. Zha et al., "Two supervised learning approaches for name disambiguation in author citations," in *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries*, pp. 296–305, Ontario Canada, June 2004.

[5] B. Zhang, M. Dundar, M. A. Hasan et al., "Bayesian non-exhaustive classification A case study: online name disambiguation using temporal record streams," in *Proceedings of the Conference on Information and Knowledge Management*, pp. 1341–1350, Indianapolis, IN, USA, October 2016.

[6] B. Amit and B. Baldwin, "Entity-based cross-document coreferencing using the vector space model," in *Proceedings of the Meeting of the Association for Computational Linguistics & International Conference on Computational Linguistics*, pp. 79–85, Montreal, Quebec, Canada, August 1998.

[7] G. S. Mann and D. Yarowsky, "Unsupervised personal name disambiguation," in *Proceedings of the North American Chapter of the Association for Computational Linguistics*, pp. 33–40, Seattle. DC, USA, May 2003.

[8] Y. Chen and J. H. Martin, "Towards robust unsupervised personal name disambiguation," *Empirical Methods in Natural Language Processing*, vol. 80, pp. 190–198, 2007.

[9] H. Han, H. Zha, C. L. Giles et al., "Name disambiguation in author citations using A K-way spectral clustering method," in *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries*, pp. 334–343, Denver, CO, USA, June 2005.

[10] W. Zhang, *Research on Disambiguation of Authors with the Same Name in Literature Database*, Shandong University, Jinan, China, 2019.

[11] D. Zhang, J. Tang, J. Li et al., "A constraint-based probabilistic framework for name disambiguation," in *Proceedings of the Conference on Information and Knowledge Management*, pp. 1019–1022, Lisbon, Portugal, November 2007.

[12] J. Tang, A. C. M. Fong, B. Wang, and J. Zhang, "A unified probabilistic framework for name disambiguation in digital library," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 6, pp. 975–987, 2012.

[13] Y. Song, J. Huang, I. G. Councill et al., "Efficient topic-based unsupervised name disambiguation," in *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries*, pp. 342–351, Vancouver, BC, Canada, June 2007.

[14] Y. Zhang, F. Zhang, P. Yao et al., "Name disambiguation in AMiner: clustering, maintenance, and human in the loop," *Knowledge Discovery and Data Mining*, vol. 18, pp. 1002–1011, 2018.

[15] H. Yan, H. Peng, C. Li, J. Li, and L. Wang, "Bibliographic name disambiguation with graph convolutional network," *Web Information Systems Engineering*, vol. 11881, pp. 538–551, 2019.

[16] D. R. Amancio, O. N. Oliveira, and L. da F Costa, "Topological-collaborative approach for disambiguating authors' names in collaborative networks," *Scientometrics*, vol. 102, no. 1, pp. 465–485, 2015.

[17] X. Fan, J. Wang, X. Pu et al., "On graph-based name disambiguation," *Journal of Data and Information Quality*, vol. 2, no. 2, 2011.

[18] L. Peng, S. Shen, J. Xu, Y. Fu, D. Li, and A. L. Jia, "Diting: an author disambiguation method based on network representation learning," *IEEE Access*, vol. 7, pp. 135539–135555, 2019.

[19] V. Franzoni, M. Lepri, Y. Li et al., "Efficient graph-based author disambiguation by topological similarity in DBLP," in *Proceedings of the International Conference on Artificial Intelligence*, pp. 239–243, London, UK, October 2018.

[20] L. Peng, S. Shen, D. Li et al., "Author disambiguation through adversarial network representation learning," in *Proceedings of theInternational Joint Conference on Neural Network*, pp. 1–8, Budapest, Hungary, July 2019.

[21] T. Mikolov, K. Chen, G. Corrado et al., "Efficient estimation of word representations in vector space," 2013, https://arxiv.org/pdf/1301.3781.

[22] E. A. Corrêa, A. A. Lopes, and D. R. Amancio, "Word sense disambiguation: a complex network approach," *Information Sciences*, vol. 442-443, pp. 103–113, 2018.