information

MDPI

Article

Dual-Channel Heterogeneous Graph Network for Author Name Disambiguation

Xin Zheng [1], Pengyu Zhang [1], Yanjie Cui [1], Rong Du [2] and Yong Zhang [1,*]

[1] Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China; xz@bjut.edu.cn (X.Z.); zhangpengyu@emails.bjut.edu.cn (P.Z.); realpizzaonline@gmail.com (Y.C.)
[2] Institute of Microelectronics, Chinese Academy of Sciences, Beijing 100029, China; durong@ime.ac.cn
* Correspondence: zhangyong2010@bjut.edu.cn

Abstract: Name disambiguation has long been a significant issue in many fields, such as literature management and social analysis. In recent years, methods based on graph networks have performed well in name disambiguation, but these works have rarely used heterogeneous graphs to capture relationships between nodes. Heterogeneous graphs can extract more comprehensive relationship information so that more accurate node embedding can be learned. Therefore, a Dual-Channel Heterogeneous Graph Network is proposed to solve the name disambiguation problem. We use the heterogeneous graph network to capture various node information to ensure that our method can learn more accurate data structure information. In addition, we use fastText to extract the semantic information of the data. Then, a clustering method based on DBSCAN is used to classify academic papers by different authors into different clusters. In many experiments based on real datasets, our method achieved high accuracy, which proves its effectiveness.

Keywords: author name disambiguation; big data; heterogeneous graph network

Citation: Zheng, X.; Zhang, P.; Cui, Y.; Du, R.; Zhang, Y. Dual-Channel Heterogeneous Graph Network for Author Name Disambiguation. *Information* **2021**, *12*, 383. https://doi.org/10.3390/info12090383

Academic Editor: Diego Reforgiato Recupero

Received: 9 August 2021
Accepted: 7 September 2021
Published: 18 September 2021

1. Introduction

Every day, new papers are added to the academic paper database. In 2020, the total number of publications in the DBLP database was close to 5.5 million, and the growth rates in the past three years were 10.46%, 10.44%, and 10.37%, respectively. In large-scale digital libraries, such as IEEE (https://www.ieee.org/), DBLP (https://dblp.uni-trier.de/), and ACM (https://dl.acm.org/), researchers commonly search for the author's name, and name ambiguity unavoidably affects the search results. The name ambiguity problem arises when different authors share the same name. Then, when a user searches for this name, the search results will contain many authors or articles that are unrelated to the desired search result. For example, in the DBLP digital library, when users search the database for the name "Li Jie", there will be a large number of authors whose names are written as "Li Jie" in English, but the Chinese names of these authors are different; they are not the same person. In such cases, the accuracy of paper search results will decrease. Some databases require users to apply filters through additional operations to obtain their intended results, while others may require users to click on the search results one by one to confirm which paper they need. The name disambiguation problem affects the accuracy of the search results. Therefore, with the aim of improving the accuracy of search results, the name disambiguation problem has become a popular research topic in recent years.

Traditional name disambiguation usually requires people to manually disambiguate. This method is time-consuming and requires staff to have a lot of work experience. In recent years, many disambiguation works, such as Levin et al. [1] and Bo et al. [2], have used machine learning methods. However, traditional machine learning methods have high data requirements, and the disambiguation results are not ideal due to incomplete records or the inconsistent writing of some datasets [3]. Existing deep learning methods extract and cluster the semantic information of the text in the data to obtain better results.

However, few methods use heterogeneous graphs [4] to extract the structural information between data and then combine it with the semantic information of the text to learn more accurate node representation. Compared with homogeneous graphs, heterogeneous graphs can more comprehensively capture the high-order relationships between nodes, resulting in more accurate node embeddings.

In response to the above problems, we propose the Dual-Channel Heterogeneous Graph Network (DHGN) method. Our method uses fastText (https://fasttext.cc/docs/en/python-module.html, accessed on 7 September 2021) to generate the semantic representation vector of each paper and uses the heterogeneous graph to generate the paper relationship vector. Finally, the semantic similarity matrix and the relationship similarity matrix are merged, and the similarity matrix is clustered by DBSCAN. Our contributions are summarized as follows.

- Heterogeneous graphs incorporate more comprehensive information into node clustering, which allows our method to capture more comprehensive information.
- The semantic information extracted by fastText is merged with the relational information extracted from the heterogeneous graph, and the nodes are clustered using DBSCAN.
- Extensive experiments on real-world datasets prove the effectiveness of our method.

2. Related Work

In order to solve the problem of name ambiguity, we use heterogeneous graphs to extract structural information between nodes. Therefore, we divide the related work in this section into studies on Heterogeneous Graph Networks and Author Name Disambiguation.

2.1. Heterogeneous Graph Network

Heterogeneous graph networks have been attracting considerable attention lately because of their success in various graph classification tasks. Early works, such as Sun and Han [5], used more than one type of node or edge, so the model can capture complex node relationships in the real world. Then, Nandanwar et al. [6] used Vertex-Reinforced Random Walk over a heterogeneous graph. This ensures diversity by discouraging the recommendation of multiple influential nodes. In order to fully mine latent structure features of users and items, Shi et al. [7] designed a meta-path-based random walk strategy to generate meaningful node sequences for network embedding. Outside the field of graph classification or graph clustering, heterogeneous graph networks have also shown strong results in natural-language processing. Anchiêta et al. [8] explored a graph structure representation and modeled the paraphrase identification task over a heterogeneous network. As a result, the model can identify whether two sentences convey the same meaning (even with different words).

The above works used the heterogeneous graph network to solve the recommendation problem, node classification problem, and other problems. However, few studies have used both the relationship information between nodes and the nodes' information to perform node clustering.

2.2. Author Name Disambiguation

The purpose of name disambiguation in an academic paper is to distinguish between different authors with the same name in a large database. Han et al. [9,10] used the typical method, which is to calculate the similarity between different papers and then cluster the papers. Kang et al. [11] used four typical features of an academic paper—the title, author's name, name of the conference, and year of publication—to determine the authorship of the paper. Shin et al. [12] solved the name ambiguity problem by using social networks constructed based on the relations among authors. Based on previous works, Schulz et al. [13] established a similarity metric, which is based on common co-authors, self-citations, shared references, and citations, that first connects individual papers and then merges similar clusters. Zhang et al. [14] presented a complementary study from another point of view: they used Wikipedia to improve the accuracy of the disambiguation

result. The above methods mainly use the features of the publication address of the academic paper, the author's email address, the author's institution, the co-keywords, and the author's homepage. However, not every article has the above information. For example, there are very few authors with homepage information; furthermore, the author's email and the citation information of each paper are more challenging to obtain in some datasets.

3. Problem Formulation

This section introduces some of the concepts and symbols used in this paper.

Concept 1. Academic Paper. Assuming that a given academic paper is denoted as P, which includes the title, author's name and organization, abstract, keywords, publication year, journal/conference name, and other information, it can be expressed as:

$$P = \{ \text{Title, Authors, Author - Orgs, Abstract, Keywords, Pubyear, Venue} \}. \quad (1)$$

Concept 2. Author Name Ambiguity. Given an author named A, the collection of academic papers under that name is denoted as P^A, and there are a total of n academic papers in the collection, that is, $P^A = \{P_1^A, P_2^A, P_3^A, \cdots, P_n^A\}$. There is only one reason why different papers are classified into P^A: that is, different papers have the same author name. If there are two papers in P^A that do not belong to the same author, and algorithms or observation show that although the two papers have the same author name, the name belongs to different organizations, then author A has the name ambiguity problem. For example, in the academic papers $P^{LiJie} = \left\{P_1^{LiJie}, P_2^{LiJie}\right\}$, P_1^{LiJie}'s Title is "qzbSeKfx", and its Author-Org is "Sichuan Academy of Food and Fermentation Industries", while P_2^{LiJie}'s Title is "6Rdqk4Ea", and its Author-Org is "Fermi National Accelerator Laboratory". Since the AMiner dataset (https://www.aminer.cn/whoiswho) encrypts the title of the paper, the title here is represented by garbled characters.

We can see that although the two papers belong to an author named "Li Jie", the organizations of the two papers are different: one is "Sichuan Academy of Food and Fermentation Industries", and the other is "Fermi National Accelerator Laboratory". Hence, the name ambiguity problem exists.

Concept 3: Author Name Disambiguation. Given a collection of academic papers published under a given name, a suitable author must be matched to each paper in the collection. That is, given the academic paper collection P^A under author A, suppose that the papers can be divided into m authors, that is, $A = \{A_1, A_2, A_3, \cdots, A_m\}$. Thus, author name disambiguation is the process of dividing P^A into m subsets so that each subset $P_i^{A_m}$ corresponds to a certain author A_i $(1 \leq i \leq m)$.

Concept 4: Heterogeneous Graph Network. A heterogeneous graph network has a different network structure from a homogeneous graph network. It refers to a network in which the number of node types in the network is greater than 1, or the number of edge types is greater than 1. In order to capture and utilize the heterogeneity of nodes and links, heterogeneous graph networks have been proposed and widely used in many network analysis scenarios, such as meta-path-based similarity search [15], node classification and clustering, and knowledge base complementation and recommendation.

The heterogeneous graph network is defined as $G = (V, E, T)$, where V represents the node set in the network, and E represents the edge set in the network. The function of each node is $\varphi(v) : V \to T_V$, and the function of each edge is $\varphi(e) : E \to T_E$, where T_V and T_E represent the types of nodes and edges, respectively, and satisfy $|T_V + T_E| > 2$.

According to the features of academic papers, the relationship between papers can be represented using a heterogeneous graph network. The heterogeneous graph network can contain node types such as "academic paper", "author", "journals and conferences", and "author organization". Therefore, these relationships can be expressed as a heterogeneous graph network, as shown in Figure 1.

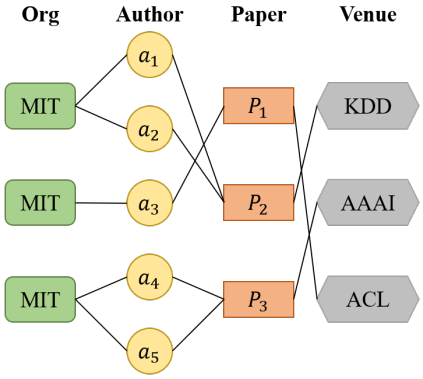

Figure 1. Heterogeneous graph network based on academic papers.

4. DHGN: The Proposed Method for Author Name Disambiguation

To facilitate mathematical analysis, we abstract real-world individuals as nodes and abstract the relationships between nodes as edges, and nodes and edges together form the graph. However, due to the diverse relationships between nodes in the real world, it is difficult for the method to learn accurate node representations, so it is difficult to obtain accurate node clustering results.

In response to the above problems, we propose the Dual-Channel Heterogeneous Graph Network for author name disambiguation. The method learns the semantic information of the papers through the textual information, including the title, the journal/conference that published the paper, author institution, the abstract of the paper, and keywords. Then, fastText is used to generate the semantic representation vector of each paper, and then the semantic similarity matrix of the paper is obtained. A heterogeneous graph network based on the paper is constructed, and the meta-path-based random walk algorithm is used to obtain the relationship features of the paper, the relationship vector of the paper, and the relationship similarity matrix. Finally, the semantic similarity matrix and the relational similarity matrix are merged, and the similarity matrix is clustered by DBSCAN. In this way, papers by different authors can be allocated to different clusters, thus achieving author name disambiguation. Because our method does not have strict requirements for the dataset, it is suitable for academic paper digital libraries in different organizations or different public datasets, such as the DBLP digital library dataset or Aminer digital library dataset. The framework is shown in Figure 2.

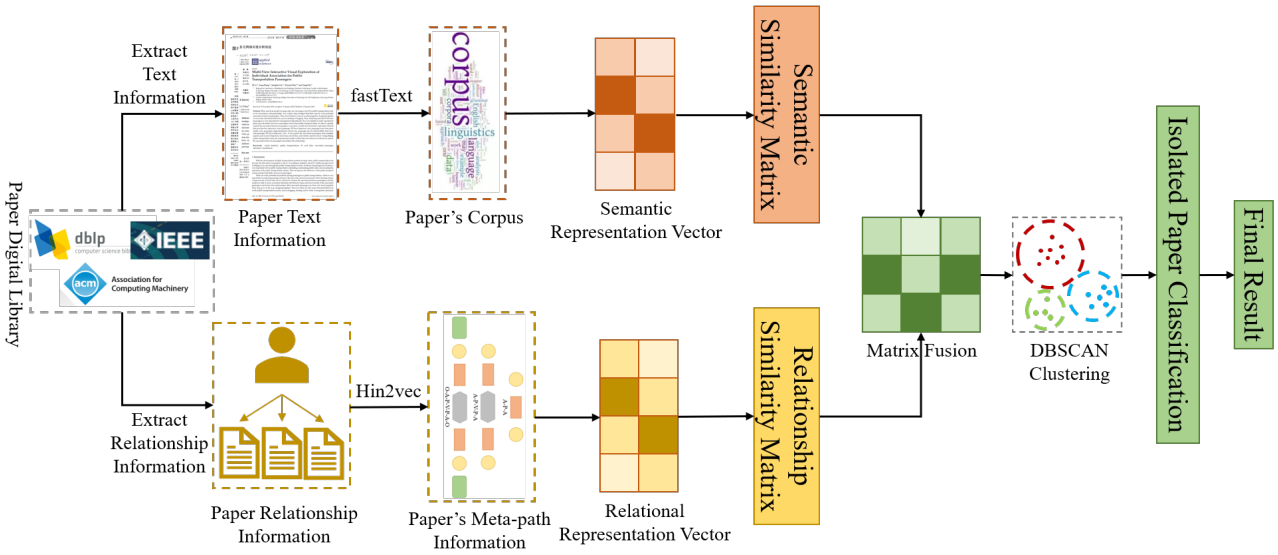

Figure 2. Dual-Channel Heterogeneous Graph Network Framework.

4.1. Use FastText to Construct Semantic Representation Vector

The purpose of semantic representation learning is to obtain the vector representation of the semantic information based on Cai et al. [16] and Shi et al. [17]. For this purpose, we used fastText [18], which was trained on a large-scale academic paper database through unsupervised learning to obtain a word vector model. In order to obtain the semantic representation vector of the entire article, it is necessary to obtain the vector representation of the entire article by performing a weighted summation on the required word vectors selected in the word vector model.

Construction of the training corpus. Before training the word vector model through fastText, a corpus suitable for training needs to be constructed. In order to improve the effectiveness of the fastText training word vector model, this training corpus should include a large amount of data. Our research is based on the AMiner open-source dataset, which has a small amount of data. In order to improve the result of the fastText training word vector model, the amount of data in this training corpus needs to be further expanded. The data that we use include two parts. One part comes from the collected academic paper dataset, about 200,000 academic papers; the other part comes from the DBLP dataset, about 1.4 million academic papers.

First, the text information of each article in the above-mentioned dataset is extracted into a text corpus. The title, journal/conference, author's institution, keywords, abstract, and publication year of each paper are included in its textual information. The text information is saved in a text file as a training corpus. Then, the fastText model is used to train the word vector model.

Training parameter settings. After comparing the effects of the word vector model trained under multiple sets of different parameter settings in the disambiguation experiment, a set of relatively optimal parameter setting schemes was obtained. The specific training parameters are shown in Table 1.

Table 1. Parameter setting of fastText training semantic word vector model.

Parameter	Parameter Description	Parameter Settings
sentence	training text per line	Academic paper information
size	dimension of word vector	100
windows	context word range	5
sg	CBOW or Skip-gram	0
hs	optimal strategy	1
min_n	minimum number of characters	3
max_n	maximum number of characters	6

As shown in the table, the dimension of the semantic vector generated by fastText is set to 100. The training model selected in the training process is the CBOW [19] model. The $Softmax$ algorithm is used for optimization, the context word window size is set to 5, and the minimum and maximum numbers of training characters are set to 3 and 6, respectively. After the training is completed, a word vector model file is obtained, mainly composed of "word-corresponding word vector and character-corresponding character word vector" for semantic vector generation.

The fastText algorithm is used to train text word vector models not only because of its ability to train word vectors at word granularity but, more importantly, because of its ability to train word vectors at character granularity [20]. For example, the word "matter", assuming that the 3-gram feature is used, can be represented as five 3-gram features.

Using n-gram has the following advantages: (1) It generates better word vectors for rare words. According to the above character-level n-gram, even if this word appears very few times, the characters that make up the word and other words have shared parts, so this can optimize the generated word vector. (2) Even if the word does not appear in the training corpus, the word vector of the word can still be constructed from the character-level

n-gram [21]. (3) *n*-gram can allow the method to learn word order information. If *n*-gram is not considered, the information contained in the word order cannot be considered, which can also be understood as context information. Therefore, *n*-gram is used to associate several adjacent words, which allows the method to maintain word order information during training.

Semantic matrix of academic papers. First, text information, such as the title, published journal/conference, author's institution, keywords, abstract, and publication year, is separated by spaces to synthesize the paragraph representing this paper. Then, the text information is processed to lowercase letters; various non-letter symbols, extra spaces, stop words, and words with a length less than 2 are removed, and spaces are used for word segmentation.

Through the above-trained fastText word vector model, a corresponding word vector is generated for each word in the processed text, and each word vector is weighted to obtain a semantic representation vector representing each paper. The reason for assigning different weights to each word vector is that the importance of different words varies. It is generally observed that certain words are only used in specific fields and appear less frequently. The importance of some common words is generally lower than that of some specific words. We counted the words in the AMiner dataset. After removing the stop words, the dataset has nearly 370,000 words, which constitute the training corpus. After further statistical analysis, we found that most words appear less than 10 times in the dataset. The results are shown in Table 2.

Table 2. Statistics of word frequency in training corpus.

Frequency	Total Number of Words
less than or equal to 10 times	316,802
10–100 times	49,972
more than 100 times	19,738

Finally, the cosine similarity between the two papers is calculated by the obtained semantic vector of academic papers. This results in the semantic similarity matrix of papers.

4.2. Use Heterogeneous Graph to Construct Relational Representation Vector

The academic paper contains semantic information and information about the co-authors and their institutions, which means that a social network relationship also exists between academic papers. Exploring this kind of network relationship and combining it with semantic information is the key to name disambiguation. We used the related information to obtain a relationship vector representation for subsequent research on name disambiguation.

The process of obtaining the relationship representation vector of each paper is mainly divided into the following two steps: (1) Construct a heterogeneous graph network of the paper using the author's information, author's institution information, and the journal or conference where the paper is published. (2) After constructing the heterogeneous graph network, use the meta-path random walk method to learn the representation vector of each paper [22]. There has been much research and progress related to heterogeneous graph networks. Our work was influenced by related research on methods such as metapath2vec [23] and Hin2Vec [24], and we adopted a Hin2Vec-based heterogeneous graph network [25] to obtain the relationship vector representation of the paper. In contrast to the way that metapath2vec walks according to the given meta-path, the HIN2Vec model is completely based on a random walk, and it can walk as long as the nodes are connected.

Constructing the heterogeneous graph network. First, the author's name to be disambiguated is extracted; then, the relationship between all published academic papers corresponding to this name is extracted, and a heterogeneous graph network is constructed. Python is used to construct the adjacency matrix that represents the node relationship. If

there is a connection between nodes i and j, then in the adjacency matrix, the value in row i and column j is 1. If there is no connection between nodes i and j, then the value in row i and column j is 0. The connection between the node and itself is not considered; that is, the connection between the node and itself is 0. We constructed a heterogeneous graph network containing one type of node: academic paper (T_i). That is, each academic paper represents a node in the heterogeneous graph network. The network also contains three types of edges: co-author, author's organization (Co-Org), and journal that published the paper (Co-Venue). The heterogeneous graph network that we constructed is shown in Figure 3.

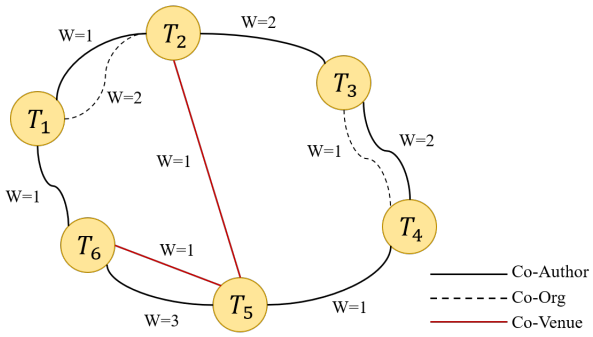

Figure 3. Local structure of heterogeneous graph network.

If two papers contain co-authors other than the author to be disambiguated, we construct a Co-Author edge between the two papers in the heterogeneous graph network. The weight W of the edge represents the number of co-authors other than the authors to be disambiguated. For example, in Figure 3, two papers T_1 and T_2 have a co-author; then, an edge named Co-Author is constructed between them, and the weight W of the edge is 1.

Similarly, in the heterogeneous graph network, Co-Org represents the similarity relationship between co-authors' institutions. The weight W of the edge represents the number of words in common in the names of the two institutions. For example, in Figure 3, the institutional information of two papers T_3 and T_4 has a common word that is not a stop word, so an edge named Co-Org is constructed between the two papers, and its W is 1. Co-Venue type edges are constructed in the same way as Co-Org edges, so we do not repeat the description.

Random walk of meta-paths in the heterogeneous graph network. The meta-path can be defined in the following form. $R = R_1 R_2 \cdots R_{l-1}$ represents the pathset from vertex V_i to vertex V_j.

$$V_1 \xrightarrow{R_1} V_2 \xrightarrow{R_2} \cdots V_i \xrightarrow{R_i} V_{i+1} \xrightarrow{R_{i+1}} \cdots V_{l-1} \xrightarrow{R_{l-1}} V_l \tag{2}$$

Taking the heterogeneous graph network in Figure 4 as an example, the vertex types in the network are Org, Author, Paper, and Venue. There are several types of meta-paths: (1) "Author-Paper-Author", (2) "Author-Paper-Venue-Paper-Author", and (3) "Org-Author-Paper-Venue-Paper-Author-Org".

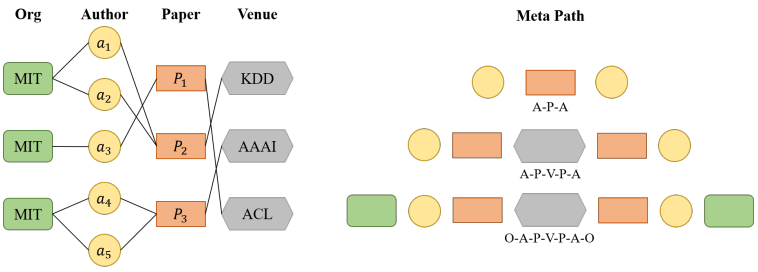

Figure 4. Local structure of heterogeneous graph network.

After constructing the heterogeneous information network, we generate a pathset composed of the paper's ID based on the random walk of meta-paths. Then, the paper's ID is used as the input of the Hin2Vec model. Finally, the corresponding relationship representation vector for each ID is obtained. The relationship vector of papers indicates that the relationship between different papers has been learned, and the similarity between the two papers can be obtained by calculating the cosine similarity.

In a random walk, a node is randomly selected in the network as the initial node, and the process then walks along the edge connected to the initial node to the next node in the network. The number of nodes, also called the path length, needs to be set in advance of the walk. After the random walk, the path set is saved for subsequent training. The random walk based on the meta-path refers to a random walk on the edge of the network. It is not completely random but can be guided by the specified meta-path [26]. In randomly selecting the next node, we consider the weight of the edge in the network. The greater the weight, the closer the relationship between the two nodes, and the greater the probability that the node will walk along this edge. We stipulate that the probability and the weight are proportional. For example, in Figure 5, T_1 is selected as the initial node, the next relationship of the random walk is "Co-Author", and the three nodes that have this relationship with T_1 are T_2, T_3, and T_8. According to the edge weight, the probability of walking from T_1 to T_2 is $2/3$, the probability of walking to T_3 is $1/4$, and the probability of walking to T_8 is $1/4$.

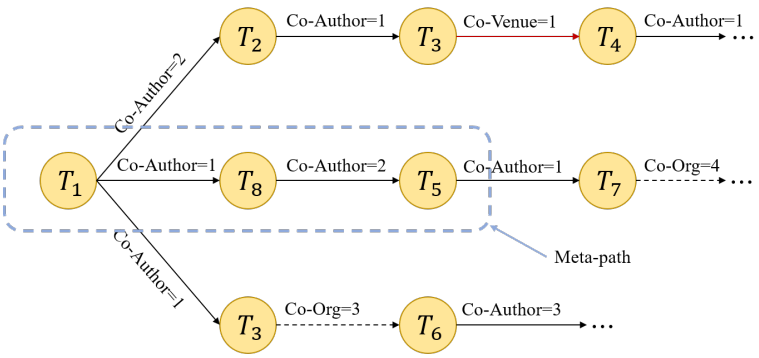

Figure 5. Local structure of heterogeneous graph network.

The specific random walk process selects the next node according to the edge specified by the current meta-path; it randomly selects a node connected to the current node through this edge as the next node. In each path, the meta-path is repeatedly sampled several times. That is, the last node of the previous meta-path is used as the first node of the next meta-path. Iterations are continuously carried out until a certain number of rounds is reached, after which another node is selected as the initial node to walk. Finally, several paths are generated, where the node of each path is the ID of the paper, and each path is stored for the generation of the relation vector.

There is no guarantee that all three types of edges and two types of nodes exist in a heterogeneous graph network. For example, in a particular paper, authors other than the author to be disambiguated may not appear, so then the "Co-Author" relationship of this paper is missing. When this happens, other random walk strategies need to be adopted. For example, a random walk can be performed based on the "Co-Org" relationship or the "Co-Venue" relationship. When a node cannot construct any type of edge with other nodes, we classify it as an isolated node, and we save it separately for subsequent processing.

Academic paper relational representation learning. When obtaining the relation vector representation of each paper, nodes T_i and T_j and their relation R are subjected to

one-hot encoding as the input of Hin2Vec, and the vector representation of the node is learned by maximizing the relation R. The objective function is defined as O,

$$O \propto \log O = \sum_{(T_i, T_j, R) \in D} \log O_{T_i, T_j, R}(T_i, T_j, R), \tag{3}$$

where D represents the training dataset, and T_i and T_j represent two nodes in the heterogeneous graph network. R represents the relationship between the two nodes. During training, the training dataset is given in the following form:

$$O_{T_i, T_j, R}(T_i, T_j, R) = \begin{cases} P(R \mid T_i, T_j), L(R \mid T_i, T_j) = 1, \\ 1 - P(R \mid T_i, T_j), L(R \mid T_i, T_j) = 0, \end{cases} \tag{4}$$

where $P(R \mid T_i, T_j)$ represents the probability that there is a relationship R between nodes T_i and T_j in the heterogeneous graph network, and $L(T_i, T_j, R)$ is a binary value. When it is equal to 1, it means that the objective function is maximized. It is simplified to:

$$\begin{aligned} \log O_{T_i T_j, R}(T_i, T_j, R) = L(T_i, T_j, R) \log P(R \mid T_i, T_j) + \\ \left[1 - (T_i, T_j, R)\right] \log\left[1 - P(R \mid T_i, T_j)\right]. \end{aligned} \tag{5}$$

After learning the vector representation of each node in the heterogeneous graph network, the cosine similarity between every two papers is calculated to obtain the relationship similarity matrix. Hin2Vec learns the node vector representation using the parameter settings shown in Table 3.

Table 3. Parameter settings of relation vector of training articles.

Parameter	Parameter Description	Parameter Settings
walk	times of random Walks	5
walk_length	random walk length	20
embed_size	dimension of word vector	100
n_epoch	number of training sessions	5
batch_size	number of training sample	20

For each node, the set number of walks is five, and the maximum length of each walk is 20. In the representation learning stage, the model is trained five times, each training sample is 20 groups, the vector representation of each node is learned, and the embedding vector is 100 dimensions.

4.3. Use DBSCAN for Node Clustering

Clustering is performed through the semantic similarity matrix and the relationship similarity matrix obtained in the previous two sections. This section describes how the DBSCAN clustering algorithm is used to cluster academic papers.

DBSCAN clustering. We add the relationship similarity matrix and the semantic similarity matrix to find the average value and obtain the final academic paper similarity matrix.

The unsupervised clustering algorithm DBSCAN [27] is used for disambiguation, and papers by the same author are clustered into one class. The DBSCAN clustering algorithm does not require the number of clusters to be specified in advance and is not sensitive to outliers in the data. Because we cannot accurately obtain the actual value of the number of real authors under the one name, we cannot determine the number of clusters in the final cluster, but DBSCAN avoids this problem. The parameter settings of DBSCAN are shown in Table 4.

Table 4. DBSCAN clustering parameter settings.

Parameter	Parameter Description	Parameter Settings
eps	cluster neighborhood threshold	0.2
min_simples	minimum number of samples in cluster	4
metric	distance measurement	precomputes

Matching of isolated papers. After the above DBSCAN clustering, some clusters can be obtained, and each cluster contains academic papers belonging to the same author. In addition, for the isolated papers that cannot establish a relationship with any node, the similarity feature matching method is used to redistribute them to clusters that have already been generated. The remaining isolated papers that cannot be matched are treated as new clusters.

An isolated paper arises because this part of the paper has too little feature information, and its similarity to other papers is relatively low, or the paper itself belongs to an author who has published few papers. Using a feature-based matching approach works better for this type of paper. Therefore, the matching method for this scenario is divided into the following two steps.

Step 1: Each paper in the isolated paper set is compared with other papers in the dataset through feature matching, and the paper with the highest similarity is identified. If the feature similarity between the two documents is not less than the threshold α, then the former is allocated to the cluster that contains the latter. Otherwise, the former is individually classified as a new cluster.

Step 2: After completing feature matching for each isolated paper, the papers in new clusters are compared. If the similarity between the two papers is not less than the threshold β, then the two papers belong to the same cluster. Otherwise, the two papers do not belong to the same cluster. This step is completed until matching is complete.

Specifically, when matching the feature similarity of an isolated paper set, first, the stop words are removed from the title, keywords, and abstract, and then a new piece of text is synthesized. The Jaccard Index of the text of papers T_i and T_j is calculated. The formula is defined as:

$$J(X, Y) = 1 - \frac{|X \cap Y|}{|X \cup Y|}, \tag{6}$$

where X and Y represent the bag of words in papers T_i and T_j, and the result range is (0, 1). After the above processing, the papers in the isolated paper set can also be assigned to their respective clusters, thereby obtaining the clustering disambiguation results of all academic papers.

5. Experimental Results and Analysis

5.1. Datasets

Datasets. We used the AMiner (https://www.aminer.cn/whoiswho) author disambiguation dataset. It contains 221 author names, 22,839 real authors, and 205,498 academic papers. It also has many academic papers compared with other manually annotated author disambiguation datasets that are publicly available. Additionally, each name to be disambiguated contains more authors and articles than other datasets, making the AMiner dataset more challenging. To promote reproducibility, our code and datasets will be published on the following website (https://github.com/pizzaonline/Dual-channel-Heterogeneous-Graph-Network-for-Author-Name-Disambiguation, accessed on 7 September 2021). The AMiner dataset contains two types of files. The first type is the relationship-type file with the mapping relationship of "name to be disambiguated"-"author ID"-"paper ID". The data format is shown in Table 5.

Table 5. Data format of academic literature relationship.

Field	Data Type	Meaning	Example
Name	String	Name to be disambiguated	Li_guo
Name ID	String	Author ID	sCKCrny5
ID	String	Paper ID	UG32p2zs

The other is a metadata file that contains the basic information of each paper. The metadata file is saved in the form of a dictionary, the key is the ID of the paper, and the value is the basic information of the paper, which mainly includes the title, the author's name, the author's institution, the abstract, the publication year, and the journal or conference that published the paper. The data format is shown in Table 6.

Table 6. Format of academic paper metadata.

Field	Data Type	Meaning	Example
ID	String	Paper ID	P9a1gcvg
Title	String	Paper Title	Rapid determination of central nervous drugs in plasma by solid-phase extraction and GC-FID and GC-MS
Venue	String	Journal/conference	Chinese Pharmaceutical Journal
Author.name	String	Author's name	Li Guo
Author.org	String	Author's organization	Institute of Pharmacology and Toxicology
Keywords	String	Paper's keywords	Cholecystokinin-4; Enzymatic synthesis; Peptide;
Abstract	String	Paper's abstract	The enzymatic synthesis of CCK-8 tripeptide derivative Phac-Met-Asp(OMe)-Phe-NH
Year	Int	Paper's publication year	2019

Data preprocessing. For the convenience of subsequent experiments, the data need to be preprocessed. The specific processing flow is as follows:

(1) The author's name. The name of the same author is usually written in different ways in different papers. For example, the author's name is "Huang JianCheng", which can be written in many ways, such as "Huang_JianCheng", "Huang_Jian_Cheng", and "JianCheng Huang". It is necessary to unify the name into one written form during data preprocessing.
(2) The author's institution, name of the journal, name of the conference, abstract, and title. This information usually contains a large number of special symbols. Special symbols need to be removed during preprocessing, and the institution names need to be converted to the lowercase English form.
(3) Keywords. Observations indicate that keywords generally do not contain special symbols. Keywords need to be converted to lowercase.
(4) Escape characters. Some papers contain escape characters. For example, "\u03b2" means "β" in the paper. We found that removing such escape characters has little impact on our research results, so we deleted them. We also found that if special symbols are removed in step (2), escape characters cannot be effectively removed. Therefore, it is necessary to remove escape characters before performing step (2).

Data preprocessing ensures the consistency and reliability of the data, which can provide a solid data foundation for the subsequent analysis.

5.2. Baselines

Evaluation Indicators. For the name disambiguation problem, we use Precision, Recall, and F1 as the evaluation indicators of the experimental results. The three evaluation indicators are calculated as follows:

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{7}$$

$$\text{Recall} = \frac{TP}{TP + FN}, \tag{8}$$

$$F1 = \frac{2 \times \text{Precision}}{\text{Precision} + \text{Recall}}. \tag{9}$$

TP indicates the number of papers that belong to the same author and are correctly clustered into one category. TN indicates the number of papers that do not belong to the same author and are correctly clustered into different categories. FN denotes the number of papers that belong to the same author and are incorrectly clustered under other authors' names. FP indicates the number of papers that do not belong to the same author and are incorrectly clustered under the same author's name.

Baselines. In order to verify the effectiveness of our method, it was experimentally compared with groups of other methods.

(1) GSDPMM (https://github.com/junyachen/GSDPMM, accessed on 7 September 2021) [28]. A collapsed Gibbs sampling algorithm was used for the Dirichlet Process Multinomial Mixture model for text clustering, which does not require the number of clusters to be specified in advance and can cope with the high-dimensional problem of text clustering.

(2) LightGBM (https://github.com/microsoft/LightGBM, accessed on 7 September 2021) [29]. Two novel techniques were used in this study: Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB). GOSS excludes a significant proportion of data instances with small gradients and uses the rest to estimate the information gain. EFB bundles mutually exclusive features to reduce the number of features.

(3) TF-IDF (https://github.com/mayank408/TFIDF, accessed on 7 September 2021). This method uses the TF-IDF representation of academic papers, calculates the similarity of TF-IDF vectors between papers, and finally disambiguates through DBSCAN clustering.

(4) GraRep (https://github.com/ShelsonCao/GraRep, accessed on 7 September 2021) [30]. This method aims to capture the k-order proximities by factorizing the k-step transition matrices.

(5) metapath2vec (https://github.com/apple2373/metapath2vec, accessed on 7 September 2021) [23]. This method leverages meta-path-based random walks and a skip-gram model to perform node embedding.

(6) Deepwalk (https://github.com/phanein/deepwalk, accessed on 7 September 2021) [31]. This method performs a random walk on networks and then learns low-dimensional node vectors via the skip-gram model.

5.3. Results

The method mentioned above was trained on the training set and then verified using the validation set, with Precision, Recall, and F1 used as evaluation indicators. The final experimental results are shown in Table 7.

Table 7. Name disambiguation experimental results (%) (Bold: Best).

	GSDPMM	LightGBM	TF-IDF	GraRep	metapath2vec	Deepwalk	Ours
Precision	0.5987	0.5896	0.3721	0.6190	0.6179	0.6024	**0.6834**
Recall	0.5921	0.8818	**0.8755**	0.6010	0.6172	0.6142	0.6872
F1	0.5181	0.5796	0.4094	0.6103	0.6135	0.6029	**0.6242**

Our method achieved an excellent F1 score on the verification set, with a value of 0.6242, and the Precision and Recall are relatively balanced. The overall results of GSDPMM and GraRep are not ideal. The main reason is that they only use semantic information in the paper and do not make reasonable use of its relationship information. The LightGBM method results indicate that clustering through partial supervision information is better than using single clustering, but it does not consider the features of the relationship between papers. The TF-IDF method is less effective, indicating that an unsupervised clustering method based on TF-IDF or Deepwalk does not effectively extract features for disambiguation. The improvement obtained with our method is mainly due to the consideration of the equal importance attributed to semantic features and relational features in the paper.

By extracting some names from the testing set, we can further analyze the effect of name disambiguation. The specific indicators are shown in Table 8.

Table 8. Name disambiguation results.

Name	Paper's Number	Author's Number	Precision	Recall	F1
Kenji_kaneko	148	5	0.7258	0.7701	0.7473
guohua_chen	828	24	0.8057	0.6524	0.7210
hai_jin	56	5	1.0000	0.6051	0.7540
guoliang_li	744	36	0.7149	0.8467	0.7752
jiang_he	416	10	1.0000	0.9855	0.9927
jianping_wu	219	21	0.9980	0.7229	0.8384
peng_shi	815	11	0.7288	0.8545	0.7867
xiaoyang_zhang	447	30	0.9755	0.9624	0.9689
mei_han	388	19	0.9786	0.9108	0.9435
d_zhang	223	16	0.8905	0.8971	0.8938
akira_ono	195	9	0.7508	0.9504	0.8389
bin_gao	398	16	0.9603	0.8812	0.9190
chao_yuan	555	63	0.7587	0.9367	0.8383
hong_yan_wang	84	18	0.7589	0.8718	0.8115
c_c_lin	105	9	0.8249	0.8205	0.8227

According to the table, our method performs better when the number of authors with the same name is small and when the number of articles by the authors is 20 or more. Additionally, our method achieves quite high Precision for some data, such as the authors "hai_jin" and "jiang_he" in the table, which shows that the method can predict different papers belonging to the same author very well. In summary, the performance of our method is relatively balanced, and the test results are improved compared with other methods. This shows that our method of extracting relationship features by combining heterogeneous graph networks generates good name disambiguation results.

6. Conclusions

In order to solve the problem of author name disambiguation, we propose a method that uses a heterogeneous graph network to disambiguate the papers. The paper data are analyzed and divided into semantic features and relational features. For semantic features, we use fastText to extract node features. For relational features, real-world complex relationships between different nodes are captured as comprehensively as possible by constructing a heterogeneous graph network. Finally, the merged features are clustered by DBSCAN. The results of multiple sets of experiments prove the effectiveness of our method.

Our work achieved good results on the problem of name disambiguation. However, there is still much room for improvement in the results of our method, especially for massive amounts of paper data, for which many complex name disambiguation issues have not been well resolved.

Author Contributions: X.Z. helped in conceiving the research, performing experiments, and writing the research paper. P.Z., Y.C. and R.D. helped in the development and writing of the research. Y.Z. supervised and helped in writing the research paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Science Foundation of China (Grant No. 62072015), China Association of Higher Education (Grant No. 2020XXHYB16), and Beijing Municipal Science and Technology Project (Grant No. KM202010005014).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. Some of the data are not publicly available because they are still being used in an ongoing project.

Acknowledgments: We would like to thank the Beijing Artificial Intelligence Institute, Faculty of Information Technology, Beijing University of Technology for providing us with all of the resources and tools that were necessary to carry out the research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Levin, M.; Krawczyk, S.; Bethard, S.; Jurafsky, D. Citation-based bootstrapping for large-scale author disambiguation. *J. Am. Soc. Inf. Sci. Technol.* **2012**, *63*, 1030–1047. [CrossRef]
2. Bo, D.; Wang, X.; Shi, C.; Zhu, M.; Lu, E.; Cui, P. Structural deep clustering network. In Proceedings of the Web Conference 2020, Taipei, Taiwan, 20–24 April 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 1400–1410.
3. Wang, R.; Mou, S.; Wang, X.; Xiao, W.; Ju, Q.; Shi, C.; Xie, X. Graph Structure Estimation Neural Networks. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; Association for Computing Machinery: New York, NY, USA, 2021; pp. 342–353.
4. Hu, B.; Zhang, Z.; Shi, C.; Zhou, J.; Li, X.; Qi, Y. Cash-out user detection based on attributed heterogeneous information network with a hierarchical attention mechanism. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January 27–1 February 2019; AAAI Press: Palo Alto, CA, USA, 2019; Volume 33, pp. 946–953.
5. Sun, Y.; Han, J. Mining heterogeneous information networks: Principles and methodologies. *Synth. Lect. Data Min. Knowl. Discov.* **2012**, *3*, 1–159. [CrossRef]
6. Nandanwar, S.; Moroney, A.; Murty, M.N. Fusing diversity in recommendations in heterogeneous information networks. In Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, Los Angeles, CA, USA, 5–9 February 2018; pp. 414–422.
7. Shi, C.; Hu, B.; Zhao, W.X.; Philip, S.Y. Heterogeneous information network embedding for recommendation. *IEEE Trans. Knowl. Data Eng.* **2018**, *31*, 357–370. [CrossRef]
8. Anchiêta, R.T.; Sousa, R.F.d.; Pardo, T.A. Modeling the Paraphrase Detection Task over a Heterogeneous Graph Network with Data Augmentation. *Information* **2020**, *11*, 422. [CrossRef]
9. Han, H.; Zha, H.; Giles, C.L. A model-based k-means algorithm for name disambiguation. In Proceedings of the 2nd International Semantic Web Conference (ISWC-03) Workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data, Sanibel Island, FL, USA, 20–23 October 2003.
10. Han, H.; Giles, L.; Zha, H.; Li, C.; Tsioutsiouliklis, K. Two supervised learning approaches for name disambiguation in author citations. In Proceedings of the 2004 Joint ACM/IEEE Conference on Digital Libraries, Tucson, AZ, USA, 11 June 2004; pp. 296–305.
11. Kang, I.S.; Na, S.H.; Lee, S.; Jung, H.; Kim, P.; Sung, W.K.; Lee, J.H. On co-authorship for author disambiguation. *Inf. Process. Manag.* **2009**, *45*, 84–97. [CrossRef]
12. Shin, D.; Kim, T.; Jung, H.; Choi, J. Automatic method for author name disambiguation using social networks. In Proceedings of the 2010 24th IEEE International Conference on Advanced Information Networking and Applications, Perth, Australia, 20–23 April 2010; pp. 1263–1270.
13. Schulz, C.; Mazloumian, A.; Petersen, A.M.; Penner, O.; Helbing, D. Exploiting citation networks for large-scale author name disambiguation. *EPJ Data Sci.* **2014**, *3*, 1–14. [CrossRef]
14. Zhang, R.; Shen, D.; Kou, Y.; Nie, T. Author name disambiguation for citations on the deep web. In *International Conference on Web-Age Information Management*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 198–209.
15. Sun, Y.; Han, J. Meta-path-based search and mining in heterogeneous information networks. *Tsinghua Sci. Technol.* **2013**, *18*, 329–338. [CrossRef]
16. Cai, H.; Zheng, V.W.; Chang, K.C.C. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Trans. Knowl. Data Eng.* **2018**, *30*, 1616–1637. [CrossRef]

17. Shi, C.; Zhang, Z.; Luo, P.; Yu, P.S.; Yue, Y.; Wu, B. Semantic path based personalized recommendation on weighted heterogeneous information networks. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, Melbourne, Australia, 18–23 October 2015; pp. 453–462.
18. Joulin, A.; Grave, E.; Bojanowski, P.; Mikolov, T. Bag of tricks for efficient text classification. *arXiv* **2016**, arXiv:1607.01759.
19. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
20. Hinton, G.E. Learning distributed representations of concepts. In Proceedings of the Eighth Annual Conference of the Cognitive Science Society, Amherst, MA, USA, 15–17 August 1986; Volume 1, p. 12.
21. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems;* Curran Associates Inc.: Long Beach, CA, USA, 2017; pp. 5998–6008.
22. Shi, C.; Li, Y.; Zhang, J.; Sun, Y.; Philip, S.Y. A survey of heterogeneous information network analysis. *IEEE Trans. Knowl. Data Eng.* **2016**, *29*, 17–37. [CrossRef]
23. Dong, Y.; Chawla, N.V.; Swami, A. metapath2vec: Scalable representation learning for heterogeneous networks. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and DATA Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 135–144.
24. Fu, T.y.; Lee, W.C.; Lei, Z. Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 6–10 November 2017; pp. 1797–1806.
25. Gori, M.; Monfardini, G.; Scarselli, F. A new model for learning in graph domains. In Proceedings of the 2005 IEEE International Joint Conference on Neural Networks, Montreal, QC, Canada, 31 July–4 August 2005; Volume 2, pp. 729–734.
26. Agesen, O. The cartesian product algorithm. In *European Conference on Object-Oriented Programming*; Springer: Berlin/Heidelberg, Germany, 1995; pp. 2–26.
27. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. Density-based spatial clustering of applications with noise. In Proceedings of the International Conference on Knowledge Discovery and Data Mining, Portland, OR, USA, 2–4 August 1996; Volume 240, p. 6.
28. Yin, J.; Wang, J. A model-based approach for text clustering with outlier detection. In Proceedings of the 2016 IEEE 32nd International Conference on Data Engineering (ICDE), Helsinki, Finland, 16–20 May 2016; pp. 625–636.
29. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. Lightgbm: A highly efficient gradient boosting decision tree. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 3146–3154.
30. Cao, S.; Lu, W.; Xu, Q. Grarep: Learning graph representations with global structural information. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, Melbourne, Australia, 18–23 October 2015; pp. 891–900.
31. Perozzi, B.; Al-Rfou, R.; Skiena, S. Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 701–710.