Unsupervised Author Disambiguation using Heterogeneous Graph Convolutional Network Embedding

Ziyue Qiao*, Yi Du*,[1], Yanjie Fu†, Pengfei Wang‡, Yuanchun Zhou*
*Computer Network Information Center, Chinese Academy of Sciences, Beijing, China
†Department of Computer Science, University of Central Florida, Orlando, USA
‡Alibaba DAMO Academy, Alibaba Group, China
Email: {qiaoziyue, duyi}@cnic.cn, yanjie.fu@ucf.edu, wpf228618@alibaba-inc.com, zyc@cnic.cn

***Abstract*—People share same names in real world. When a digital library user searches for an author name, he may see a mixture of publications by different authors who have the same name. Making distinctions between them is an important prerequisite to improve the quality of services and contents in digital libraries. The general task of author disambiguation is to associate publications which belong to an identical name or names with highly similar spellings to different people entities. In recent years, many researches have been conducted to solve this challenging task. However, some works rely heavily on external knowledge bases and manually annotated data. Some unsupervised learning based works require complex feature engineering. In this paper, we propose a novel and efficient author disambiguation framework which needs no labeled data. We first construct a publication heterogeneous network for each ambiguous name. Then, we use our proposed heterogeneous graph convolutional network embedding method that encodes both graph structure and node attribute information to learn publication representations. After that, we propose a graph enhanced clustering method for name disambiguation that can greatly accelerate the clustering process and need not require the number of distinct persons. Our framework can be continually retrained and applied on incremental disambiguation task when new publications are put in. Experimental results on two datasets show that our framework clearly performs better than several state-of-the-art methods for author disambiguation.**

***Index Terms*—Name Disambiguation, Network Embedding, Clustering, Graph Convolutional Network, Meta Path**

I. INTRODUCTION

Nowadays, digital libraries(DLs) have become an important source of scholarly information retrieval. When an user search an author name on a DL, he/she wants to obtain quick and accurate results from a huge set of publications that correspond to the query name. However, many DLs' search service provide a broad range of publications by different authors who share the same name. Using name disambiguation technologies to automatically distinguish publications of various authors can help users spend less effort to verify the search results. In this paper, we focus on the general task of author disambiguation, which is to associate publications which belong to an identical name or names with highly similar spellings to different people entities. Plenty of researches have been conducted to solve the name ambiguity problem. Supervised methods [1], [2] regard this problem as a classification of authors with same name, however, they require labeled data. A widely used and effective idea is to learn publication representations via multiple features, then measure the similarity between publications and identify whether they belong to the same author. However, representation learning based methods for this problem still have challenges.

The first challenge is how to learn effective representations for publications by extracting and combining different types of features, such as the semantic information of text information and discrete features (e.g., authors, venue). High quality representations play a critical role to quantify distinctions and similarities between publications [1]. The majority of existing solutions utilize biographical features such as title, abstract, organization, author and venue, as well as relationship features such as citation to generate these representations. Most works design relationships such as coauthorship between publications by these characteristics and construct publication networks, then use graph-based [3]–[5] or heuristic methods [6], [7] to learn the similarity between publications. Some methods [4], [8]–[10] use the the word co-occurrence information in publications' content to design relations between publications, which may lose semantic information. Some methods focus on the content information but cannot measure the high-order relationship among publications and usually leverage supervised algorithms. Besides, many methods solve the problem on homogeneous networks. They either construct network for each type of relation, or combine different relationships to construct one publication network. Those strategies ignore the heterogeneous relationship between publications.

The second challenge is how to efficiently determine the assignment of publications when we do not exactly know the number of distinct person for an ambiguous name. Some researches [5], [11] assume that the cluster number of ambiguous author K is known in advance, but in real world we actually have no clue about the number. Some [4], [8] propose different strategies to estimate the number. However, there are usually pre-defined parameters in these strategies. In addition, with the increasing data in digital libraries, clustering process becomes more time-consuming, more efficient clustering strategies for name disambiguation are demanded.

The third challenge is how to efficiently process the new published publications that may contain potentially ambiguous names. Many methods [3], [5], [6], [12], [13] are implemented on a static collection extracted from DLs, and need to run the author disambiguation process on the whole collection when new records are added in. However, this update solution

The research is supported by the National Key Research and Development Plan (2017YFC1601504), the Natural Science Foundation of China (61836013), the CNTC (China National Tobacco Corporation) Science and Technology Major Project (110201901027(SJ-06)), and the Guangdong Provincial Key Laboratory of Biocomputing (2016B030301007).

[1]Corresponding author.

978-1-7281-0858-2/19/$31.00 © 2019 IEEE
978-1-7281-0858-2/19/$31.00 ©2019 IEEE

discards the existing publication representations and clustering results. Besides, it is very computationally expensive as the size of DLs are becoming larger and larger. An ideal solution is to use incremental disambiguation approach that use existing clustering results as well as learned model parameters to generate the partitions of new publications without reprocessing the whole collection.

In this paper, we propose a framework to address the challenges above to our best. We construct a publication heterogeneous network which contains multiple types of relations. Then we propose a heterogeneous network embedding method based on graph convolutional network as well as a relation weight and meta-path guided random walk strategy. After that, we propose an efficient clustering strategy that does not need to specify the number of clusters or any other parameters. Lastly, we introduce the incremental disambiguation strategies. The main contributions of this work are summarized as follows:

1) We propose a heterogeneous graph convolutional network embedding method integrating multi-layer and heterogeneous relationships between publications and the semantic information of publications, and learn a low-dimensional representation of each publication with high quality for name disambiguation.
2) We design an efficient and effective framework for author name disambiguation. A graph enhanced cluster strategy is introduced to accelerate the clustering speed without requiring the number of distinct persons. We also propose incremental disambiguation strategies that can process new coming publications.
3) The proposed method is evaluated and analysed on two benchmark datasets. The performance of our solution is significantly better than several state-of-the-art methods.

II. PRELIMINARIES

Treating publications as nodes, by constructing a network connecting the publication nodes, the relationship information between publications can be well preserved and expressed. Because of the multiplicity of publications attributes, there are many types of relationships between them. Therefore, we can retain these complex contacts and construct a heterogeneous network of publications.

Definition 1 (Heterogeneous Network). *A heterogeneous network is a graph $G = (V, E, \mathcal{N}, \mathcal{R})$ in which each node $v \in V$ is associated with a node type mapping functions $\phi(v) : V \rightarrow \mathcal{N}$ and each relation $e \in E$ is associated with a relation type mapping function $\phi(e) : E \rightarrow \mathcal{R}$. This suggests that G has multiple node types and relation types, formulated as $|\mathcal{N}| + |\mathcal{R}| > 2$.*

Definition 2 (Weighted Heterogeneous Network). *A weighted heterogeneous network $G_w = (V, E, \mathcal{N}, \mathcal{R})$ is a heterogeneous network in which each relation has a weight. For a relation $e \in E$, its value can be expressed as $|e|$.*

Definition 3 (Publication Heterogeneous Network). *In our paper, the publication Heterogeneous Network (PHNet) that we construct can be seen as a special case of weighted heterogeneous network which has one type of nodes and three types of relations. For each ambiguous name a, its PHNet can be expressed as $G^a = (V, E, \mathcal{R})$. As shown in Figure 1(a), every publication is expressed as a single node in the PHNet and the set of relation types $\mathcal{R}$ includes **CoAuthor**(A), **CoVenue**(V), and **CoTitle**(T), which respectively represent three kinds of undirected relations between publications. The detailed definition of each is as follows:*

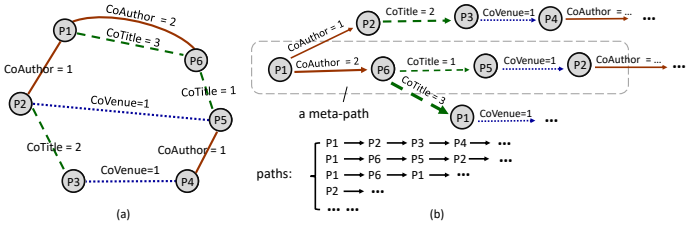


Fig. 1. An illustration of (a) a publication heterogeneous network with one node type, three relation types, and (b) paths sampled by meta-path ($r = CoAuthor \circ CoTitle \circ CoVenue$) and relation weight guided random walks.

CoAuthor *Each publication has a list of authors, so it can be coauthored by multiple names. For any name a to be disambiguated, suppose two of its publications are p_1^a and p_2^a, which have the author sets A_1 and A_2 respectively. Let the set $A_1^{'}$ denote A_1 excluding the author a, i.e., $A_1^{'} = A_1 - \{a\}$, If $A_1^{'} \cap A_2^{'} \neq \varnothing$, there is a CoAuthor relation between p_1^a and p_2^a, and the weight of the CoAuthor relation is $|A_1^{'} \cap A_2^{'}|$.*

CoVenue *If two publications are published at the same conference or journal, then we create a CoVenue relation between them. Usually a publication can only belong to one venue. Therefore, if the CoVenue relation exist, its value is defined as 1.*

CoTitle *Titles always contain a lot of topic words to summarize the main content of the publication. For each publication p_i's title, first, we remove the stop words and then transform each word into its word stem. After that, the title is processed into a words set T_i. For two publications p_1 and p_2, if $T_1 \cap T_2 \neq \varnothing$, there is a CoTitle relation between them, and the value of the CoTitle relation is $|T_1 \cap T_2|$.*

Other attributes of publications are also usable in our model when they are available. Considering that some attributes such as affiliation, citation maybe not available or incomplete in some digital libraries, especially the affiliation information usually has the synonym problem [14] and hard to be cleaned, so in this paper, we only use the three common attributes mentioned above. Except the ambiguous name a, the attributes of publications we used to construct PHNet maybe also ambiguous. For example, if two publications coauthored by a, in the same time, they each have another author b, and these two authors named b are actually two different people. But empirically, the probability of this coincidence happening is so small that the impact of these noises on the downstream model training can be ignored.

In a PHNet, two nodes p_i and p_j may be connected via multiple undirected relations. The node sequence linked by these relations can be seen as a path from p_i to p_j. For example, in Figure 1, there is a path $p_1 \xrightarrow{CoAuthor} p_2 \xrightarrow{CoTitle} p_3$ between p_1 and p_3, which means p_3 shares some title words with another publication written by one of p_1's authors.

Definition 4 (Meta-path). *In a heterogeneous network $G = (V, E, \mathcal{N}, \mathcal{R})$, a meta-path is defined as a path in the form of $n_1 \xrightarrow{r_1} n_2 \xrightarrow{r_2} ... \xrightarrow{r_l} n_{l+1}$, where $r_i \in \mathcal{R}$ and $n_i \in \mathcal{N}$,*

the sequence of relations $r = r_1 \circ r_2 \circ ... \circ r_l$ represents the composite relation including relations from type r_1 to type r_l.

For example, we can design $r = CoAuthor \circ CoTitle \circ CoVenue$ as the compositional relation of a meta-path. Then, in the PHNet, as shown in Figure 1(b), a path $p_1 \xrightarrow{CoAuthor=1} p_2 \xrightarrow{CoTitle=2} p_3 \xrightarrow{CoVenue=1} p_4$ is generated based on r.

Definition 5 (**Name Disambiguation Problem**). *Given an author name a to be disambiguated, we define $D^a = \{p_1^a, p_2^a, ..., p_m^a\}$ as the collection of m publications coauthored by any person named a, where the p_i^a means the i-th publication of author a. Each publication contains attributes such as authors list, title and venue (e.g., a conference or journal). Assuming that the number of distinct authors named a is K^a, our goal is to distinguish these publications into different author entities automatically by using the attribute information of them. In other words, we want to cluster the publications in D^a into K^a disjoint clusters $C^a = \{C_1^a, C_2^a, ..., C_k^a\}$, so that each publication within a cluster is authored by the same person and every two publications in different clusters have two different authors who share the same name a.*

III. Proposed Model

In this section, we first introduce the heterogeneous graph convolutional network embedding method which encodes the textual content of publications and graph structures of PHNets to learn the continuous distribution representations of publications. Then, we propose our clustering algorithm to partition the publications. At last, we introduce the incremental disambiguation strategies to disambiguate on new publications.

A. Publication Heterogeneous Network Embedding

1) Heterogenous Graph Convolutional Network Encoder: In this section, we introduce a Heterogenous Graph Convolutional Network (HGCN) module that operate on local graph neighborhoods to generate embeddings for nodes in PHNet, this model and related graph neural network models can be understood as special cases of a simple differentiable message-passing model.

Given a PHNet $G = (V, E, \mathcal{R})$, we start with initialize node attributes, we use Doc2vec [15] to encode the text information (i.e, title or abstract) of each publication $p_i \in V$ into a fixed length feature vector $u_i^{(0)}$. For each publication node, it has neighbors with maximal $|\mathcal{R}|$ types. Different from regular GCNs, we introduce relation-specific transformations that can aggregate the representations of neighbors under different types of relations. We define a multi-layer HGCN with following layer-wise convolutional rule for calculating the forward-pass update of each node p_i:

$$u_i^{(l+1)} = ReLU\left(\sum_{r\in\mathcal{R}}\sum_{j\in N_i^r}\frac{1}{c_{ij}^r}u_j^{(l)}W_r^{(l)}\right) \tag{1}$$

where $u_i^{(l)} \in \mathbb{R}^{1\times m^{(l)}}$ is the hidden state of node p_i in the l-th layer of the neural network and $m^{(l)}$ is the dimension of this layer's representation. $ReLU(\cdot) = max(0, \cdot)$, N_i^r denotes the set of neighbors indices of node p_i under the relation $r \in \mathcal{R}$, $c_{ij}^r = \sqrt{k_i^r k_j^r}$ is a normalization constant for the edge (p_i, p_j), where $\phi(p_i, p_j) = r$ and k_i^r is the sum of the weights of relations connecting to p_i and with type r. $W_r^{(l)}$ is a layer-specific and relation-specific trainable weight matrix. To ensure that the presentation of a node at each layer also has impact on one's presentation at the next layer, we assume there are single-connections of each type $r \in \mathcal{R}$ with weight 1 for each node.

Then, we define L layers in the HGCN model, where the output of the previous layer is the input to the next layer, $u_i^{(0)}$ is the input node attribute of 1st-layer. For each publication node p_i, HGCN encode the text embedding $u_i^{(0)}$ with its local neighborhoods on G to a representation u_i, formulated as:

$$u_i = u_i^{(L)} = \theta(u_i^{(0)}, G) \tag{2}$$

where θ is the parameters in HGCN. In each HGCN convolutional layer, each node receives information of one-hop neighbors to update its representation, the information of each node's neighbors under the same relation type share same transformation parameters. When multiple HGCN layers are stacked, the information of one-hop neighbors that each node receives has already includes their neighborhood information through last convolutional layer, with L layers a node can receives information from neighbors across multiple relational types at most L hops away.

2) Sampling Paths via Meta-path and Relation Weight Guided Random Walks: Our task is to train the HGCN model so that it can encode each publication node in PHNet to a high quality representation. Inspired by the network embedding methods DeepWalk [16] and Metapath2Vec [17] which use a random walk strategy and the skip-gram model to learning node representation in network. First, we propose a meta-path and relation weight guided random walk strategy to sampling paths on weighted heterogenous network.

The meta-path walks can capture correlation of nodes through heterogenous relations and widely used in heterogenous network embedding. Besides, We take the weights of relations in the PHNet into account while sampling paths. Intuitively, the larger value of a relation between two nodes, the more similarity of them. At each step, when the walker is walking to a neighbor, the higher value of a relation which connects the current node to a neighbor node, the more likely it is that this neighbor should be sampled.

Specifically, given a PHNet $G = (V, E, \mathcal{R})$, the meta-path scheme $\mathcal{P}$ is denoted in the form of $\mathcal{P} = \{p^t \xrightarrow{r_1} p^{t+1} \xrightarrow{r_2} ... \xrightarrow{r_l} p^{t+l+1}\}$, where $p^t \in V$, $r_i \in \mathcal{R}$ and l is the length of the composite relations between p^t and p^{t+l+1}. The transition probability when the walker at step t is defined as follows:

$$p(p^{t+1}|p^t, \mathcal{P}) = \begin{cases} \frac{|(p^t, p^{t+1})|}{||E^t(r_i)||} & (p^t, p^{t+1}) \in E, \phi(p^t, p^{t+1}) = r_i \\ 0 & (p^t, p^{t+1}) \in E, \phi(p^t, p^{t+1}) \neq r_i \\ 0 & (p^t, p^{t+1}) \notin E \end{cases} \tag{3}$$

where $r_i \in \mathcal{R}$ is the next relation type of $\phi(p^{t-1}, p^t)$ according to the predefined scheme $\mathcal{P}$, and if p^t is the first node of the path, $r_i = r_1$, $|(p^t, p^{t+1})|$ means the weight of the relation between p^t and p^{t+1}, $E^t(r_i)$ denotes the set of relations which connect p^t and those whose types are r_i. In addition, meta-path is recurrently sampled in G to generate

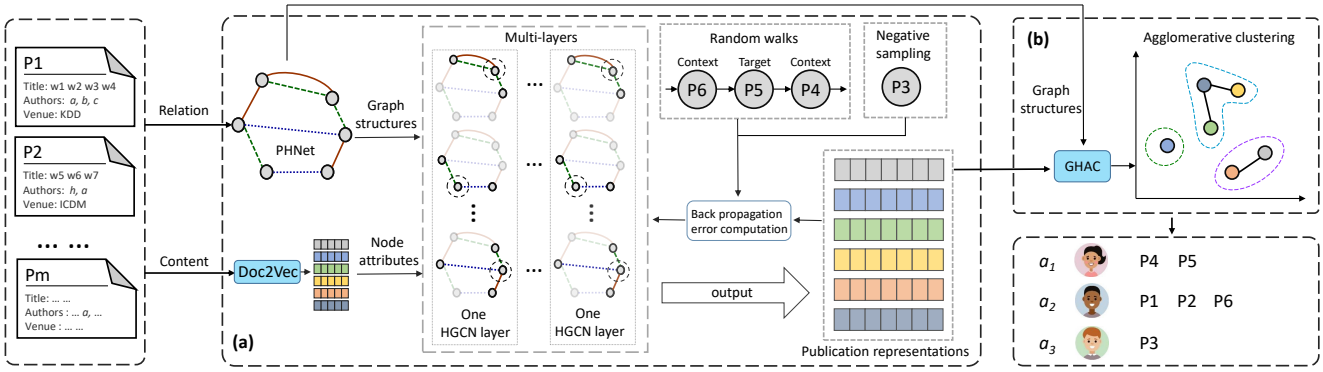


Fig. 2. An illustration of disambiguating the author name a. This framework consists of (a) the publication heterogeneous network embedding method to learn publication representations and (b) the graph-enhanced hierarchical agglomerative clustering method to partition the publications.

paths. Specifically, first, we successively select one publication node in the PHNet as the first node of the path and generate a meta-path whose length is l, and then choose the last node as the first node of another meta-path. In this way, each random work recursively samples nodes in the network to generate a long path guided by $\mathcal{P}$ until it meets the fixed length.

Taking the $\mathcal{P} = \{p^t \xrightarrow{CoAuthor} p^{t+1} \xrightarrow{CoTitle} p^{t+2} \xrightarrow{CoVenue} p^{t+3}\}$ (abbreviated as "-A-T-V-") as example, as shown in Figure 1(a), if the walker is sampling a next node, the current node is p_1, and the next relation type guided by $\mathcal{P}$ is *CoAuthor*, then all the nodes connected with p_1 with a *CoAuthor* type relation form the candidate sample set. Assuming that p_2 and p_6 are the only two nodes in this set, according to the value of the *CoAuthor* type relation they connect with p_1, the probability of the walker sampling node p_2 is $1/3$, and that of node p_6 is $2/3$. A demo of path sampling is illustrated in Figure 1(b), where the thickness of the arrow represents the probability of walking to the next node. In some cases, the walker meets a relation which is missing for the current node, then it skips this relation and walks to the next relation in the meta-path when this happens.

Compared with random walk, this meta-path guided random walk strategy avoids the biased influence caused by central nodes and deviation of different relation quantities via the ordered guidance of meta-path [17]. Besides, our designed random walk strategy considers the multiple possibilities of arrangement of a path, the multiple kinds of relationships between nodes in the path as well as the weight of the relations, which makes the generated paths can well preserve the heterogeneous relationships between nodes.

3) Weighted Heterogeneous Skip-gram Model: Based on the paths generated in the previous section, we proposed a Weighted Heterogeneous Skip-Gram (WHSG) model to learn the representations of publications on the weighted heterogeneous network. Given a PHNet $G = (V, E, \mathcal{R})$, we aim to learn effective node representations by maximizing probability of any node p_i having its neighbor node p_c:

$$\arg\max_{\theta} \sum_{p_i \in V} \sum_{r \in \mathcal{R}} \sum_{p_c \in N_r(p_i)} |(p_c, p_i)| \log p(p_c|p_i, \theta) \quad (4)$$

where $N_r(p_i)$ is a set of neighbors of p_i, and $\forall p_c \in N_r(p_i), (p_c, p_i) \in r$, θ represents the parameters of the HGCN model, The relation weight $|(p_c, p_i)|$ in this function is a weight to make those neighbors who has large relation value with p_i have higher probability output. $p(p_c|p_i, \theta)$ is defined as a softmax function:

$$p(p_c|p_i, \theta) = \frac{\exp(u_i \cdot u_c)}{\sum_{j=1}^{|D|} \exp(u_i \cdot u_j)} \quad (5)$$

where u_i represents the embedding vector of p_i encoded from its initial features $u_i^{(0)}$ by HGCN in eq.2. Then, we adopt the popular negative sampling method proposed in [18] to sample negative nodes to increase the optimization efficiency. Then, the probability can be approximately defined as:

$$\log p(p_c|p_i, \theta) \approx \log \sigma(u_i \cdot u_c) + \sum_{p_j \in D_{neg}^i} \log \sigma(-u_i \cdot u_j) \quad (6)$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function and D_{neg}^i is a negative node set for p_i sampled from a pre-computed node frequency distribution in paths.

Given the meta-path scheme $\mathcal{P}$ and relation weight in the network, we can generate a set of collected random walks $RW^{\mathcal{P}}$ by choosing different nodes as the start nodes of paths. The frequency of a neighbor appears in the context of the target node is in proportion to how close the relationship between them. so maximizing the objective in eq.4 is approximately equivalent to minimizing the loss function:

$$\mathcal{L}^{\mathcal{P}} = -\sum_{w \in RW^{\mathcal{P}}} \sum_{p_i \in w} \sum_{p_c \in C_i^k} \log p(p_c|p_i, \theta) + \lambda \|\theta\|^2 \quad (7)$$

where k is the windows size of context, C_i^k is the context set containing the previous k context nodes and next k context nodes of p_i in the walk w. In the generated random walks, the nodes in context set with different windows size represents different kinds of neighbors. For example, when $k = 2$, the context set contains the 1-hop and 2-hop neighbors. Parameter λ controls penalty of regularization for over-fitting.

The framework of our proposed network embedding method is illustrated in Figure 2(a). Our method is similar with many meta-path and skip-gram based heterogeneous network embedding models, comparing with these models, the WHSG model has these main differences: (1) our model can be applied

on the weighted heterogenous network and can capture both the relation weight information and heterogenous relations between nodes to generate node representations, (2) and more importantly, the representations are encoded by HGCN propagation model for integrating publication semantic information and local heterogenous graph structure information.

Finally, we use a mini-batch Adam optimizer to minimize $\mathcal{L}^{\mathcal{P}}$. Our model is very efficient, all the parameters of our model are the transformation matrices $\{W_r^{(l)}\}$ in each layer, where $r \in \mathcal{R}$ and $l \in \{0, ..., L-1\}$, and the sizes of parameters are independent with the size of publication set. Besides, we use the alias table method [19] to sample neighbors and negative nodes, which only takes $O(1)$ time when repeatedly sampling nodes from the same discrete distribution. In this paper, we set the number of layers L as 2 and the output dimension of all convolutional layers to be equal.

B. Clustering Algorithm

For each author name to be disambiguated, we learn its publications' representations by the network embedding model in Section III-A. Then, clustering algorithms are used to assign publications into disjoint clusters, each belonging to a distinct author entities (shown in Figure 2(b)). We find that the publication number of each author is skews, which means most of publications belong to a few dominant authors. Hierarchical Agglomerative Clustering (HAC) method works well for skewed data and is widely used in many name disambiguation methods [1], [5], [9], [20], [21]. However, the HAC has the following drawbacks: its time complexity is high compared with some other clustering methods and it needs to take the number of clusters K as input while determining the number of clusters K is usually a Gordian knot. To address those problems, we propose an efficient clustering strategy without any pre-set parameter called Graph-enhanced Hierarchical Agglomerative Clustering (GHAC).

For each ambiguous name a, given its PHNet $G = (V, E, \mathcal{R})$ and its publication embeddings, we firstly reconstruct a homogenous publication graph $G_r = (V, E_r)$. Each node in G_r represent a publication of a, and the weight of the edge $e_{ij} \in E_r$ between node p_i and node p_j is formulated as:

$$|e_{ij}| = \sigma(u_i \cdot u_j) \cdot \delta((p_i, p_j) \in E) \tag{8}$$

Where $\sigma(\cdot)$ is the sigmoid function, $\delta(x)$ is 1 if x is true and 0 otherwise. So the range of edge weight in G_r is $(0, 1)$, We apply the idea of HAC to this weighted homogenous graph G_r. The similarity between two clusters is defined as the average of the weights of all their adjacent edges in G_r. Firstly, each sample is treated as one single cluster, then the two closest clusters (with biggest similarity) are merged in each step until the number of clusters reaches the specified K.

When K is unknown, we adopt an optimal modularity [22] partitioning mechanism to determine the partition of publications. The modularity M of a partition of nodes on graph G_r is defined as follows:

$$M = \frac{1}{2m} \sum_{p_i, p_j \in V} \left[|e_{ij}| - \frac{w_i w_j}{2m} \right] \delta(c_i = c_j) \tag{9}$$

Where $m = \frac{1}{2} \sum_{p_i, p_j \in V} |e_{ij}|$, which is the sum of weights of all edges in G_r, $w_i = \sum_{p_j \in V} |e_{ij}|$ representing the sum of weights of all edges connecting to node p_i, c_i represents the cluster which p_i belongs to. Specifically, in GHAC, the modularity M of G_a under current partition is calculated after each cluster merging process until there is no edge among clusters, then we choose the partition which achieves largest M as the final clustering result.

Reconstructing G_r takes $O(|E_r|)$ time (where $|E_r| < |E|$). For each clustering iteration, to find the two closest clusters takes a maximum of $|E_r|$ times adding or comparing operations, so the computational complexity of GHAC when specifying K is $O((|V| - K)|E_r|)$. When K is unknown, the value of modularity M at each clustering iteration can be calculated based on the value of which at last iteration because agglomerating clusters just increases the terms in eq.9 at each iteration. So the computational complexity of modularity in the whole clustering process is $O(|V|^2)$, and the computational complexity of GHAC is $O((|V| + |E_r|)|V|)$. It is worth mentioning that there are mainly scalar operations in our algorithm. Compared with traditional clustering algorithm, GHAC introduces topological structure of graph to avoid the distance calculations (mainly vector operations) between disconnecting publications to accelerate the clustering process. To further evaluate the efficiency of GHAC, we also implement sufficient experiments on it in Section IV-F.

C. Incremental Disambiguation

Our designed disambiguation model can efficiently deal with new publication records. Suppose a name a that needs to be disambiguated, we learn the presentations of its m existing publications in the set $D_a = \{p_1, p_2, ..., p_m\}$ and get the clustering result $C^a = \{c_1, c_2, ..., c_k\}$ of these publications. Suppose there comes m' new publications $D'_a = \{p_{m+1}, p_{m+2}, ..., p_{m+m'}\}$ having the author name a which also need to be assigned to distinct authors. First, we encode the text information of each publication p_i in D'_a into a semantic embedding $u_i^{(0)}$. Then, we use the learned network embedding model to generate the publication representations in D'_a and update those in D_a, There are two strategies:

Real-time. We take the new-coming publications as nodes and add them into the pre-constructed PHNet and add relations according to the definition 3. Based on the new PHNet G^* and the pre-learned parameters θ, we can get the representations of publications:

$$u_i = \theta(u_i^{(0)}, G^*), pi \in \{D_a, D'_a\} \tag{10}$$

Incremental training. In this strategy, after updating the PHNet, we sample a few new paths by successively selecting each new publication node in the PHNet as the initial node to generate new path (the number of which is positive correlation to m') via the random walk strategy introduced in section III-A2, then we use these new paths to train our network embedding model to update the parameters θ. By using updated θ^*, the new publication representations are expressed as:

$$u_i = \theta^*(u_i^{(0)}, G^*), pi \in \{D_a, D'_a\} \tag{11}$$

Unlike other network embedding methods, the parameters in our methods are independent of the size of the training dataset, so it is unnecessary to retrain a new model

when new publications are introduced. These two publication representation updating strategies can be alternately used in practice. If a small number of publication are added at a time, we use the real-time strategy to update the publications' representations. If the number of new publications is greater than a predetermined threshold, we can sample new paths on the updated PHNet to incrementally train our model.

The network G_r is reconstructed after updating the publication representations. We add new publication nodes, edges between new nodes as well as edges between new nodes and existing nodes to G_r according to eq.8. During the incremental clustering procedure, each publication in D'_a is treated as one single cluster. We combine them with the clustering result C^a of D_a as the initial clustering input of GHAC, then clusters are merged in each step to generate new clustering results. After clustering, new publications are partitioned into the existing clusters or new clusters, each represents a different person.

IV. EXPERIMENTS

We perform several experiments to validate the performance of our proposed author name disambiguation method on two datasets. The results demonstrate the superiority of our proposed method over several state-of-the-art methods.

A. Datasets

To evaluate the proposed method, we use two datasets from Aminer[1] and CiteSeeX[2], which are two widely used benchmarks in author name disambiguation. These datasets both contain several ambiguous author name references with respective label for each person entity. The Aminer disambiguation dataset contains 110 ambiguous names, 1515 distinct authors and 7022 publications. CiteSeeX contains 14 ambiguous names, 468 distinct authors and 8453 publications.

B. Comparison Methods

To validate the performance of our proposed method, we compare our method against several different methods of name disambiguation. The brief descriptions of these baseline methods are as follows:

Component: This method simply partitions each PHNet into connected components to generate the clustering results for each name to be disambiguated.

Zhang et al. [1]: This method uses a global metric learning and local linkage learning based on a graph auto-encoder method to learn the publications embeddings, then it propose an end-to-end model to estimate the number of clusters using a recurrent neural network and use HAC to determine the assignment of publications.

Xu et al. [4]: For an ambiguous name, this method build several publication networks for different kinds of relations, then use a merge strategy to coarsen networks. Publications embeddings are learned by a novel network embedding method, the final result of clustering is determined adaptively based on the results of HDBSCAN and AP clustering algorithm.

Zhang et al. [5]: This method constructs three different networks on relational data for each ambiguous name's publication set, and use a network embedding method to learn the publication embeddings, then use a HAC algorithm to cluster publications to different person entity.

As our framework is mainly based on a heterogeneous network embedding method, we design some author disambiguation methods based on several network embedding model. For a fair comparison, We use them to learn publication representations on the networks with same construction as PHNet, and use HAC to generate the clustering results:

DeepWalk [16]: DeepWalk is a network embedding method based on random walks to learn latent node representations and it is only applicable for homogeneous unweighted network.

LINE [23]: LINE can preserve both the first-order and second-order proximities of nodes and is applicable for homogeneous weighted network.

Metapath2Vec [17]: Metapath2Vec is applicable for heterogeneous network with binary edges. It uses meta-path-based random walks to construct the heterogeneous neighborhood of a node and then leverages a heterogeneous skip-gram model to perform node embeddings.

Hin2Vec [24]: Hin2Vec is also an unweighted heterogeneous network embedding method that can capture rich semantic of relationships and the details of network structure to learn representations of nodes.

GraphSAGE [25]: GraphSAGE learn node embeddings through different aggregation function form a node local neighborhood, which is similar with our designed HGCN layer, but the difference is the unsupervised loss function and it can not be used for heterogeneous network. We use the GCN module as the aggregator of GraphSAGE.

C. Experiment Settings

Only three features(title, author, and venue) of publications are used in each method, so the performance of some baselines maybe a little lower than that demonstrated in their own paper. These comparison methods use different kinds of cluster strategy. For example, [5] need to specify the number of distinct author, [1] need labeled data to estimate the number. For a fair comparison, we assume the number of clusters is set to real value and choose HAC as the clustering method of [4]. We use pairwise F1-score [14] to evaluate the clustering results of our method and the compared ones. We also calculate the Macro-F1 score on each dataset, which is the average of F1 scores of disambiguation results on all ambiguous names. We run all experiments on a desktop computer with Inter Core i7-8700U CPU (3.2 GHz) and 16GB memory.

There are a few tunable parameters in our method: The learning rate of our network embedding method is 10^{-3}, the regularization coefficient λ is 10^{-4}. We also set the windows size of context k in weighted heterogeneous skip-gram mode as 2, the meta-path scheme we used is "-A-V-A-T-". We set the dimension of presentation vectors learned by all methods to be 64. The parameter setting for other methods are based on default values or the values specified in their own papers. All the experiments are repeated many times to make sure the results can reflect the performances of methods.

D. Experimental Results

Table I and II show performance between our proposed method and compared methods on two datasets. For example, as shown in Table I, considering the name "Jie Tang" which

[1]https://www.aminer.cn/disambiguation

[2]http://clgiles.ist.psu.edu/data/

TABLE I
THE PERFORMANCE OF DIFFERENT METHODS ON AMINER DISAMBIGUATION DATASET

Name	Our method	Component	Zhang et al. 2018 [1]	Xu et al. 2018	Zhang et al. 2017 [5]	DeepWalk	LINE	Metapth2Vec	Hin2Vec	GraphSAGE
Ajay Gupta	**0.750**	0.329	0.568	0.552	0.618	0.370	0.578	0.298	0.684	0.654
Alok Gupta	**1**	0.690	0.689	0.892	0.590	0.582	0.835	0.663	0.734	0.651
Bin Yu	**0.696**	0.292	0.431	0.585	0.614	0.490	0.475	0.354	0.490	0.441
David Cooper	0.900	0.327	0.737	0.884	**0.931**	0.737	0.833	0.833	**0.931**	0.862
David Nelson	**0.944**	0.219	0.750	0.735	0.556	0.353	0.523	0.788	0.635	0.710
Fei Su	**1**	0.648	0.933	0.630	0.941	0.684	0.721	0.930	0.917	0.948
Hao Wang	0.604	0.086	0.403	0.557	0.543	0.382	0.400	0.420	**0.624**	0.192
Jie Tang	**0.982**	0.883	0.657	0.522	0.910	0.738	0.432	0.902	0.825	0.741
Thomas Wolf	**0.860**	0.502	0.703	0.522	0.352	0.320	0.357	0.390	0.516	0.710
Yang Wang	0.548	0.118	0.273	**0.574**	0.409	0.171	0.211	0.310	0.443	0.204
Avg.	**0.786**	0.507	0.715	0.681	0.680	0.563	0.606	0.643	0.629	0.678

TABLE II
THE PERFORMANCE OF DIFFERENT METHODS ON CITESEEX DISAMBIGUATION DATASET

Name	Our method	Component	Zhang et al. 2018 [1]	Xu et al. 2018	Zhang et al. 2017 [5]	DeepWalk	LINE	Metapth2Vec	Hin2Vec	GraphSAGE
A Kumar	**0.648**	0.392	0.412	0.443	0.307	0.367	0.389	0.478	0.498	0.369
C Chen	**0.442**	0.091	0.299	0.437	0.384	0.155	0.239	0.274	0.431	0.248
D Johnson	0.736	0.454	**0.745**	0.696	0.667	0.487	0.613	0.595	0.590	0.729
J Martin	**0.731**	0.512	0.649	0.495	0.481	0.483	0.529	0.567	0.665	0.596
J Robinson	**0.695**	0.360	0.384	0.626	0.369	0.498	0.450	0.507	0.540	0.554
J Smith	**0.889**	0.201	0.613	0.824	0.753	0.296	0.671	0.796	0.717	0.657
M Brown	**0.802**	0.368	0.710	0.590	0.498	0.526	0.617	0.729	0.560	0.752
M Miller	**0.944**	0.578	0.730	0.913	0.885	0.566	0.621	0.621	0.639	0.895
S Lee	**0.602**	0.078	0.401	0.573	0.553	0.121	0.417	0.417	0.560	0.411
Y Chen	**0.777**	0.124	0.441	0.762	0.770	0.446	0.664	0.665	0.402	0.436
Avg.	**0.698**	0.288	0.538	0.646	0.561	0.429	0.507	0.547	0.563	0.523

belongs to 6 distinct real-life authors with one author has 89% publications of all their publications, our method achieves 0.982 F1 score on it. It proves that our method can handle well the imbalance distribution of publications(46% authers have written only one publication in their entire career [26], whereas some may have written more than 100 ones). The Macro-F1 score of methods' results on all names in each dataset(shown in last row) indicates that our method significantly outperforms all the baselines(+9.9-39.6% in Aminer, +8.0-62.7% in CiteSeeX). Paired t-test shows that all improvements are significant at the 0.05 level.

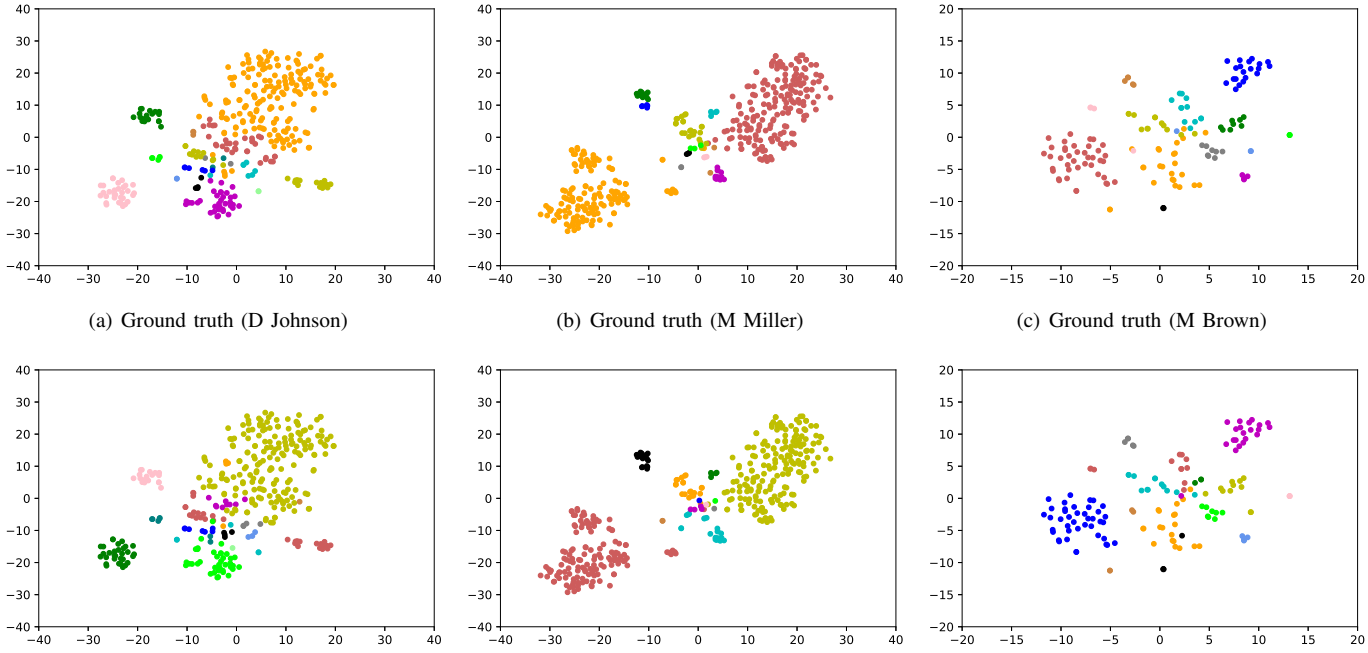
The reason why our proposed method has better performance is that our method is able to learn high quality publication representations by our HGCN based heterogenous network embedding method. Among the compared methods, both Zhang et al. [5] and Xu et al. construct several graphs based on various relationships between publications, and use network embedding based methods to learn representations of publications. However, Zhang et al. [5] ignore the text information, and Xu et al. just use the word co-occurrence information of text and loss a certain amount of semantic information. Zhang et al. [1] also use a graph convolutional network based encoder-decoder model but on homogeneous graph that can not extract multi-layer relationship that contains various relation types. However, our designed network embedding model can integrate publication attributes and heterogeneous relationship between publications into embeddings.

In network embedding, The great superiority of our method to DeepWalk and Metapath2Vec indicates that our designed relation weight and meta-path guided random walks brings large benefit to skip-gram and random walk based network embedding model. LINE does not consider the types of relation in the PHNet, Metapath2Vec and Hin2Vec does not consider different relation weights, while our model focus on learning latent embeddings on weighted heterogeneous networks, it has the ability to integrate these two features of network into effective embeddings. Our method is superior to GraphSAGE which proves that our heterogeneous graph convolutional network aggragator and loss function is more suitable for author disambiguation task.

Figure 3 shows the spatial distribution of publications on the embedding spaces learned by our complete method and the performance of our designed clustering strategy. For each name, we can easily observe that in Figure 3(a,b,c), publications from different clusters are well separated in embedding space. Figure 3(d,e,f) demonstrates the clustering result generated by GHAC method, most of the publications points are assigned into the correct clusters. It proves that in such embedding space, our designed GHAC is very advantageous on the clustering task and can achieve great performance.

E. Meta Path Analysis

To understand the impacts of different meta-path schemes on the network embedding model's performance, we design four kinds of meta-path scheme: "-A-V-","-V-T-","-A-V-T-" and "-A-V-A-T-" and compare them with pure random walk. Then, we use five different random walk strategies respectively to generate paths to train our network embedding model, and



(a) Ground truth (D Johnson) (b) Ground truth (M Miller) (c) Ground truth (M Brown)

(d) Predicted result (D Johnson,F1:0.736) (e) Predicted result (M Miller,F1:0.944) (f) Predicted result (M Brown,F1:0.802)

Fig. 3. t-SNE Visualization of embedding spaces on publications of three different names (D Johnson, M Miller, M Brown). Each color in (a), (b), (c) denotes an ground truth cluster which contains publications of a distinct author entity, while each color in (d), (e), (f) denotes a predicted cluster generated by our graph enhanced hierarchical agglomerative clustering.

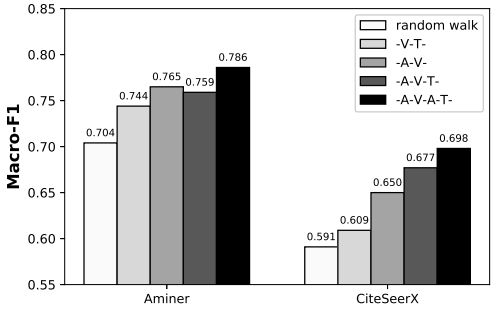


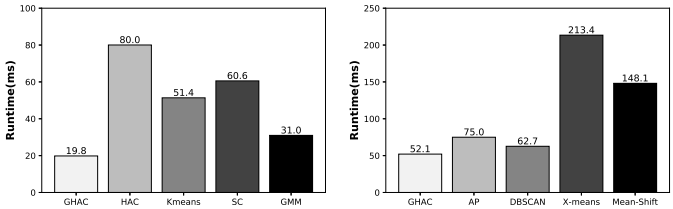
Fig. 4. The author disambiguation performance by using different meta-path guided random walk strategies.

the learned publication representations are used to generate clustering results by GHAC.

From the experiment results in terms of Macro-F1 on two datasets in Figure 4, we can observe that using these four meta-paths to guide the random walk will achieve better performance than not using meta-paths, and using the meta-path "-A-V-A-T-" can obtain relative higher macro-F1 than using other three.

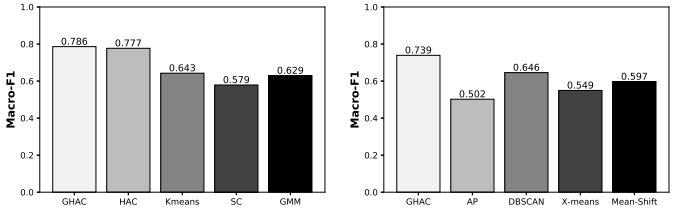
F. Clustering Algorithm Analysis

As our clustering method is suitable for either the cases of knowing the number of clusters or not, we compare the performance of GHAC with HAC, K-means, Gaussians mixture model (GMM) and Spectral Clustering (SC) by specifying K. We also compare GHAC with Affinity Propagation (AP),



(a) K is known (b) K is unknown

Fig. 5. The average running time of different clustering methods on Aminer dataset.



(a) K is known (b) K is unknown

Fig. 6. The performance of different clustering methods on Aminer dataset.

DBSCAN, X-means, and Mean-Shift (MS), which does not require the number of clusters.

Figure 5 shows average running time required for clustering the publications by each method. GHAC is significantly superior to most methods because it only compute the similarity between publication that has edges in the reconstructed graph, which is usually sparse in a real-life scenario.

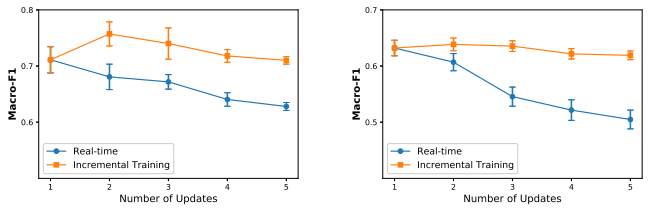
As shown in Figure 6(a), the Macro-F1 of GHAC's clus-

TABLE III
RESULTS OF CLUSTERING SIZE ESTIMATION

Name	Actual	GHAC	AP	DBSCAN	X-means	MS
Alok Gupta	2	1	8	7	2	4
Bin Yu	17	17	13	19	7	1
Bing Liu	18	14	26	28	7	4
David C. Wilson	5	5	10	7	5	6
David E. Goldberg	3	2	30	29	9	15
Fei Su	4	4	4	4	2	5
Gang Chen	47	31	20	36	8	1
Hiroshi Tanaka	7	7	8	8	3	7
Ji Zhang	16	15	10	13	3	1
Yue Zhao	9	10	5	4	2	11
MSLE	-	**0.206**	0.692	0.490	1.541	2.510
Accuracy	-	**21.1%**	5.5%	9.2%	7.3%	10.1%

tering result on Aminer dataset is higher than all the comparison methods, which indicates that the idea of hierarchical agglomerating is effective in solving the skewed publication data. Our method is slightly higher than HAC, which indicates that ignoring the relationship information among publications that has no edge connections in the reconstructed graph can not only speed up the clustering process, but also improve the clustering results.

Figure 6(b) shows that our method outperforms all other methods when the number of clusters is unknown. The reason is that using optimal modularity partitioning mechanism can well extract the community relations among publications from the reconstructed graph and find the better partition result of publications. We also compared our method with others in predicting the number of distinct authors when K is unknown. As shown in Table III, our method achieves a MSLE of 0.206 and an accuracy of 21.1%, which means that our estimated Ks are within a reasonable error range for most names.



(a) Aminer (b) CiteSeeX

Fig. 7. The performance of two incremental disambiguation strategies.

G. Incremental Disambiguation Analysis

There are two strategies on incremental disambiguation in our framework: "Real-time" and "Incremental training". To evaluate them, we first choose the ambiguous names with over 100 publications in two datasets, and equally split each name's publications into 5 batches. We randomly select one batch as the existing publications and use our framework to get its partition result. Then we use two incremental disambiguation strategies to respectively process each of the rest batches in the sequential order of arrival.

Figure 7(a) and Figure 7(b) show the performances of two incremental strategies on two datasets, the mean and standard deviation were computed at each update. We can observe that with the arrive of each new publications batch, in real-time updating strategy, the macro-F1 scores are slightly decreasing on two datasets, and in incremental training the scores maintain at a high level after 5 batches. These results prove that for a small set of new publications, real-time updating is effective way to disambiguate as it is more time-saving than incremental training, and the incremental training strategy performs better when conducted on large new publication dataset.

V. RELATED WORK

Name Disambiguation. To solve name disambiguation problem, most of existing researches select various features from digital libraries and use them to quantify the similarity of publications, then cluster them into disjoint clusters, each of the which belongs to a distinct person. [8] propose a Markov random fields based framework to extract multiple types of characteristics and relations in publication database. [1] use a global metric learning and local linkage graph auto-encoder algorithm to learn the representation of publications, but it requires lots of human labeled data to train the model. [9] propose a hierarchical agglomerative clustering based approach using the coauthor and title attributes. Other works attempt to utilize graph topology and linkage to solve this problem. [3] construct a publication network only by co-author relation and devise a novel similarity metric. Then they use affinity propagation clustering algorithm to group result into clusters. [13] introduce a pairwise factor graph model which can be extended by incorporation various features. The vast majority of name disambiguation solutions are conducted on static datasets. To disambiguate the new introduced publications in DLs, some works design incremental name disambiguation methods, [27] propose a probabilistic model that use a rich set of metadata and classifies new publications to existing author entities. [28] combines several domain-specific heuristics in order to automatically create and update publication clusters and determine the author of each publication.

Network Embedding. Recently, there has been a growing interest in the network embedding technology. DeepWalk [16] and Node2Vec [29] use random walk strategy on network and skip-gram [30], [31] model to learn the representation of each node in network. However, when they are applied on heterogeneous networks, such random walks ignore the types of relations, are biased by highly visible relation types and concentrated nodes, and generate incorporated node paths [32]. Metapath2Vec [17] proposes an embedding method on heterogeneous network based on meta-path. Some other methods have offered different solutions to network embedding. LINE [23] aims to learn the node embedding that preserve both first-order and second-order proximities. Graph neural network (GNN) based methods that applies deep neural networks on graph-structured data are significant developed in recent years, GNNs can also be used as a node encoder to learn network embeddings. DNGR [33] proposes a deep neural networks based method to learn a low-dimensional vector representation for each vertex by capturing the graph structural information. GCN [34] and GraphSage [25] use graph convolution neural networks to obtain node representations. [35] introduces a relational graph convolutional network to link prediction task and entity classification task. [36] propose a heterogeneous graph neural network model which considers both types and heterogeneous attributes of nodes. There are a large amount of works on name disambiguation

that attempt to use network embedding methods to learn publication embeddings by constructing publication networks. [5] utilize a network representation learning based approach on three anonymized network, [4] construct five relationship networks among publications and use a network embedding algorithm to learn representations of publications via their neighbors, [21] introduce a simple random walk strategy and a graph embedding method on a homogeneous network and use HAC to partition the publications. However, these methods lacks deeper model to learn publication representations. To the best of our knowledge, there has been no heterogeneous graph neural network embedding based methods implemented on name disambiguation problem so far.

VI. CONCLUSION

In this paper, we propose an effective framework to address the author disambiguation problem. Our framework consists of a novel heterogeneous graph convolutional network embedding method to learn publication representations, an efficient clustering method to determine the partition of publications, and two incremental disambiguation strategies for new introduced publications. Experiments on two datasets show that our pipeline can learn publication representations with higher quality than state-of-the-art methods, the clustering strategy is better than baselines in name disambiguation task, and the incremental disambiguation is effective. In our future work, we will try to apply our framework on distributed computing system to further improve the disambiguation speed on large academic database.

REFERENCES

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

[2] H. N. Tran, T. Huynh, and T. Do, "Author name disambiguation by using deep neural network," in *ACIIDS*. Springer, 2014, pp. 123–132.

[3] X. Fan, J. Wang, X. Pu, L. Zhou *et al.*, "On graph-based name disambiguation," *JDIQ*, vol. 2, no. 2, p. 10, 2011.

[4] J. Xu, S. Shen, D. Li, and Y. Fu, "A network-embedding based method for author disambiguation," in *CIKM*. ACM, 2018, pp. 1735–1738.

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

[6] A. F. Santana, M. A. Gonçalves, A. H. Laender, and A. A. Ferreira, "On the combination of domain-specific heuristics for author name disambiguation: the nearest cluster method," *JODL*, vol. 16, no. 3-4, pp. 229–246, 2015.

[7] W.-S. Chin, Y. Zhuang, Y.-C. Juan, , and F. o. Wu, "Effective string processing and matching for author disambiguation," *JMLR*, vol. 15, no. 1, pp. 3037–3064, 2014.

[8] J. Tang, A. C. Fong, B. Wang, and J. Zhang, "A unified probabilistic framework for name disambiguation in digital library," *TKDE*, vol. 24, no. 6, 2012.

[9] X. Lin, J. Zhu, Y. Tang, F. Yang *et al.*, "A novel approach for author name disambiguation using ranking confidence," in *DASFAA*. Springer, 2017, pp. 169–182.

[10] Y. Qian, Q. Zheng, T. Sakai, J. Ye, and J. Liu, "Dynamic author name disambiguation for growing digital libraries," *Information Retrieval Journal*, vol. 18, no. 5, pp. 379–412, 2015.

[11] H. Han, L. Giles, H. Zha, C. Li *et al.*, "Two supervised learning approaches for name disambiguation in author citations," in *JCDL*. ACM, 2004, pp. 296–305.

[12] S. Cucerzan, "Large-scale named entity disambiguation based on wikipedia data," in *EMNLP-CoNLL*, 2007.

[13] X. Wang, J. Tang, H. Cheng, and S. Y. Philip, "Adana: Active name disambiguation," in *ICDM*. IEEE, 2011, pp. 794–803.

[14] I. Hussain and S. Asghar, "A survey of author name disambiguation techniques: 2010–2016," *The Knowledge Engineering Review*, vol. 32, 2017.

[15] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International conference on machine learning*, 2014, pp. 1188–1196.

[16] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *SIGKDD*. ACM, 2014, pp. 701–710.

[17] Y. Dong, N. V. Chawla, and A. Swami, "metapath2vec: Scalable representation learning for heterogeneous networks," in *SIGKDD*. ACM, 2017, pp. 135–144.

[18] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[19] A. Q. Li, A. Ahmed, S. Ravi, and A. J. Smola, "Reducing the sampling complexity of topic models," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 891–900.

[20] S. Zhang, E. Xinhua, and T. Pan, "A multi-level author name disambiguation algorithm," *IEEE Access*, 2019.

[21] W. Zhang, Z. Yan, and Y. Zheng, "Author name disambiguation using graph node embedding method," in *2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. IEEE, 2019, pp. 410–415.

[22] M. E. Newman, "Modularity and community structure in networks," *Proceedings of the national academy of sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[23] J. Tang, M. Qu, M. Wang, M. Zhang *et al.*, "Line: Large-scale information network embedding," in *WWW*. International World Wide Web Conferences Steering Committee, 2015, pp. 1067–1077.

[24] T.-y. Fu, W.-C. Lee, and Z. Lei, "Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 2017, pp. 1797–1806.

[25] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems*, 2017, pp. 1024–1034.

[26] V. I. Torvik and N. R. Smalheiser, "Author name disambiguation in medline," *TKDD*, vol. 3, no. 3, p. 11, 2009.

[27] Z. Zhao, J. Rollins, L. Bai, and G. Rosen, "Incremental author name disambiguation for scientific citation data," in *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2017, pp. 175–183.

[28] A. F. Santana, M. A. Gonçalves, A. H. Laender, and A. A. Ferreira, "Incremental author name disambiguation by exploiting domain-specific heuristics," *Journal of the Association for Information Science and Technology*, vol. 68, no. 4, pp. 931–945, 2017.

[29] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *SIGKDD*. ACM, 2016, pp. 855–864.

[30] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[31] X. Rong, "word2vec parameter learning explained," *arXiv preprint arXiv:1411.2738*, 2014.

[32] Y. Sun, J. Han, X. Yan, P. S. Yu *et al.*, "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks," *Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, 2011.

[33] S. Cao, W. Lu, and Q. Xu, "Deep neural networks for learning graph representations," in *AAAI*, 2016, pp. 1145–1152.

[34] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[35] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *European Semantic Web Conference*. Springer, 2018, pp. 593–607.

[36] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, "Heterogeneous graph neural network," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2019, pp. 793–803.