Bibliographic Name Disambiguation with Graph Convolutional Network

Hao Yan[1,2(✉)], Hao Peng[1,2], Chen Li[1,2], Jianxin Li[1,2], and Lihong Wang[3]

[1] Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing, China
{yanhao,penghao,lichen,lijx}@act.buaa.edu.cn

[2] State Key Laboratory of Software Development Environment, Beihang University, Beijing, China

[3] National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing, China
wlh@isc.org.cn

Abstract. Name disambiguation, which aims to distinguish real-life person from documents associated with a same reference by partition the documents, has received extensive concern in many intelligent tasks, e.g., information retrieval, bibliographic data analysis and mining system. Existing methods implement name disambiguation utilizing linkage information or biographical feature, however, only a few work try to combine them effectively. In this paper, we propose a novel model that incorporates structural information and attribute features based on the Graph Convolutional Network to learn discriminating embedding, and achieves individual distinction by equipping a hierarchical clustering algorithm. We evaluate the proposed model on real-world academic networks Aminer, and experimental results show that the proposed method is competitive with the state-of-the-art methods.

Keywords: Name disambiguation · Graph Convolutional Network · Clustering

1 Introduction

While you are searching for academic publications by an author name, the response may disappoint you. For instance, sometimes you want to peruse masterpieces of "Tom Mitchell", a professor of Carnegie Mellon University, well-known in machine learning fields. After typing the name in search box, the query result is a long list of papers having a author named "Tom Mitchell". Unexpectedly, the topics ranging from computer science, biology to economics, and only several papers is relevant to the scholar you concerned. It would be better if you search author in digital libraries, for example, DBLP[1], Cite Seer[2] and Aminer[3]. These search engine will list candidates named "Tom Mitchell"

[1] http://dblp.uni-trier.de.
[2] http://citeseerx.ist.psu.edu.
[3] http://www.aminer.cn.

R. Cheng et al. (Eds.): WISE 2019, LNCS 11881, pp. 538–551, 2019.
https://doi.org/10.1007/978-3-030-34223-4_34

with corresponding institute. It is much easier to find the scholar you searching, clicking the target candidate, and you will acquire all his publications. Technology behind the convenience is a lot of machine learning algorithms including name disambiguation [6,26].

Name disambiguation is an important problem, which has numerous applications in information retrieval, bibliographic data analysis and other fields [4,20]. In information retrieval, name disambiguation is crucial for understanding query purpose. As mentioned before, while querying "Tom Mitchell", name disambiguation is necessary to split query result into different groups according to entities behind the name. In addition to the literature search facility, digital libraries also provide useful analysis that is being used for better decision making by funding agencies and academic institutions for grants and individual's promotion decisions [13]. If publications of different persons with same name can not be attributed accurately, the analysis would be misleading.

Due to its importance, the name disambiguation problem has attracted substantial attention from information retrieval and data mining communities. Many existing methods [2,11] used biographical features to distinguish people with same name, e.g, name, address, institutional affiliation, email address, etc. But biographical features are hard to obtain and liable to change. Usually, publications can reveal author research fields and interests, such as the similarity between two papers is a clue to find whether they have same authorship. Recently work [18] solved name disambiguation problem based on paper attributes similarity, e.g, keywords and title. Other methods uses relational data in the form of a collaboration graph, and solved name disambiguation by using graph topological features. For instance, Hermansson [10] used a classification model based on graphlet kernels, and Zhang [28] used a network embedding based method on anonymized graphs. Through previous studies, we find that both attribute features and graph structural information have contribution to solve name disambiguation problem. It's well known that Graph Convolutional Network (GCN) [15] is an efficient model to integrate both attribute features and structural information. Zhang [29] proposed a graph auto-encoders [14] based method involving graph topology and attribute features, but this method neglects the linkage between papers and authors and co-authorship.

To utilize information as much as possible and achieve better performance, we propose a novel graph structure and attribute features involved representation learning model. Specifically, we make use of two personalized GCNs embeddings of papers and authors into a low-dimensional space, and then maintain close linked entities proximate to each other in embedding space with minimizing the careful designed objective function. Then, a Hierarchical Agglomerative Clustering (HAC) algorithm [8][4] could be integrated to solve name disambiguation problem. The proposed method is evaluated on real-world large-scale academic networks Aminer dataset. The experimental results show that our proposed method is competitive with several state-of-the-art methods.

[4] https://github.com/mstrosaker/hclust.

The remainder of this paper is divided into five sections. In Sect. 2, we briefly reviewing previous representative works directly related to ours. Then we detailed formulate name disambiguation problem, define three types of graph in Sect. 3 and introduced our method in Sect. 4. In Sect. 5, we show experiment results on real-world large data and compare our model with several state-of-the-art methods. In the end, we draw our conclusions.

2 Related Work

2.1 Name Disambiguation

Recently, name disambiguation has been defined as clustering problem. Previous studies have focused on how to strike the balance between documents similarity quantization, determination of cluster size, and achieving better disambiguation. According to the selection of clustering basis, the existing literature can be roughly classified into three categories: attribute features based, linkage based, and hybrid methods.

Attribute features based methods generally focus on how to measure the similarity between documents. The work of Huang [12] introduces Support Vector Machine (SVM) to distinguish candidate documents which are initially grouped according to name similarity, and then utilizes DBSCAN to cluster documents. Yoshida [27] achieves better cluster results by using efficient feature from a two-stage clustering process. Different from the unsupervised methods described above, Han [9] employs SVM and Naive Bayes to implement name disambiguation in a classified manner. Similarly, Louppe [18] proposes a semi-supervised hierarchical clustering method based on a classifier to achieve more efficient document similarity metrics.

Linkage based methods are more focused on the graph structure (composed of articles and authors) information than the attribute features based methods. GHOST [5] achieves node clustering on a co-author-based graph through mining the relationships between documents more granularly. By considering the linkage between documents as a transfer process, Tang [25] uses Hidden Markov Random Fields (HMRFs) to model the document chain uniformly and solve the name disambiguation using probability model. The work of Zhang [28] embeds documents into low-dimensional space without involving private data, and implements name disambiguation through HAC.

Besides, there are some methods try to combine the advantages of the two methods above. Zhang [29] proposes a novel representation learning method that can contain both global and local information, achieves a good performance, and was applied in Aminer.

2.2 Graph Convolutional Networks

As a method for efficiently integrating attribute features of graph structural information, GCN has been widely studied and applied. Firstly, Bruna [1] define the convolution operation in an irregular graph structure by using convolution Fourier formula, which has achieved competitive results. Defferrard [3] advance GCN through multi-order information diffusion, and implements an approximate calculation using the Chebyshev formula. The work of Kipf [15] proposes a first-order approximation GCN, which defines a new information diffusion matrix to achieve efficient node feature learning, and achieved good results in semi-supervised classification tasks.

Recently, GCN has been applied to a large number of tasks including graph mining, text classification, traffic prediction and event mining [17,21–23], and it has been verified that it can effectively combine structural information with node feature. Kipf [24] employ GCN to relation learning by equipping update module with an information passing component. Then apply GCN to the event detection with a novel pooling method and achieved better results [19]. To analyze the compositional principles of protein molecular networks, Alex [7] utilize GCN to obtain molecular embedding and model the composition of proteins. Besides, GCN is widely used in tasks, e.g., named entity recognition and relation extraction. To our best, we are the first to introduce GCN and triplet loss to solve name disambiguation problem.

3 Preliminaries

In this section, we first present the formulation of the problem, and then introduce the three types of graph used in our model.

3.1 Problem Formulation

Let a be a given name reference, and $\mathcal{D}^a = \{D_1^a, D_2^a, ..., D_N^a\}$ be a set of N documents associated with the author name reference a. $\{A_1, A_2, ..., A_M\}$ is the collaborator set of author named a in $\mathcal{D}^a$, denoted as $\mathcal{A}^a$ where $a \notin \mathcal{A}^a$. We assume there is no disambiguation in $\mathcal{A}^a$ that means each name reference could identify a collaborator. In real-life it is common that several person have same name. The goal of name disambiguation is to divide $\mathcal{D}^a$ into K disjoint sets $\mathcal{C}^a = \{C_1^a, C_2^a, ..., C_K^a\}$, in each set C_k^a, all documents belong to the same person p_k and documents associated with author p_k must in same set C_k^a. The problem could be formalized as follow.

Definition 1 Name Disambiguation. *Denote $\Theta(d_i^a)$ as a function to get the person p_k who named a and associated with d_i^a, the task of name disambiguation is to find a partition function Φ to divide $\mathcal{D}^a$ into K disjoint clusters, i.e.,*

$$\Phi(\mathcal{D}^a) \rightarrow C^a \tag{1}$$

every cluster in C^a meets $\forall d_i^a \in C_k^a, \Theta(d_i^a) = p_k$ and $\forall d_i^a \in \{d_i^a | \Theta(d_i^a) = p_k\}, d_i^a \in C_k^a$, that is equivalent to

$$C_k^a = \{d_i^a | d_i^a \in \mathcal{D}^a, \Theta(d_i^a) = p_k\} \tag{2}$$

3.2 Graphs in Bibliographic Domain

In bibliographic domain, such as dblp and Aminer, there are linked and attributed information we can utilize to solve this problem. For example, given two papers, the authors of the two papers collaborated closely with each other, there is a high probability that the two papers belong to the same author. This is appropriate that we assume the topic of papers associate to same author would be similar, because different scholar has his own interests and specific research fields. The paper's attribute information such as title, keywords, abstract and venue would reveal that. Besides, the author's attribute information is also useful to solve name disambiguation problem. We denote feature of i_{th} document associated with the author name a as f_i^d, similarly, feature of j_{th} person in collaborator set is represented as f_j^p. Document and person feature matrix is F_d and F_p respectively.

Definition 2 Person-Person Graph. *Given a name reference a, the person-person graph denoted as* $G_{pp} = (\mathcal{A}^a, E_{pp})$*, nodes in this graph is the collaborator set* $\mathcal{A}^a$*, the weight of* e_{ij} *is defined as the number of distinct documents in which* A_i *and* A_j *have collaborated.*

Definition 3 Document-Person Graph. *Given a name reference a, the person-document graph is represented as* $G_{dp} = (\mathcal{D}^a \cup \mathcal{A}^a, E_{dp})$*, a bipartite graph.* $\mathcal{D}^a$ *is documents associated with name reference a,* $\mathcal{A}^a$ *is the collaborator set. If a person node* A_j *is the author of a document node* D_i*, then the edge weight* w_{ij} *is 1, otherwise is 0.*

Definition 4 Document-Document Graph. *Given a name reference a, the graph is represented as* $G_{dd} = (\mathcal{D}^a, E_{dd})$*, if two documents are similar enough, build a edge between them. We measure the similarity of two documents based on the common features shared by the two documents, firstly, calculate IDF (Inverted Document Frequency) of each feature, and then sum up the IDF of common features shared by two documents, if the result above a threshold, set the weight* w_{ij} *between document i and document j to 1.*

In this study, we use two personalized GCNs embedding both structural and attribute features within three types of graph into a same d-dimensional space, and then we use document embedding matrix as input and applies HAC to partition $\mathcal{D}^a$ into K disjoint clusters. At this stage, K is a user-defined parameter which we match with the ground truth during the evaluation phase.

4 Methodology

In this section, we discuss the design and implementation of our method to solve author disambiguation problem in detail. Our work concentrate on the representation learning of documents and persons. As embedding acquired, HAC is applied as other clustering based methods did. The overview of our embedding model is shown as Fig. 1.

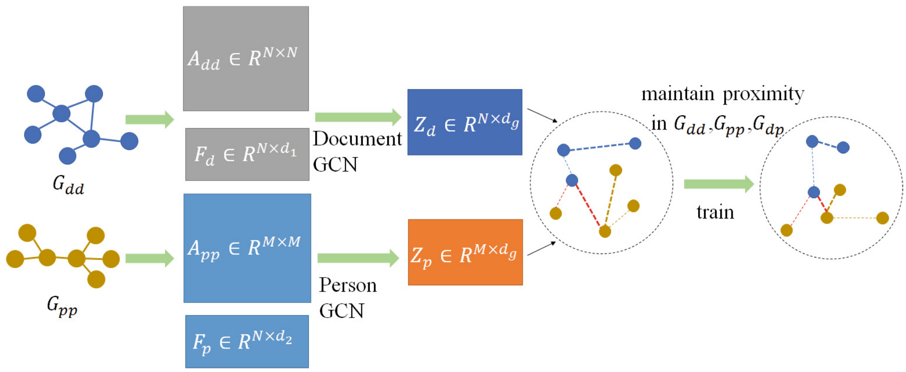

Fig. 1. The architecture of our proposed embedding model, G_{pp}, G_{dd}, G_{dp} is Person-Person Graph, Document-Document Graph and Document-Person Graph based on authorship and documents' similarity. A_{dd}, A_{pp} is adjacent matrix corresponding with G_{pp} and G_{dd}, and F_d, F_p is feature matrix. The documents and persons embedding (denoted as Z_d, Z_p) is acquired by Document-GCN and Person-GCN respectively, then keep close linked (bold dotted line) entities in G_{pp}, G_{dd} and G_{dp} proximate to each other in embedding space with minimizing triplet loss.

4.1 Graph Embedding

As mentioned in Sect. 3, there are two types of node in three types of graph. We use two personalized GCNs acquire embedding matrix for nodes in Person-Person Graph and Document-Document Graph respectively, due to its effectiveness for modeling networked data.

The goal of GCN is to learn a function of a graph $G = \{N, E\}$ which takes nodes feature matrix F and graph adjacent matrix A as input and produces a node-level output embedding matrix Z which incorporates both structural information and nodes feature. The function can be describe as follow:

$$H^{(l)} = g(H^{(l-1)}, A) = \sigma(\hat{A}H^{(l-1)}W^{(l)}) \tag{3}$$

where H^l is a feature matrix which is output of l^{th} graph convolutional layer, when it is first layer, $H^0 = F$. The output of final layer is Z, $W^{(l)}$ is a weight matrix for l^{th} neural network layer, $\sigma(\cdot)$ is an activation function and $\hat{A}$ is a symmetrically normalized adjacency matrix, $\hat{A} = D^{\frac{1}{2}}AD^{-\frac{1}{2}}$, D is the degree matrix of G.

Denote document embedding matrix as Z_d and person embedding matrix as Z_p, we formulize the two personalized GCNs as follow:

$$Z_d = \sigma(\hat{A}_{dd}\sigma(\hat{A}_{dd}F_dW_d^{(1)})W_d^{(2)}) \tag{4}$$

$$Z_p = \sigma(\hat{A}_{pp}\sigma(\hat{A}_{pp}F_pW_p^{(1)})W_p^{(2)}) \tag{5}$$

where $\hat{A}_{dd}$ and $\hat{A}_{pp}$ is symmetrically normalized adjacency matrix of G_{dd} and G_{pp} respectively, F_d and F_p is document and person nodes feature matrix. We call the two personalized GCN as Document-GCN and Person-GCN.

4.2 Objective Function

In intuition, scholars who collaborate more often are more likely to have similar research interests than those who do not work together or seldom coauthor a paper. This relation should be maintained in embedding space. Given a triplet consists of three person node A_i, A_j and A_k, the corresponding embedding learned from Person-GCN is z_{pi}, z_{pj} and z_{pk}, if A_i collaborates with A_j more frequently than $A_k (w_{ij} > w_{ik})$, the distance between z_{pi} and z_{pj} should be smaller than the distance between z_{pi} and z_{pk}. All triplets should satisfy

$$\|z_{pi} - z_{pj}\|_2 < \|z_{pi} - z_{pk}\|_2 \tag{6}$$

where $\|\cdot\|_2$ is the Euclidean norm. The loss function to make triplets meet this condition is:

$$\mathcal{L}_{pp}(A_i, A_j, A_k) = max\{\|z_{pi} - z_{pj}\|_2 - \|z_{pi} - z_{pk}\|_2, 0\} \tag{7}$$

Similar to the relation between person, the more common features shared two documents, the closer their embeddings are to each other. Specifically, the distance between documents D_i and D_j in embedding space is smaller than the distance between documents D_i and D_k if $w_{ij} = 1$ and $w_{ik} = 0$. For Document-Document graph, the loss function is:

$$\mathcal{L}_{dd}(D_i, D_j, D_k) = max\{\|z_{di} - z_{dj}\|_2 - \|z_{di} - z_{dk}\|_2, 0\} \tag{8}$$

where z_{di} is embedding of document D_i learned by Document-GCN.

As documents embedding and persons embedding acquired from the two personalized GCNs, the linkage between document and person in Document-Person could restrict personalized GCNs to map the two types of nodes into a same space that distance between different types of nodes could be measured. In intuition, the distance between a document i and its author j is smaller than the distance between document i and another person k. With $w_{ij} = 1$ and $w_{ik} = 0$, the loss function is

$$\mathcal{L}_{dp}(D_i, A_j, A_k) = max\{\|z_{di} - z_{pj}\|_2 - \|z_{di} - z_{pk}\|_2, 0\} \tag{9}$$

The three loss functions has same structure, they all have an anchor node i, a positive node j and a negative node k, the objective is to maximize the distance between anchor and negative node and minimize the distance between anchor and negative node. For each document and person as anchor node, we sample positive node according to the weight w to anchor node, the bigger the weight, the more likely it is to be selected. The negative node is chosen by their distance to anchor node, the smaller the distance, the more likely it is to be selected.

To preserve all constrains simultaneously, we propose a model which combines all loss functions derived from three different graphs and joint minimizes the following objective function:

$$\begin{gathered}\mathcal{L} = \sum_{A_i \in \mathcal{A}} \mathcal{L}_{pp}(A_i, A_j, A_k) + \sum_{D_i \in \mathcal{D}} \mathcal{L}_{dd}(D_i, D_j, D_k) + \sum_{D_i \in \mathcal{D}} \mathcal{L}_{dp}(D_i, A_j, A_k) \\ s.t.\ w_{ij} > w_{ik} \quad \forall G \in \{G_{dd}, G_{pp}, G_{dp}\}\end{gathered} \tag{10}$$

In training stage, we minimize $\mathcal{L}_{dd}$, $\mathcal{L}_{pp}$, $\mathcal{L}_{dp}$ and update their corresponding gradients successively. The complete learning algorithm is summarized in Algorithm 1.

Algorithm 1. Graph structure and attribute features based name disambiguation method

Require: name reference a associated $\mathcal{D}^a$, $\mathcal{A}^a$, cluster size K.
Ensure: K disjoint clusters.

```
Construct G_pp, G_dd, G_dp, acquire F_d, F_p.
for each epoch do:
    get documents and persons embedding by Document-GCN and Person-GCN.
    sample triplets from G_pp.
    minimize L_pp(A_i, A_j, A_k) and update parameters in Person-GCN with Adagrad.
    sample triplets from G_dd.
    minimize L_dd(D_i, D_j, D_k) and update parameters in Document-GCN with Adagrad.
    sample triplets from G_dp.
    minimize L_dd(D_i, A_j, A_k) and update parameters in both Document-GCN and Person-GCN with Adagrad.
end for
Given K, perform HAC to partition D^a into K disjoint clusters C^a with documents embedding as input.
return C^a
```

5 Experiments

In this section, we analyze the proposed model empirically on a challenging benchmark proposed by Aminer [29]. The benchmark consists of 70,258 documents from 12,798 authors. The document contains rich information such as title, author, keywords, published year and venue. We random select 20 name references from the dataset. There are 264 documents and 8 distinct persons for a name reference in average. For author's name "L.Song", there are 700 associated documents and 33 distinct real-life authors. This is a difficult disambiguation task.

5.1 Baseline Methods

To validate the effectiveness of our model, we compare it against three state of the art methods.

Aminer [29]**:** This is a two steps method, firstly train a model with a little mount data to map document feature into global embedding space in which documents associated with same author would be close to each other. And then a GCN based graph auto-encoder with global embedding as node feature is used to learn document representations. The objective is to minimize the reconstruction error between dot product of embedding and origin documents feature similarity based graph. Finally, the clustering result is generated by HAC.

Zhang [28]**:** This method constructs three types of graph based on coauthors and document similarity. A graph embedding is learned by minimizing the triplet loss which aims to make the distance between linked nodes is smaller than others, and then perform cluster algorithm. This method is similar to ours but it neglects nodes' attribute feature.

Louppe [18]**:** This method trains a function for measure distance between a document pair based on document feature, and then used a semi-supervised HAC to determine clusters.

5.2 Experimental Settings

In all experiments, we use Aminer proposed global embedding [29] as document feature, specifically, we sample 500 name references from Aminer dataset (as training data for Louppe's method [18] too), and then train a supervised model to learn document embedding with metric learning. Taking the document feature as input, the model's output is global embedding. We use one-hot embedding of author name as person feature due to author information is scarce in dataset. The IDF threshold to construct document-document graph is set as 32. For both document-GCN and person-GCN, the first GCN layer size is 64 and the second layer size is 128. Our model is trained with 0.01 learning rate and 1000 epochs. The parameters of baseline methods is set according to the origin paper or open source code. We run all the experiments on a 32 cores machine with 128G memory.

5.3 Effectiveness Evaluation

Table 1 shows the performance comparison of name disambiguation between our proposed model and other competing methods for all 20 name references. As commonly performed in name disambiguation research, we compare our model with baseline methods in pairwise precision, recall and F1 [16]. Each row is a name reference evaluated in our experiments, the columns (3, 4, 5) is various baseline methods, the last is the average of evaluate metrics of all 20 name references. For the accuracy of the experiment, we execute every method 5 times on each name reference.

As we observed, due to mainly modeling document similarity, AMiner's method and Louppe's method could distinguish real-world authors more precisely, for 17 names they are the best in precise. It's also worth noting that, although Zhang's method learns documents embedding with structural information only, it achieves the best for 5 name references' recall. Specifically, for "L.Song" and "J.Shao", it exceeds Zhang's and Louppe's methods for 15.9% to 30.9%, for average recall, the superiority is 10.4% and 9.9%. The significant improvement shows that the relation within authorship is helpful to gather together documents with same author as much as possible.

With combining structural information and documents attribute features, our method makes a better trade-off between precise and recall, performs the best for 9 name references in terms of recall and 10 names in terms of F1. Shown as Fig. 2, for average, the recall and F1 of our method is the best.

Table 1. Comparison of precision, recall and F1 between our proposed method and other baseline methods for name disambiguation task on 20 name references.

Name	Our method			AMiner			Zhang			Louppe		
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
M. Chen	98.8	99.2	99.0	**98.9**	**99.9**	**99.4**	88.6	98.8	93.4	94.1	97.1	95.5
W. Zhang	45.3	61.6	52.1	**54.2**	50.2	52.1	44.5	**84.3**	**58.3**	47.2	67.9	55.7
J. Du	70.5	72.8	71.6	68.4	67.2	67.7	15.3	73.8	25.3	**81.4**	**75.4**	**78.3**
H.B. Li	56.0	**86.1**	67.7	63.4	75.2	68.3	13.7	61.6	22.4	**75.3**	66.9	**70.8**
Y.Y. Li	43.9	**93.3**	59.7	74.1	65.5	69.5	25.3	51.3	33.9	**72.7**	66.8	**69.6**
X. Zhang	84.0	81.9	**83.0**	**88.3**	64.0	74.2	60.0	58.0	59.0	62.9	**82.3**	71.3
J.M. Fu	**97.3**	99.4	**98.3**	**97.3**	50.6	66.6	**97.3**	98.9	98.1	94.2	**100**	97.0
J.G. He	76.3	**90.1**	82.6	**92.2**	82.7	**87.2**	36.8	89.5	52.1	82.4	88.8	85.4
B. Hong	79.5	82.9	**81.1**	76.2	72.9	74.5	17.2	**85.0**	28.6	**83.4**	71.6	77.1
W. Yang	81.5	97.5	87.5	**96.5**	**98.2**	**97.4**	48.5	95.7	64.3	91.2	76.5	83.2
R. Lu	69.7	**83.5**	75.8	77.7	83.0	**80.2**	11.6	80.7	20.3	**86.4**	65.5	74.5
J. Feng	91.2	**95.8**	**93.4**	**92.0**	90.8	91.2	13.9	88.0	23.9	76.2	82.9	79.4
X. Qin	91.9	**95.2**	**93.5**	**92.1**	94.6	93.3	51.9	93.8	66.8	81.4	94.5	87.5
S. Wang	**57.7**	**92.8**	**71.1**	56.8	64.4	60.3	20.2	84.9	32.7	56.0	85.4	67.6
L. Song	61.0	86.6	**71.6**	62.0	75.0	67.8	24.1	**93.1**	38.2	**69.2**	71.4	70.3
F. Teng	94.0	99.1	**96.5**	**99.5**	87.6	93.2	94.0	98.2	96.0	87.9	**100**	93.6
S. Song	81.4	91.3	**86.1**	**92.4**	78.4	84.8	28.5	**93.0**	43.6	88.0	74.1	80.5
K. Xu	**91.4**	**98.6**	**94.9**	91.3	75.7	81.1	71.3	94.4	81.2	82.9	97.1	89.4
J. Shao	65.9	90.4	74.8	**90.7**	63.2	74.5	51.5	**94.1**	66.6	88.7	78.2	**83.1**
J. Lu	77.4	**98.8**	86.7	**95.7**	66.5	78.5	70.8	96.3	81.6	83.4	89.7	**87.5**
Avg	75.7	**89.8**	**81.3**	**83.0**	75.3	78.1	44.3	85.7	54.3	76.6	75.8	74.7

5.4 Component Contribution Analysis

Our proposed model consists of three types of graphs. For each graph we design a triplet loss function for maintaining graph proximity in embedding space. In this section, we analysis the contribution of each of the three components for the name disambiguation task by incrementally adding the components in the embedding model. We first add Document-Person graph, followed by Document-Document graph, and Person-Person graph. Specifically, we evaluate $\mathcal{L}_{dd}$, $\mathcal{L}_{dd} + \mathcal{L}_{dp}$, $\mathcal{L}_{dd} + \mathcal{L}_{pp} + \mathcal{L}_{dp}$ three types of loss function combinations.

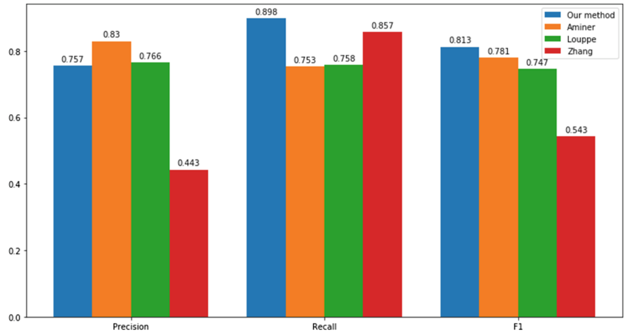

Fig. 2. Comparison of average pairwise precision, recall and F1

Table 2 shows the name disambiguation performance in terms of pairwise precision, recall and F1 using our proposed embedding model with different component combinations. As we see, after adding each component, we observe improvement for recall and decline for precise while F1 is rising, that means our model could make a better trade-off with more structural information.

Table 2. Component contribution analysis

Object function	Precision	Recall	F1
$\mathcal{L}_{dd}$	**77.52**	83.20	80.25
$\mathcal{L}_{dd} + \mathcal{L}_{dp}$	76.44	86.21	81.04
$\mathcal{L}_{dd} + \mathcal{L}_{pp} + \mathcal{L}_{dp}$	75.73	**89.84**	**81.34**

6 Conclusion

In this paper, we have proposed a novel representation learning based solution to address the name disambiguation problem. Our proposed representation learning model embed both document and person entities into a same space with two personalized GCNs and maintain proximity of close linked entities in embedding space by minimizing the careful designed objective function. Benefited from structural information and attribute features, the learned embedding could be effectively utilized for name disambiguation. Experimental results shows our proposed method makes a better trade-off between precise and recall, it is competitive with many of the existing state-of-the-arts for name disambiguation.

Learning embedding with same epochs for different graphs (different name reference) is likely to overfit, how to avoid and achieve a better performance could be future work.

Acknowledgements. This work is supported in part by National Key R&D Program of China 2017YFB0803305, NSFC 61772151&61872022, Beijing Advanced Innovation Center for Big Data and Brain Computing.

References

1. Bruna, J., Zaremba, W., Szlam, A., LeCun, Y.: Spectral networks and locally connected networks on graphs. In: 2nd International Conference on Learning Representations, ICLR 2014, Conference Track Proceedings, Banff, AB, Canada, 14–16 April 2014 (2014)
2. Cen, L., Dragut, E.C., Si, L., Ouzzani, M.: Author disambiguation by hierarchical agglomerative clustering with adaptive stopping criterion. In: The 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2013, Dublin, Ireland, July 28–August 01 2013, pp. 741–744 (2013)
3. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, Barcelona, Spain, 5–10 December 2016, pp. 3837–3845 (2016)
4. Elmacioglu, E., Tan, Y.F., Yan, S., Kan, M., Lee, D.: PSNUS: web people name disambiguation by simple clustering with rich features. In: Proceedings of the 4th International Workshop on Semantic Evaluations, SemEval@ACL 2007, Prague, Czech Republic, 23–24 June 2007, pp. 268–271 (2007)
5. Fan, X., Wang, J., Pu, X., Zhou, L., Lv, B.: On graph-based name disambiguation. J. Data Inf. Qual. **2**(2), 10:1–10:23 (2011)
6. Ferreira, A.A., Gonçalves, M.A., Laender, A.H.F.: A brief survey of automatic methods for author name disambiguation. ACM SIGMOD Rec. **41**(2), 15–26 (2012)
7. Fout, A., Byrd, J., Shariat, B., Ben-Hur, A.: Protein interface prediction using graph convolutional networks. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017, pp. 6533–6542 (2017)
8. Froud, H., Lachkar, A.: Agglomerative hierarchical clustering techniques for Arabic documents. In: Nagamalai, D., Kumar, A., Annamalai, A. (eds.) Advances in Computational Science, Engineering and Information Technology, vol. 225, pp. 225–267. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-319-00951-3_25
9. Han, H., Giles, C.L., Zha, H., Li, C., Tsioutsiouliklis, K.: Two supervised learning approaches for name disambiguation in author citations. In: ACM/IEEE Joint Conference on Digital Libraries, JCDL 2004, Tucson, AZ, USA, 7–11 June 2004, Proceedings, pp. 296–305 (2004)
10. Hermansson, L., Kerola, T., Johansson, F., Jethava, V., Dubhashi, D.P.: Entity disambiguation in anonymized graphs using graph kernels. In: 22nd ACM International Conference on Information and Knowledge Management, CIKM 2013, San Francisco, CA, USA, October 27–November 1 2013, pp. 1037–1046 (2013)

11. Hoffart, J., et al.: Robust disambiguation of named entities in text. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27–31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A Meeting of SIGDAT, a Special Interest Group of the ACL, pp. 782–792 (2011)
12. Huang, J., Ertekin, S., Giles, C.L.: Efficient name disambiguation for large-scale databases. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) PKDD 2006. LNCS (LNAI), vol. 4213, pp. 536–544. Springer, Heidelberg (2006). https://doi.org/10.1007/11871637_53
13. Hussain, I., Asghar, S.: A survey of author name disambiguation techniques: 2010–2016. Knowl. Eng. Rev. **32**, e22 (2017)
14. Kipf, T.N., Welling, M.: Variational graph auto-encoders. CoRR (2016)
15. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017, Conference Track Proceedings (2017)
16. Levin, M., Krawczyk, S., Bethard, S., Dan, J.: Citation-based bootstrapping for large-scale author disambiguation. J. Am. Soc. Inf. Sci. Technol. **63**(5), 1030–1047 (2012)
17. Li, J., et al.: Graph CNNs for urban traffic passenger flows prediction. In: SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI 2018, Guangzhou, China, 8–12 October 2018 (2018)
18. Louppe, G., Al-Natsheh, H.T., Susik, M., Maguire, E.J.: Ethnicity sensitive author disambiguation using semi-supervised learning. In: Ngonga Ngomo, A.-C., Křemen, P. (eds.) KESW 2016. CCIS, vol. 649, pp. 272–287. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45880-9_21
19. Nguyen, T.H., Grishman, R.: Graph convolutional networks with argument-aware pooling for event detection. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-2018), the 30th Innovative Applications of Artificial Intelligence (IAAI-2018), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-2018), New Orleans, Louisiana, USA, 2–7 February 2018, pp. 5900–5907 (2018)
20. On, B.W.: Social network analysis on name disambiguation and more. In: International Conference on Convergence & Hybrid Information Technology (2008)
21. Peng, H., et al.: Fine-grained event categorization with heterogeneous graph convolutional networks. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, 10–16 August 2019 (2019)
22. Peng, H., et al.: Hierarchical taxonomy-aware and attentional graph capsule RCNNs for large-scale multi-label text classification. CoRR (2019)
23. Peng, H., et al.: Large-scale hierarchical text classification with recursively regularized deep graph-CNN. In: Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, 23–27 April 2018 (2018)
24. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. ESWC 2018. LNCS, vol. 10843, pp. 593–607. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_38
25. Tang, J., Fong, A.C.M., Wang, B., Zhang, J.: A unified probabilistic framework for name disambiguation in digital library. IEEE Trans. Knowl. Data Eng. **24**(6), 975–987 (2012)

26. Tran, H.N., Huynh, T., Do, T.: Author name disambiguation by using deep neural network. In: Nguyen, N.T., Attachoo, B., Trawiński, B., Somboonviwat, K. (eds.) ACIIDS 2014. LNCS (LNAI), vol. 8397, pp. 123–132. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-05476-6_13
27. Yoshida, M., Ikeda, M., Ono, S., Sato, I., Nakagawa, H.: Person name disambiguation by bootstrapping. In: Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, Geneva, Switzerland, 19–23 July 2010, pp. 10–17 (2010)
28. Zhang, B., Hasan, M.A.: Name disambiguation in anonymized graphs using network embedding. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, 06–10 November 2017, pp. 1239–1248 (2017)
29. Zhang, Y., Zhang, F., Yao, P., Tang, J.: Name disambiguation in AMiner: clustering, maintenance, and human in the loop. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, 19–23 August 2018, pp. 1002–1011 (2018)