Shanghai Jiao Tong University
EE447 – Mobile Internet
2nd Semester, 2019-2020

Author Name Disambiguation via Network Embedding

Report for EE447 Final Project

Mingjie Li (李明杰) 517030910344

June 8, 2020

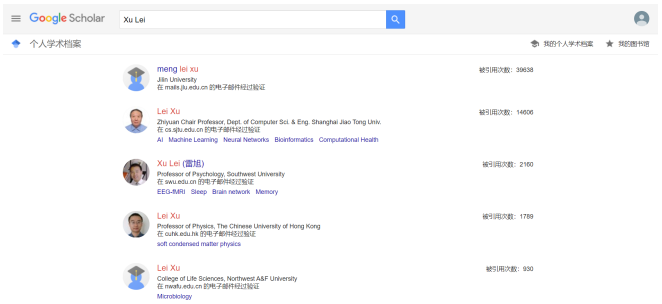
Abstract

As the final project of *EE447: Mobile Internet*, I have first reviewed the existing network-embedding-based learning frameworks of author name disambiguation. I have also implemented one classical algorithm of them. After that I analyzed the probable shortcomings of this algorithm and proposed a new framework. Experimental results showed that the proposed algorithm outperforms the classical one and is also much easier to generalize to other problems. In this course project, I have learned a lot, from problem formulation and model design and analysis.

1 Introduction

Name queries are usually treated by search engines as normal keyword searches without attention to the ambiguity of particular names. However, due to issues like name abbreviations, identical names, etc., a person cannot be uniquely identified by his or her name, especially the case of Chinese names. This is a very important issue in the academic area. For example, an online search query for "Xu Lei" my retrieve nearly 100 researches of this name, as is shown in Figure 1(a). This propose great challenge for an academic searching engine to provide information of an exact scholar accurately. However, in many cases, the Google scholar page of a certain researcher may contain papers of another scholar with the same name. Figure 1(b) shows an example. On the Google scholar page of Prof. Bo Yuan, who is a professor at the computer science department of Shanghai Jiao Tong University, there exists several irrelevant papers. For instance, *Investigation into the Anodic Dissolutino Processes of Copper in Neutral and Acidic Sulfate Solutions with the In-line Digital Holography* published on Electrochemistry is obviously not a paper in Prof. Bo Yuan's research area. Another example is *Rib spalling mechanism and prevention technology for soft seam large mining height face* published on Safety in Coal Mines.

To address this problem, network embedding is a suitable approach. In recent years, there are many works of network embeddings. [1] summarized traditional methods such as DeepWalk,



(a) The searching result for name "Xu Lei".

(b) The google scholar page of Bo Yuan.

Figure 1: Two examples of the challenges existing in name disambiguation.

LINE, PTE and Node2Vec in a unified form via matrix factorization. [2] proposed a unified framework for community detection and network representation learning. [3] proposed a more expressive hypercomplex representations in the network. [4] proposed to conduct spectral clustering in heterogeneous information network. There are many network embedding algorithms.

However, when it comes to the AND problem, researchers usually come up with specifically formulated algorithms. Existing approaches that address the issue of name disambiguation generally fall into two categories: supervised learning and unsupervised learning methods. [5] proposed an unsupervised learning method based on network embedding as a pioneering work. Inspired by [5], in paper [6] proposed an extended but un-anonymized version of the same method. As for supervised learning methods, [7] proposed a framework but this required human annotators. Recently, [8] proposed a name disambiguation framework based on adversarial representation learning on heterogeneous information network.

In this report, I will focus on the unsupervised network embedding framework. I will review and analyze the shortcomings of the framework proposed in [5] and propose a new framework easy to generalize. The rest of this report will be organized as follows. In Section 2, I will give a mathematical formulation of the author name disambiguation problem (a.k.a. AND problem). Section 3 reviews one existing method [5] and I have done a re-implementation. In Section 4, I analyzed the shortcomings of the framework in Section 3 and proposed a new learning pipeline. Section 5 presents some experimental results. At last, I summarized this report in Section 6.

2 Problem Formulation

In this section, I will give a mathematical formulation and the evaluation metric used in [5] of the AND problem, as follows.

For a given name reference a, we suppose there are K person entities[1] with name a, *i.e.* there are K people with the same name a. We denote $\mathcal{E}^a = \{E_1^a, E_2^a, \ldots, E_K^a\}$ as the *author entity set*. Moreover, for the documents, we denote the *document entity set* $\mathcal{D}^a = \{D_1^a, D_2^a, \ldots, D_N^a\}$ as a set of N different documents (papers), in which a is one of the authors. For each document D_i^a, we

[1] A person entity uniquely represent one real-life person. In this AND problem, we suppose K is given.

are also given its text features T_i^a and relation features R_i^a. Our aim is to partition the set $\mathcal{D}$ into K disjoint sets, such that each set contains papers belonging to exactly one author entity. In mathematical form, let I be a mapping from the document entity set to the author entity set, *i.e.* $I : \mathcal{D} \to \mathcal{E}$, to denote the ground truth partition function with $I(D_i^a) = E_j^a$. The mapping function we learned is $\Phi : \mathcal{D} \to \mathcal{E}$. Our goal is to let $\Phi \sim I$.

Then the problem occurs about how to measure the accuracy of $\Phi \sim I$. In this project, I adopted the evaluation metric in [5]. The definition is given below.

Suppose the ground partition is $C_1^\star, C_2^\star, \ldots, C_K^\star$, and the predicted partition is $\hat{C}_1, \hat{C}_2, \ldots, \hat{C}_K$. To match the partition, we define a matching $C_j = \arg\max_{\hat{C}_i} |\hat{C}_i \cap C_j^\star|$. Then we have the following metrics.

$$\text{precision}(C_i) = \frac{|\hat{C}_i \cap C_i|}{\sum_j |\hat{C}_i \cap C_j|} \tag{1}$$

$$\text{recall}(C_i) = \frac{|\hat{C}_i \cap C_i|}{\sum_j |\hat{C}_j \cap C_i|} \tag{2}$$

$$\text{F1}(C_i) = \frac{2 \times \text{precision}(C_i)^2}{\text{precision}(C_i) + \text{recall}(C_i)} \tag{3}$$

We use the macro-F1 score as the final evaluation metric, as follows. The larger the macro-F1 score means better accuracy.

$$\text{macroF1}(C) = \frac{1}{K} \sum_{i=1}^{K} \text{F1}(C_i) \tag{4}$$

3 Joint Network Embedding

This section is a review of the framework proposed in [5]. In the experiment part, I have re-implemented this method. I have also analyzed the shortcomings of this algorithm and developed a new learning pipeline in the next section. The method in [5] proposed to joint learn the embedding of the following three networks.

Definition 1: Person-person Network. For a given name reference x, the person-person network, denoted as $G_{pp} = (\mathcal{A}^x, E_{pp})$, captures collaboration between a pair of persons within the collection of documents associated with x. $\mathcal{A}^x$ is the collaborator set, and $e_{ij} \in E_{pp}$ represents the edge between the persons a_i and a_j, who collaborated in at least one document. The weight w_{ij} of the edge e_{ij} is defined as the number of distinct documents in which a_i and a_j have collaborated.

The intuitive understanding of the person-person network is that it can form several "clusters" that the target person has collaborated with. However, it does not account for the fact that the target person may have collaborated with two or more clusters. Therefore, the person-document network and the document-document network cover for this shortcoming.

Definition 2: Person-document Network. Person-document network, represented as $G_{pd} = (\mathcal{A} \cup \mathcal{D}, E_{pd})$, is a bipartite network where $\mathcal{D}$ is the set of collaborators of a over all the documents in $\mathcal{D}$. E_{pd} is the set of edges between persons and documents. The edge weight w_{ij} between a

person node a_i and document d_j is simply defined as the number of times a_i appears in document d_j. For a bibliographic dataset, $w_{ij} = 1$.

Definition 3: Document-document Network. Document-document network, represented as $G_{dd} = (\mathcal{D}, E_{dd})$, where each vertex $d_i \in \mathcal{D}$ is a document. The weight of edge w_{ij} is defined as the similarity between d_i and d_j. The similarity is defined as the two-hop collaborators of the two documents. For more details please see [5].

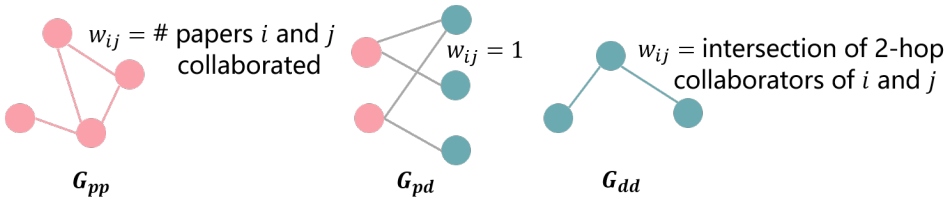


Figure 2: An illustration of the three networks.

Figure2 shows an illustration of the three networks mentioned above. To jointly learn the embeddings for the nodes in the three networks, the learning objective is straight-forward: to maximize the gap between positive edges and negative edges.

Suppose the learned embeddings for the documents form a matrix $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_N] \in \mathbb{R}^{k \times N}$ and the embeddings for the authors form a matrix $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_N] \in \mathbb{R}^{k \times |\mathcal{A}|}$. The main intuition of this network embedding model is that neighboring nodes in a graph should have more similar vector representation in the embedding space than non-neighboring nodes. Then in network G_{pp}, we have loss: (For illustration please see Figure 3.)

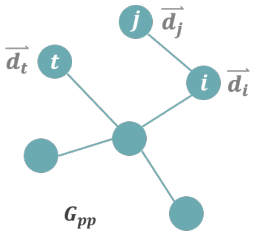


Figure 3: An illustration of the loss.

$$\mathcal{L}_{pp} = -\sum_{(i,j)\in P_G,(i,t)\in N_G} \Big[\log \sigma(\mathbf{d}_i \cdot \mathbf{d}_j - \mathbf{d}_i \cdot \mathbf{d}_t)\Big] \tag{5}$$

The set P_G (N_G) represents the sampled positive (negative) pair in graph G. *I.e.* for $(i,j) \in E_G$ ($(i,j) \notin E_G$), we have $(i,j) \in P_G$ ($(i,t) \notin P_G$). In each step, we sample the positive pair with the probability proportional to the edge weight for the model. On the other hand, for sampling of negative, we utilize uniform sampling technique.

The loss function for the other two networks have the same pattern. The total loss is as follows, where $\|\mathbf{D}\|_F^2 + \|\mathbf{A}\|_F^2$ is the regularization term.

$$\mathcal{L} = \mathcal{L}_{pp} + \mathcal{L}_{pd} + \mathcal{L}_{dd} + \lambda(\|\mathbf{D}\|_F^2 + \|\mathbf{A}\|_F^2) \tag{6}$$

Though straight forward as this algorithm is, it does not consider textual information, such as the title. That is a reason why the performance is not so good. More details will be analyzed in Section 5.

4 Proposed Framework

In this section, I will propose a new learning framework for AND problem. The new framework is quite simple yet effective to solve many problems.

First, let us consider the preliminary embedding for these papers based on the content information. However, there is no abstract in this dataset, so we can only use the titles as our content information in the experiment. Word2vec[9] provides us with a strong tool to embed the latent semantics of words. The problem is how to use the word vectors to form latent representations for these titles. Due to the unsupervised learning pattern, we need to find an objective to train the embeddings for these titles. A possible learning framework is shown in Figure 4.

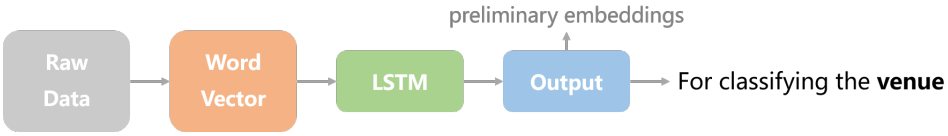


Figure 4: The process of obtaining the preliminary paper embeddings.

First, we delete the stop words in the titles. Then we feed these words to the Word2Vec module and obtain the embeddings for words. Further, since the sequential property of the data, we consider feeding the sequence to an LSTM. To enable the training, we designed the objective is that, from the embeddings we are able to classify its venue. The reason is that different venues indicate different research directions, so we believe the same venue should have similar embeddings. Besides, the preliminary embeddings also preserves the information of the input, *i.e.* the information in the word vectors.

However, the ability of this embedding to distinguish documents with different authors is limited. Thus, we seek to refine the representations of the papers using the relation information. Before that, let us get familiar with two important concepts: Heterogeneous Information Network and meta-path in an HIN. Also, I will demonstrate how these two graphs are modeled in the real problem.

Definition 4: Heterogeneous Information Network (HIN). Let $\mathcal{T} = \{T_1, T_2, \ldots, T_m\}$ be a set of m object types and $\mathcal{X}_i$ be the set of objects of type T_i. An HIN is a network $G = (V, E)$ in which $V = \cup \mathcal{X}_i$. Each link in E represents a binary relation R_{ij} between two objects of different pattern in V.

In the problem setting, there are four kinds of nodes in the network, namely paper node, word node, venue node and collaborator node, as is shown in Figure 6. The embeddings of the paper nodes are what we aim to obtain. Paper nodes are connected to other kinds of nodes.

- A paper node is connected to a word node if and only if the word appears in the title of the paper.
- A paper node is connected to a venue node if and only if the paper is published in the corresponding venue.

- A paper node is connected to a collaborator node if and only if the author list of the paper contains this person.

Definition 5: Meta-path in an HIN. A meta-path is a path concerning a composite relation $R = R_1 \circ R_2 \circ \cdots \circ R_l$. If two objects x_u and x_v are related by the composite relation R. Then there is a path, denoted by $p_{x_u \rightsquigarrow x_v}$. Moreover, the sequence of links in $p_{x_u \rightsquigarrow x_v}$ matches the sequence of relations $R = R_1 \circ R_2 \circ \cdots \circ R_l$.

In the AND problem, we consider three kinds of basic meta-paths. These three basic meta-paths only consider first-order relationship. These three kinds of meta-paths are DVD (document-venue-document), DWD (document-word-document) and DCD (document-collaborator-document). The semantic meanings of each single meta-paths of these types are shown in Figure 5. Also, longer meta-paths will encode more semantic information, and higher order relationship. Consider the two vertice in the meta-path "DWDCD" in Figure 5.

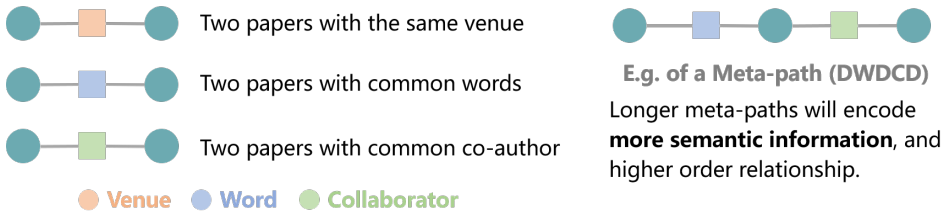


Figure 5: Examples of meta-paths in an HIN. Longer meta-paths will encode more semantic information, and higher order relationship.

Algorithm 1: Generate random walks in an HIN

Data: The HIN $G = (\mathcal{D} \cup \mathcal{V} \cup \mathcal{W} \cup \mathcal{C}, \mathcal{E})$, iteration times L, walks for every node N
Result: A set of generated random walks S

Initialize the heterogeneous information network G;
Initialize an empty path $\mathcal{P}$;
foreach *document node* $s \in G$ **do**
  for $n = 1$ **to** N **do**
    Initialize $\mathcal{P} \leftarrow \{s\}$;
    for $l = 1$ **to** L **do**
      Sample a meta-path of "DVD" $p_{s \rightsquigarrow d}$, $s \leftarrow d$;
      Sample a meta-path of "DWD" $p_{s \rightsquigarrow d}$, $s \leftarrow d$;
      Sample a meta-path of "DCD" $p_{s \rightsquigarrow d}$, $s \leftarrow d$;
    end
    Add $\mathcal{P}$ to S;
  end
end
return the set of generated paths S

Then, to get the embedding of the paper nodes, we further conducted random walk in the HIN. The method is simple. The random walk is just sampled by iteratively sample the three

kinds of basic meta-paths in the HIN. *I.e.* first sample "DVD", then "DWD" and then "DCD". The algorithm is shown in 1. There is a demo in Figure 6, where the length of the walk is 7.

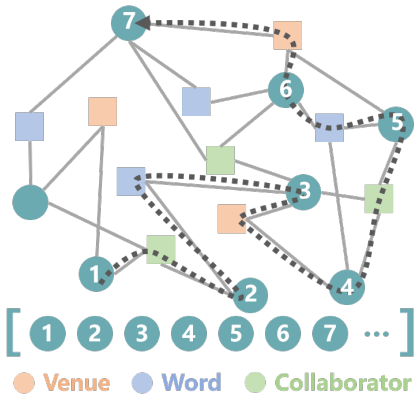


Figure 6: An example of a random-walk path in the HIN.

To enable the unsupervised training, we are inspired by the idea in Section 3. The neighboring nodes in the meta-path need to be much more similar than faraway nodes. So we only need to sample positive pairs and negative pairs in the network. The method is described as follows.

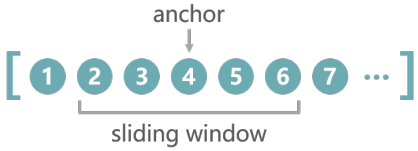


Figure 7: Demonstration of the triplet loss.

As is shown in Figure 7, we put a sliding window on the sampled path. The center of the sliding window is the anchor. For the anchor x in a sliding window, we define positive sampling and negative sampling as follows. Positive sampling: randomly sample m nodes $x_{1+}, \ldots, x_{m+}$ in the sliding window (excluding x it self); Negative sampling: randomly sample m nodes $x_{1-}, \ldots, x_{m-}$ in the network (excluding nodes in the sliding window). Then the objective triplet loss can be written as follows.

$$Loss(\mathbf{d}, \mathbf{d}_{m+}, \mathbf{d}_{m-}) = \sum_{m} Sim_{cos}(\mathbf{d}, \mathbf{d}_{m-}) - \sum_{m} Sim_{cos}(\mathbf{d}, \mathbf{d}_{m+}) \tag{7}$$

The size of the sliding window is set as 5 in experiments, to capture higher order relationship information. In particular if set as 3, only first order relationship is captured.

5 Experiment and Analysis

Dataset: We used the name disambiguation dataset on AMiner [10, 11]. This data set is used for studying name disambiguation in digital library. It contains 110 author names and their

disambiguation results (ground truth)[2].

First, I reimplemented the learning framework in Section 3, and analyzed the results. As is described in the original paper, I performed Hierarchical Agglomerative Clustering (HAC) with a given K on the learned embeddings for the papers. Since for every name reference, we did the embedding respectively, we sampled out a few name references. Table 1 shows the samples. The left half are authors with more papers, while the other half are authors with less papers.

Table 1: We sampled some typical authors from the dataset for visualization.

name	Bin Li	Lei Chen	Lei Wang	Bin Yu	Hao Wang	Jing Zhang	Yu Zhang	Yang Wang	Xiaoyan Li	Gang Luo	Z. Wang	X. Zhang	David Brown
#paper	181	196	308	105	178	231	235	195	33	47	47	62	61
#entity	60	40	112	17	48	85	72	55	6	9	38	40	25

Figure 8 shows the result. As we can see, the disambiguation performance of authors with more papers are significantly worse than that of authors with less papers. We have summarized the reasons and analyzed the pros and cons of this algorithm.

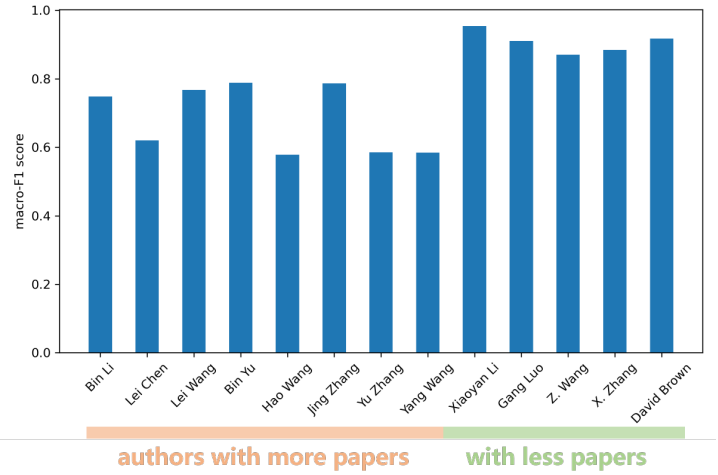


Figure 8: The disambiguation results of the algorithm in Section 3.

The algorithm in Section 3 uses three networks to deal with AND, which takes much given information into account. Also, joint learning of embeddings for different nodes is original. Besides, the objective is straight-forward and effective. However, there are many things to be improved. First is that in the AND problem, to learn the representation of co-authors and papers in the network is a little strange. Especially the reason why p and d need to be close in G_{pd}, this is really strange. Also, it does not take the content information into consideration (in this dataset, the content information is merely the title).

Therefore, we tested the performance of our proposed method, which not only directly learns the representations for the paper nodes, but also considers both the relationship information and textual information. As we can see in Figure 9, the blue bars are the original algorithm in Section 3, and the orange bars are the performance of our proposed algorithm. A higher bar indicates better

[2]https://www.aminer.org/disambiguation

performance. As we can see, the overall performance of the proposed algorithm is much more better than the original one, though the performance is slightly worse in some cases. Therefore, we believe the proposed algorithm is better than the original one.

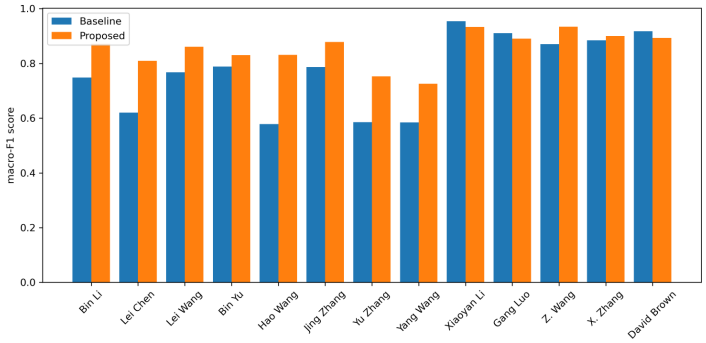


Figure 9: The disambiguation results of the proposed disambiguation pipeline.

Also need to note that, the proposed algorithm has a good ability of generalization. This pipeline can be generalized to more network embedding problems, including paper recommendation, social networks, etc...

6 Summary

As the final project of *EE447: Mobile Internet*, I have first reviewed the existing network-embedding-based learning frameworks of author name disambiguation. I have also implemented one classical algorithm of them. After that I analyzed the probable shortcomings of this algorithm and proposed a new framework. Experimental results showed that the proposed algorithm outperforms the classical one and is also much easier to generalize to other problems. In this course project, I have learned a lot, from problem formulation and model design and analysis.

However, during this project, I have also realized the limits of this problem. Firstly, The author-name-disambiguation problem is very specific, which needs to be modeled differently from traditional network embedding problems. Clearly, there exist much simpler, yet more effective ways to disambiguate authors with the same name, such as using e-mail addresses of authors extracted from papers (but this may lead to privacy problems ...) The most important issue may be the accuracy – if cannot ensure 100% correctness, the framework's application in the real world may be restricted. This is different from recommendation problems whose results only need to make sense to people.

Reference

[1] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 459–467, 2018.

[2] Cunchao Tu, Xiangkai Zeng, Hao Wang, Zhengyan Zhang, Zhiyuan Liu, Maosong Sun, Bo Zhang, and Leyu Lin. A unified framework for community detection and network representation learning. *IEEE Transactions on Knowledge and Data Engineering*, 31(6):1051–1065, 2018.

[3] Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. Quaternion knowledge graph embeddings. In *Advances in Neural Information Processing Systems*, pages 2731–2741, 2019.

[4] Xiang Li, Ben Kao, Zhaochun Ren, and Dawei Yin. Spectral clustering in heterogeneous information networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4221–4228, 2019.

[5] Baichuan Zhang and Mohammad Al Hasan. Name disambiguation in anonymized graphs using network embedding. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1239–1248, 2017.

[6] Jun Xu, Siqi Shen, Dongsheng Li, and Yongquan Fu. A network-embedding based method for author disambiguation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1735–1738, 2018.

[7] Yutao Zhang, Fanjin Zhang, Peiran Yao, and Jie Tang. 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*, pages 1002–1011, 2018.

[8] Haiwen Wang, Ruijie Wan, Chuan Wen, Shuhao Li, Yuting Jia, Weinan Zhang, and Xinbing Wang. Author name disambiguation on heterogeneous information network with adversarial representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 238–245, 2020.

[9] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[10] Jie Tang, Alvis C.M. Fong, Bo Wang, and Jing Zhang. A unified probabilistic framework for name disambiguation in digital library. *IEEE Transactions on Knowledge and Data Engineering*, 24(6), 2012.

[11] Xuezhi Wang, Jie Tang, Hong Cheng, and Philip S. Yu. Adana: Active name disambiguation. In *ICDM'11*, pages 794–803, 2011.