Name Disambiguation in Anonymized Graphs using Network Embedding*

Baichuan Zhang
Purdue University
West Lafayette, IN, USA
zhan1910@purdue.edu

Mohammad Al Hasan
Indiana University Purdue University Indianapolis
Indianapolis, IN, USA
alhasan@cs.iupui.edu

ABSTRACT

In real-world, our DNA is unique but many people share names. This phenomenon often causes erroneous aggregation of documents of multiple persons who are namesake of one another. Such mistakes deteriorate the performance of document retrieval, web search, and more seriously, cause improper attribution of credit or blame in digital forensic. To resolve this issue, the name disambiguation task is designed which aims to partition the documents associated with a name reference such that each partition contains documents pertaining to a unique real-life person. Existing solutions to this task substantially rely on feature engineering, such as biographical feature extraction, or construction of auxiliary features from Wikipedia. However, for many scenarios, such features may be costly to obtain or unavailable due to the risk of privacy violation. In this work, we propose a novel name disambiguation method. Our proposed method is non-intrusive of privacy because instead of using attributes pertaining to a real-life person, our method leverages only relational data in the form of anonymized graphs. In the methodological aspect, the proposed method uses a novel representation learning model to embed each document in a low dimensional vector space where name disambiguation can be solved by a hierarchical agglomerative clustering algorithm. Our experimental results demonstrate that the proposed method is significantly better than the existing name disambiguation methods working in a similar setting.

CCS CONCEPTS

•Information systems →Clustering; Information retrieval; Document representation;

KEYWORDS

Name Disambiguation;Neural Network Embedding;Clustering

ACM Reference format:
Baichuan Zhang and Mohammad Al Hasan. 2017. Name Disambiguation in Anonymized Graphs using Network Embedding[1]. In *Proceedings of CIKM'17 , Singapore, Singapore, November 6–10, 2017,* 11 pages.

[1]This research is sponsored by Mohammad Al Hasan's NSF CAREER Award (IIS-1149851) and also, by a research grant from CareerBuilder.

CIKM'17 , Singapore, Singapore

978-1-4503-4918-5/17/11…$15.00
DOI: 10.1145/3132847.3132873

DOI: 10.1145/3132847.3132873

1 INTRODUCTION

Name disambiguation [3, 10, 30, 32, 33] is an important problem, which has numerous applications in information retrieval, counterterrorism, and bibliographic data analysis. In information retrieval, name disambiguation is critical for sanitizing search results of ambiguous queries. For example, an online search query for "Michael Jordan" may retrieve pages of former US basketball player, the pages of UC Berkeley machine learning professor, and the pages of other persons having that name, and name disambiguation is necessary to split those pages into homogeneous groups. In counterterrorism, such an exercise is essential before inserting a person's profile in a law enforcement database; failing to do so may cause severe trouble to many innocent persons who are namesakes of a potential criminal. Evidently, name disambiguation is particularly important in the fields of bibliometrics and library science. This is due to the fact that many distinct authors share the same name reference as the first name of an author is typically written in abbreviated form in the citation of many scientific articles. Thus, bibliographic servers that maintain such data may mistakenly aggregate the articles from multiple scholars (sharing the same name) into a unique profile in some digital repositories. For an example, the Google scholar profile associated with the name "Yang Chen" (GS) [2] is verified as the profile page of a Computer Graphics PhD candidate at Purdue University, but based on our labeling, more than 20 distinct persons' publications are mixed under that profile mistakenly. Such mistakes in library science over- or underestimate a researcher's citation related impact metrics.

Due to its importance, the name disambiguation task has attracted substantial attention from information retrieval and data mining communities. However, the majority of existing solutions [1, 3, 12, 15] for this task use biographical features such as name, address, institutional affiliation, email address, and homepage. Also, contextual features such as collaborator, community affiliation, and external data source such as Wikipedia are used in some works [13, 15]. Using biographical features is acceptable for disambiguation of authors in bibliometrics domain, but in many scenarios, for example in the national security related applications, biographical features are hard to obtain, or they may even be illegal to obtain unless a security analyst has the appropriate level of security clearance. Besides, in real-world social networks (e.g., Twitter, Facebook, and LinkedIn), some users may choose a strict privacy setting that restricts the visibility of their profile information and posts. For such privacy-preserving scenarios, many existing name

[2]https://scholar.google.com/citations?user=gl26ACAAAAAJ&hl=en

This is the author's manuscript of the article published in final edited form as:

Zhang, B., & Al Hasan, M. (2017). Name Disambiguation in Anonymized Graphs Using Network Embedding. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (pp. 1239–1248). New York, NY, USA: ACM. https://doi.org/10.1145/3132847.3132873

disambiguation techniques [10, 12, 15, 22, 27], which compute document similarity using biographical attributes are not applicable.

In recent years, a few works have emerged where name disambiguation task in privacy-preserving setting has been considered [14, 32]. These works use relational data in the form of an anonymized person-person collaboration graph, and solve name disambiguation by using graph topological features. Thus they preserve the privacy of a user. Authors of [14] use graphlet kernels based classification model and the authors of [32] use Markov clustering based unsupervised approach. However, both of these works only consider a binary classification task, predicting whether a given person-node in the graph is ambiguous or non-ambiguous. This is far from a traditional name disambiguation task which partitions the records pertaining to a given name reference into different groups, each belonging to a unique person. Another limitation of the existing works is that they only utilize the person-person collaboration network, which does not generally yield a good disambiguation performance. There are other information, such as person-document association information and document-document similarity information, which can also be exploited for obtaining improved name disambiguation, yet preserving the user's privacy.

In this work, we solve the name disambiguation task by using only relational information. For a given name reference, our proposed method pre-processes the input data as three graphs: person-person graph representing collaboration between a pair of persons, person-document graph representing association of a person with a document and document-document similarity graph. These graphs are appropriately anonymized, as such, the vertices of these graphs are represented by a unique pseudo-random identifier. Nodal features (such as, biographical information of a person-node, or keywords of a document-node) of any of the above three graphs are not used, which makes the proposed method privacy-preserving.

In the graph representation, the name disambiguation task becomes a graph clustering task of the document-document graph, with the objective that each cluster contains documents pertained to a unique real-life person. A traditional method to cluster a homogeneous network cannot facilitate information exchange among the three graphs, so we propose a novel representation learning model, which embeds the vertices of these graphs into a shared low dimensional latent space by using a joint objective function. The objective function of our representation learning task utilizes pairwise similarity ranking which is different from the typical objective functions used in the existing document embedding methods, such as LINE [24] and PTE [23]; the latter ones are based on K-L divergence between empirical similarity distribution and embedding similarity distribution. K-L divergence works over the entire distribution vector and it works well for document labeling or topic modeling, but not so for clustering. On the other hand, our objective function is better suited for a downstream clustering task because it directly optimizes the pairwise distance between similar and dissimilar documents, thus making the document vectors disambiguation-aware in the embedded space, as such, a traditional hierarchical clustering of the vectors in the embedded space generates excellent name disambiguation performance. Experimental comparison with several state-of-the-art name disambiguation methods—both traditional and network embedding-based—show that the proposed method is significantly better than the existing methods on multiple real-life name disambiguation datasets.

The key contributions of this work are summarized as below:

(1) We solve the name disambiguation task by using only linked data from network topological information. The work is motivated by the growing demand for big data analysis without violating the user privacy in security sensitive domains.
(2) We propose a network embedding based solution that leverages linked structures of a variety of anonymized networks in order to represent each document into a low-dimensional vector space for solving the name disambiguation task. To the best of our knowledge, our work is the first one to adopt a representation learning framework for name disambiguation in anonymized graphs.
(3) For representation learning, we present a novel pairwise ranking based objective, which is particularly suitable for solving the name disambiguation task by clustering.
(4) We use two real-life bibliographic datasets for evaluating the disambiguation performance of our solution. The results demonstrate the superiority of our proposed method over the state-of-the-art methodologies for name disambiguation in privacy-preserving setup.

2 RELATED WORK

There exist a large number of works on name disambiguation [3, 10]. In terms of methodologies, existing works have considered supervised [1, 10], unsupervised [3, 11], and probabilistic relational models [21, 22, 33]. In the supervised setting, Han et al. [10] proposed supervised name disambiguation methodologies by utilizing Naive Bayes and SVM. In these works, a distinct real-life entity can be considered as a class, and the objective is to classify each record to one of the classes. For the unsupervised name disambiguation, the records are partitioned into several clusters with the goal of obtaining a partition where each cluster contains records from a unique entity. For example, Han et al.[11] used K-way spectral clustering for name disambiguation in bibliographical data. Recently, probabilistic relational models, especially graphical models have also been considered for the name disambiguation task. For instance, [22] proposed to use Markov Random Fields to address name disambiguation in a unified probabilistic framework.

Most existing solutions to the name disambiguation task use either biographical attributes, or auxiliary features that are collected from external sources. However, the attempt of extracting biographical or external data sustains the risk of privacy violation. To address this issue, a few works [14, 17, 20, 32] have considered name disambiguation using anonymized graphs without leveraging the node attributes. The central idea of this type of works is to exploit graph topological features to solve the name disambiguation problem without intruding user privacy through the collection of bibliographical attributes. For example, authors in [14] characterized the similarity between two nodes based on their local neighborhood structures using graph kernels and solved the name disambiguation problem using SVM. However, the major drawback

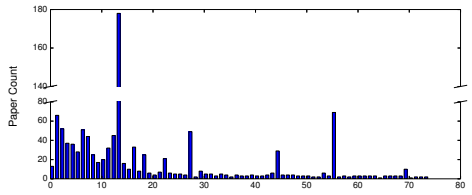


Figure 1: Paper Count Distribution of "S Lee"

of the proposed method in [14] is that it can only detect entities that should be disambiguated, but fails to further partition the documents into their corresponding homogeneous groups. Authors in [20, 32] proposed an unsupervised solution to name disambiguation in an anonymized graph by exploiting the time-stamped network topology around a vertex. However, it also suffers from the similar issue as described above.

Our proposed solution utilizes a network representation learning based approach [2, 4, 7, 9, 19, 23–26]— a rather recent development in machine learning. Many of these methods are inspired by word embedding based language model [18]. Different from traditional graph embedding methods, such as Laplacian Eigenmaps [5, 6], the recently proposed network embedding methods, such as DeepWalk [19], LINE [24], PTE [23], and Node2Vec [9], are more scalable and have shown better performance in node classification and link prediction tasks. Among these works, LINE [24] finds embedding of documents by using document-document similarity matrix, whereas our work uses multiple networks and performs a joint learning. PTE [23] performs a joint learning of multiple input graphs, but PTE needs labeled data. Finally, the embedding formulation and optimization of our proposed method is different than LINE or PTE. Specifically, we use a ranking based loss function as our objective function whereas mostly all the existing methods use K-L divergence based objective function.

3 PROBLEM FORMULATION

We first introduce notations used in this paper. Throughout the paper, bold uppercase letter (e.g., $\mathbf{X}$) denotes a matrix, bold lowercase letter such as $\mathbf{x}_i$ denotes a column vector, and $(\cdot)^T$ denotes vector transpose. $\|\mathbf{X}\|_F$ is the Frobenius norm of matrix $\mathbf{X}$. Calligraphic uppercase letter (e.g., $\mathcal{X}$) is used to denote a set and $|\mathcal{X}|$ is the cardinality of the set $\mathcal{X}$.

For a given name reference a, we denote $\mathcal{D}^a = \{d_1^a, d_2^a, ..., d_N^a\}$ to be a set of N documents with which a is associated and $\mathcal{A}^a = \{a_1, a_2, ..., a_M\}$ is the collaborator set of a in $\mathcal{D}^a$, where $a \notin \mathcal{A}^a$. If there is no ambiguity we remove the superscript a in the notations of both $\mathcal{D}^a$ and $\mathcal{A}^a$ and refer the terms as $\mathcal{D}$ and $\mathcal{A}$, respectively. For illustration, in bibliographic field, $\mathcal{D}$ can be the set of scholarly publications where a is one of the authors and A is the set of a's coauthors. In real-life, the given name reference a can be associated with multiple persons (say L) all sharing the same name. The task of name disambiguation is to partition $\mathcal{D}$ into L disjoint sets such that each partition contains documents of a unique person entity with name reference a.

Though it may appear as a simple clustering problem, name disambiguation is challenging on real-life data. This is due to the fact that it requires solving a highly class-imbalanced clustering task, as the number of documents associated with a distinct person follows a power-law distribution. We demonstrate it through an example from the bibliographic domain. In Figure 1, we show a histogram of paper counts of various real-life persons named "S Lee" in CiteSeerX [3]. As we can observe, there are a few real-life authors (dominant entities) with the name "S Lee" to whom the majority of the publications belong. Only a few publications belong to each of the remaining real-life authors with name "S Lee". Due to this severe class imbalance issue, majority of traditional clustering methods perform poorly on this task. Sophisticated machine learning models, like the one we propose below are needed for solving this task. This example is from bibliographic domain, but power-law distribution of possession is common in every aspect of real-life, so we expect this challenge to hold in other domains as well.

In this study, we investigate the name disambiguation problem in a restricted setup, where bibliographical features and information from external sources are not considered so that the risk of privacy violation can be alleviated. Instead, we formulate the problem using graphs in which each node has been assigned an anonymized identifier, and network topological structure is the only information available. Specifically, our solution encodes the local neighborhood structures accumulated from three different networks into a proposed network embedding model, which generates a k-dimensional vector representation for each document. The networks are person-person network, person-document network, and linked document network, which we formally define as below:

Definition 3.1 (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 person-person network is important because the inter-person acquaintances represented by collaboration relation can be used to discriminate the set of documents of multiple real-life persons. However, the collaboration network does not account for the fact that the documents associated with the same real-life person are inherently similar; person-document network and document-document network cover for this shortcoming.

Definition 3.2 (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 documents with which the name reference a is associated and $\mathcal{A}$ 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, a_i is simply an author of the document d_j and the weight $w_{ij} = 1$.

Definition 3.3 (Linked Document Network). Document-Document Network, represented as $G_{dd} = (\mathcal{D}, E_{dd})$, where each vertex $d_i \in \mathcal{D}$

[3] http://citeseerx.ist.psu.edu/index;jsessionid=4A26742FADC605600567F493C2D7825E

is a document. If two documents d_i and d_j are similar (more discussion is forthcoming), we build an edge between them represented as $e_{ij} \in E_{dd}$.

There are several ways document-document similarity can be captured. For instance, one can find word co-occurrence between different documents to compute this similarity. However, we refrained from using word co-occurrence due to the privacy concern as sometimes a list of a set of unique words can reveal the identity of a person [31]. Instead we define document-document similarity through a combination of person-person and person-document relationships. Two documents are similar if the intersection of their collaborator-sets is large (by using person-document relationship) or if the intersection of one-hop neighbors of their collaborator-sets is large (by using both person-document and person-person relationships).

The above definition of document similarity captures two important patterns which facilitate effective name disambiguation by document clustering. First, there is a high chance for two documents to be authored by the same real-life person, if they have a large number of overlapping collaborators. Second, even if they do not have any overlapping collaborators, large overlap in the neighbors of their collaborators signals that the documents are most likely authored by the same person. For both cases, these two documents should be placed in close proximity in the embedded space. Mathematically, we denote $\mathcal{A}^1_{d_i}$ as the collaborator set of d_i. Furthermore, $\mathcal{A}^2_{d_i}$ is the set of collaborators by extending $\mathcal{A}^1_{d_i}$ with all neighbors of the persons in $\mathcal{A}^1_{d_i}$, namely $\mathcal{A}^2_{d_i} = \mathcal{A}^1_{d_i} \cup \{\mathcal{NB}_{G_{pp}}(b)\}_{b \in \mathcal{A}^1_{d_i}}$, where $\mathcal{NB}_{G_{pp}}(b)$ is the set of neighbors of node b in person-person network G_{pp}. Then the document similarity between d_i and d_j in the graph G_{dd} is simply defined as $w_{ij} = |\mathcal{A}^2_{d_i} \cap \mathcal{A}^2_{d_j}|$.

Based on our problem formulation, the name disambiguation solution consists of two phases: (1) document representation (2) disambiguation. We discuss them as below:

Given a name reference a, its associated document set $\mathcal{D}^a$ (which we want to disambiguate) and the collaborator set $\mathcal{A}^a$, the document representation phase first constructs corresponding person-person network G_{pp}, person-document bipartite network G_{pd}, and linked document network G_{dd}. Then our proposed document representation model combines structural information from these three networks to generate a k-dimensional document embedding matrix $\mathbf{D} = [\mathbf{d}_1^T, ..., \mathbf{d}_N^T] \in \mathbb{R}^{N \times k}$.

Then the disambiguation phase takes the document embedding matrix $\mathbf{D}$ as input and applies the hierarchical agglomerative clustering (HAC) with group average merging criteria to partition N documents in $\mathcal{D}^a$ into L disjoint sets with the expectation that each set is composed of documents of a unique person entity sharing the name reference a. At this stage, L is a user-defined parameter which we match with the ground truth during the evaluation phase. In real-life though, a user needs to tune the parameter L which can easily be done with HAC, because HAC provides hierarchical organization of clusters at all levels starting from a single cluster upto the case of single-instance cluster, and a user can recover clustering for any value of L as needed without additional cost. Also, across different L values the cluster assignment of HAC is consistent (i.e., two instances that are in the same cluster for some L value will remain in the same cluster for any smaller L value), which further helps in choosing an appropriate L value.

4 METHOD

In this section, we discuss our proposed representation learning model for name disambiguation. Our goal is to encode the local neighborhood structures captured by the three networks (see Definitions 3.1 3.2 3.3) into the k-dimensional document embedding matrix with strong name disambiguation ability.

4.1 Model Formulation

The main intuition of our network embedding model is that neighboring nodes in a graph should have more similar vector representation in the embedding space than non-neighboring nodes. For instance, in linked document network, the affinity between two neighboring vertices d_i and d_j, i.e., $e_{ij} \in G_{dd}$ should be larger than the affinity between two non-neighboring vertices d_i and d_t, i.e., $e_{it} \notin G_{dd}$. The affinity score between two nodes d_i and d_j in G_{dd} can be calculated as the inner product of their corresponding embedding representations, denoted as $S^{dd}_{ij} = \mathbf{d}_i^T \mathbf{d}_j$. More specifically, we model the probability of preserving ranking order $S^{dd}_{ij} > S^{dd}_{it}$ using the logistic function $\sigma(x) = \frac{1}{1+e^{-x}}$. Mathematically,

$$P(S^{dd}_{ij} > S^{dd}_{it} | \mathbf{d}_i, \mathbf{d}_j, \mathbf{d}_t) = \sigma(S^{dd}_{ijt}) \tag{1}$$

where S^{dd}_{ijt} is defined as below:

$$\begin{aligned} S^{dd}_{ijt} &= S^{dd}_{ij} - S^{dd}_{it} \\ &= \mathbf{d}_i^T \mathbf{d}_j - \mathbf{d}_i^T \mathbf{d}_t \end{aligned} \tag{2}$$

As we observe from Equation 1, the larger S^{dd}_{ijt}, the more likely ranking order $S^{dd}_{ij} > S^{dd}_{it}$ is preserved. By assuming all the ranking orders generated from the linked document network G_{dd} to be independent, the probability $P(> |\mathbf{D})$ of all the ranking orders being preserved given the document embedding matrix $\mathbf{D} \in \mathbb{R}^{N \times k}$ is defined as below:

$$\begin{aligned} P(> |\mathbf{D}) &= \prod_{\substack{(d_i,d_j) \in \mathcal{P}_{G_{dd}} \\ (d_i,d_t) \in \mathcal{N}_{G_{dd}}}} P(S^{dd}_{ij} > S^{dd}_{it} | \mathbf{d}_i, \mathbf{d}_j, \mathbf{d}_t) \\ &= \prod_{\substack{(d_i,d_j) \in \mathcal{P}_{G_{dd}} \\ (d_i,d_t) \in \mathcal{N}_{G_{dd}}}} \sigma(S^{dd}_{ijt}) \\ &= \prod_{\substack{(d_i,d_j) \in \mathcal{P}_{G_{dd}} \\ (d_i,d_t) \in \mathcal{N}_{G_{dd}}}} \sigma(S^{dd}_{ij} - S^{dd}_{it}) \end{aligned} \tag{3}$$

where $\mathcal{P}_{G_{dd}}$ and $\mathcal{N}_{G_{dd}}$ are positive and negative training sets in G_{dd}.

From the Equation 3, the goal is to seek the document latent representation $\mathbf{D}$ for all nodes in linked document network G_{dd},

which maximizes $P(>|\mathbf{D})$. For the computational convenience, we minimize the following sum of negative log-likelihood objective, which is shown as follows:

$$\begin{aligned} OBJ_{dd} &= \min_{\mathbf{D}} -\ln P(>|\mathbf{D}) \\ &= -\sum_{\substack{(d_i,d_j)\in\mathcal{P}_{G_{dd}} \\ (d_i,d_t)\in\mathcal{N}_{G_{dd}}}} \ln P\big(S_{ij}^{dd} > S_{it}^{dd}|\mathbf{d}_i,\mathbf{d}_j,\mathbf{d}_t\big) \\ &= -\sum_{\substack{(d_i,d_j)\in\mathcal{P}_{G_{dd}} \\ (d_i,d_t)\in\mathcal{N}_{G_{dd}}}} \ln \sigma(S_{ijt}^{dd}) \\ &= -\sum_{\substack{(d_i,d_j)\in\mathcal{P}_{G_{dd}} \\ (d_i,d_t)\in\mathcal{N}_{G_{dd}}}} \ln \sigma\big(S_{ij}^{dd} - S_{it}^{dd}\big) \end{aligned} \tag{4}$$

The formulation shown in Equation 4 constructs a probabilistic framework for distinguishing between neighbor nodes and non-neighbor nodes in a linked document network by preserving a ranking order objective function.

Using the identical argument, the objective functions for capturing person-person and person-document relations are given as below:

$$\begin{aligned} OBJ_{pp} &= \min_{\mathbf{A}} -\ln P(>|\mathbf{A}) \\ &= -\sum_{\substack{(a_i,a_j)\in\mathcal{P}_{G_{pp}} \\ (a_i,a_t)\in\mathcal{N}_{G_{pp}}}} \ln \sigma(S_{ij}^{pp} - S_{it}^{pp}) \end{aligned} \tag{5}$$

$$\begin{aligned} OBJ_{pd} &= \min_{\mathbf{A},\mathbf{D}} -\ln P(>|\mathbf{A},\mathbf{D}) \\ &= -\sum_{\substack{(d_i,a_j)\in\mathcal{P}_{G_{pd}} \\ (d_i,a_t)\in\mathcal{N}_{G_{pd}}}} \ln \sigma(S_{ij}^{pd} - S_{it}^{pd}) \end{aligned} \tag{6}$$

where $\mathbf{A}\in\mathbb{R}^{M\times k}$ can be thought as the person embedding matrix and M is the number of persons in the collaborator set $\mathcal{A}$. S_{ij}^{pp} represents the affinity score between two nodes a_i and a_j in collaboration graph G_{pp}, and S_{ij}^{pd} denotes the affinity score between two nodes d_i and a_j in heterogeneous bipartite graph G_{pd}. Finally, $\mathcal{P}_{G_{pp}}$ and $\mathcal{N}_{G_{pp}}$ are positive and negative training sets in G_{pp}, $\mathcal{P}_{G_{pd}}$ and $\mathcal{N}_{G_{pd}}$ are positive and negative training sets in G_{pd} respectively.

The goal of proposed network embedding framework is to unify these three types of relations together, where the person and document vertices are shared across these three networks. An intuitive manner is to collectively embed these three networks, which can be achieved by minimizing the following objective function:

$$OBJ = \min_{\mathbf{A},\mathbf{D}} -OBJ_{pp} - OBJ_{pd} - OBJ_{dd} + \lambda Reg(\mathbf{A},\mathbf{D}) \tag{7}$$

where $\lambda Reg(\mathbf{A},\mathbf{D})$ in Equation 7 is a l_2-norm regularization term to prevent the model from overfitting. Here for the computational convenience, we set $Reg(\mathbf{A},\mathbf{D})$ as $\|\mathbf{A}\|_F^2 + \|\mathbf{D}\|_F^2$. Such pairwise ranking loss objective is in the similar spirit to the Bayesian Personalized Ranking [8, 29], which aims to predict the interaction between users and items in recommender system domain.

4.2 Model Optimization

We use stochastic gradient descent (SGD) algorithm for optimizing Equation 7. Specifically, in each step we sample the training instances involved in person-person, person-document, and document-document relations accordingly. The sampling strategy of positive instances is based on edge sampling [23]. Specifically, for example, in linked document network G_{dd}, given an arbitrary node d_i, we sample one of its neighbors d_j, i.e., $(d_i,d_j)\in\mathcal{P}_{G_{dd}}$, with the probability proportional to the edge weight for the model update. On the other hand, for sampling of negative instances, we utilize uniform sampling technique. In particular, given the sampled node d_i, we sample an arbitrary negative instance d_t uniformly, namely $(d_i,d_t)\in\mathcal{N}_{G_{dd}}$.

Therefore given a sampled triplet (d_i,d_j,d_t) with $(d_i,d_j)\in\mathcal{P}_{G_{dd}}$ and $(d_i,d_t)\in\mathcal{N}_{G_{dd}}$, using the chain rule and back-propagation, the gradient of the objective function OBJ in Equation 7 w.r.t. $\mathbf{d}_i$ can be computed as below:

$$\begin{aligned} \frac{\partial OBJ}{\partial \mathbf{d}_i} &= -\frac{\partial \ln\sigma\big(S_{ij}^{dd} - S_{it}^{dd}\big)}{\partial \mathbf{d}_i} + 2\lambda\mathbf{d}_i \\ &= -\frac{\partial \ln\sigma\big(S_{ij}^{dd} - S_{it}^{dd}\big)}{\partial \sigma\big(S_{ij}^{dd} - S_{it}^{dd}\big)} \times \frac{\partial \sigma\big(S_{ij}^{dd} - S_{it}^{dd}\big)}{\partial\big(S_{ij}^{dd} - S_{it}^{dd}\big)} \\ &\quad \times \frac{\partial\big(S_{ij}^{dd} - S_{it}^{dd}\big)}{\partial \mathbf{d}_i} + 2\lambda\mathbf{d}_i \\ &= -\frac{1}{\sigma\big(S_{ij}^{dd} - S_{it}^{dd}\big)} \times \sigma\big(S_{ij}^{dd} - S_{it}^{dd}\big) \\ &\quad \Big(1 - \sigma\big(S_{ij}^{dd} - S_{it}^{dd}\big)\Big) \times (\mathbf{d}_j - \mathbf{d}_t) + 2\lambda\mathbf{d}_i \\ &= \left(\frac{-e^{-(\mathbf{d}_i^T\mathbf{d}_j - \mathbf{d}_i^T\mathbf{d}_t)}}{1 + e^{-(\mathbf{d}_i^T\mathbf{d}_j - \mathbf{d}_i^T\mathbf{d}_t)}}\right)(\mathbf{d}_j - \mathbf{d}_t) + 2\lambda\mathbf{d}_i \end{aligned} \tag{8}$$

Using the similar chain rule derivation, the gradient of the objective function OBJ w.r.t. $\mathbf{d}_j$ and $\mathbf{d}_t$ can be obtained as follows:

$$\frac{\partial OBJ}{\partial \mathbf{d}_j} = \left(\frac{-e^{-(\mathbf{d}_i^T\mathbf{d}_j - \mathbf{d}_i^T\mathbf{d}_t)}}{1 + e^{-(\mathbf{d}_i^T\mathbf{d}_j - \mathbf{d}_i^T\mathbf{d}_t)}}\right) \times \mathbf{d}_i + 2\lambda\mathbf{d}_j \tag{9}$$

$$\frac{\partial OBJ}{\partial \mathbf{d}_t} = \left(\frac{-e^{-(\mathbf{d}_i^T\mathbf{d}_j - \mathbf{d}_i^T\mathbf{d}_t)}}{1 + e^{-(\mathbf{d}_i^T\mathbf{d}_j - \mathbf{d}_i^T\mathbf{d}_t)}}\right) \times (-\mathbf{d}_i) + 2\lambda\mathbf{d}_t \tag{10}$$

Then embedding vectors $\mathbf{d}_i$, $\mathbf{d}_j$, and $\mathbf{d}_t$ are updated as below:

$$\mathbf{d}_i = \mathbf{d}_i - \alpha \frac{\partial OBJ}{\partial \mathbf{d}_i}$$
$$\mathbf{d}_j = \mathbf{d}_j - \alpha \frac{\partial OBJ}{\partial \mathbf{d}_j}$$
$$\mathbf{d}_t = \mathbf{d}_t - \alpha \frac{\partial OBJ}{\partial \mathbf{d}_t} \tag{11}$$

where α is the learning rate.

Likewise, when the training instances come from person-person network, and person-document bipartite network, we update their corresponding gradients accordingly. We omit the detailed derivations here since they are very similar to the aforementioned ones.

Algorithm 1 Network Embedding based Name Disambiguation in Anonymized Graphs

Input: name reference a, dimension k, λ, α, L
Output: document embedding matrix $\mathbf{D}$ and its clustering membership set C

1: Given name reference a, construct its associated $\mathcal{D}^a$, $\mathcal{A}^a$, G_{pp}, G_{pd}, G_{dd}
2: Given G_{pp}, G_{pd}, G_{dd}, construct training sample sets $\mathcal{P}_{G_{pp}}$, $\mathcal{N}_{G_{pp}}$, $\mathcal{P}_{G_{pd}}$, $\mathcal{N}_{G_{pd}}$, $\mathcal{P}_{G_{dd}}$, $\mathcal{N}_{G_{dd}}$ respectively based on edge sampling and uniform sampling techniques
3: Initialize $\mathbf{A}$ and $\mathbf{D}$ as k-dimensional matrices
4: **for** each training instance in training sample sets **do**
5: Update involved parameters using SGD as described in Section 4.2
6: **end for**
7: Given $\mathbf{D}$ and L, perform HAC to partition N documents in $\mathcal{D}^a$ into L disjoint sets for name disambiguation
8: **return** $\mathbf{D}$, $C = \{c_1, c_2, ..., c_N\}$

4.3 Pseudo-code and Complexity Analysis

The pseudo-code of the proposed network embedding method for name disambiguation under anonymized graphs is summarized in Algorithm 1. The entire process consists of two phases: network embedding for document representation and name disambiguation by clustering. Specifically, given a name reference a and its associated document set $\mathcal{D}^a$ we aim to disambiguate, we first prepare the training instances in Line 1-2. Line 3 initializes the person and document embedding matrices $\mathbf{A}$ and $\mathbf{D}$ by randomly sampling elements from uniform distribution $[-0.2, 0.2]$. Then we train our proposed network embedding model and update $\mathbf{A}$ and $\mathbf{D}$ using the training samples based on the SGD optimization in Line 4-6. Then given the obtained document embedding matrix $\mathbf{D}$ and L, in Line 7, we perform HAC to partition N documents in $\mathcal{D}^a$ into L disjoint sets such that each partition contains documents of a unique person entity with name reference a. Finally in Line 8, we return document embedding matrix $\mathbf{D}$ and its clustering membership set $C = \{c_1, ..., c_i, ..., c_N\}$ for evaluation, where $1 \le c_i \le L$.

For the time complexity analysis, for the document embedding, when the training sample is $(d_i, d_j) \in \mathcal{P}_{G_{dd}}$, as observed from Equations 8, 9 and 11, the cost of calculating gradient of OBJ w.r.t. $\mathbf{d}_i$ and $\mathbf{d}_j$, and updating $\mathbf{d}_i$ and $\mathbf{d}_j$ are both $O(k)$. Similar analysis can be applied when training instances are from $\mathcal{P}_{G_{pp}}$, $\mathcal{N}_{G_{pp}}$, $\mathcal{P}_{G_{pd}}$, $\mathcal{N}_{G_{pd}}$, $\mathcal{N}_{G_{dd}}$. Therefore, the total computational cost is $\big(2 * |\mathcal{P}_{G_{pp}}| + 2 * |\mathcal{P}_{G_{pd}}| + 2 * |\mathcal{P}_{G_{dd}}|\big)O(k)$. For the name disambiguation, the computational cost of hierarchical clustering is $O(N^2 logN)$ [28]. So the total computational complexity of Algorithm 1 is $\big(2 * |\mathcal{P}_{G_{pp}}| + 2 * |\mathcal{P}_{G_{pd}}| + 2 * |\mathcal{P}_{G_{dd}}|\big)O(k) + O(N^2 logN)$.

Name Reference	# Documents	# Distinct Authors
Jing Zhang	160	33
Bin Yu	78	8
Rakesh Kumar	82	5
Lei Wang	222	48
Bin Li	135	14
Yang Wang	134	23
Bo Liu	93	19
Yu Zhang	156	26
David Brown	42	9
Wei Xu	111	21

Table 1: Arnetminer Name Disambiguation Dataset

5 EXPERIMENTS AND RESULTS

We perform several experiments to validate the performance of our proposed network embedding method for solving the name disambiguation task in a privacy-preserving setting using only linked data. We also compare our method with various other methods to demonstrate its superiority over those methods.

5.1 Datasets

A key challenge for the evaluation of name disambiguation task is the lack of availability of labeled datasets from diverse application domains. In recent years, the bibliographic repository sites, Arnetminer [4] and CiteSeerX [5] have published several ambiguous author name references along with respective ground truths (paper list of each real-life author), which we use for evaluation. From each of these two sources, we use 10 highly ambiguous (having a larger number of distinct authors for a given name) name references and show the performance of our method on these name references. The statistics of name references in Arnetminer and CiteSeerX datasets are shown in Table 1 and Table 2, respectively. In these tables, for each name reference, we show the number of documents, and the number of distinct authors associated with that name reference. It is important to understand that the name disambiguation model is built on a name reference, not on a source dataset such as, Arnetminer or CiteSeerX as a whole, so each name reference is a distinct dataset on which the evaluation is performed.

5.2 Competing Methods

To validate the disambiguation performance of our proposed approach, we compare it against 9 different methods. For a fair comparison, all of these methods accommodate the name disambiguation using only relational data. Among all the competing methods, Rand, AuthorList, and AuthorList-NNMF are a set of primitive

[4] https://aminer.org/disambiguation
[5] http://clgiles.ist.psu.edu/data/

Name Reference	# Documents	# Distinct Authors
K Tanaka	174	9
M Jones	191	10
J Smith	798	26
Y Chen	848	64
J Martin	51	13
A Kumar	149	10
J Robinson	123	9
M Brown	118	13
J Lee	891	93
S Lee	1091	74

Table 2: CiteSeerX Name Disambiguation Dataset

baselines that we have designed. But, the remaining methods are taken from recently published works. For instance, GF, DeepWalk, LINE, Node2Vec, and PTE are existing state-of-the-art approaches for vertex embedding, which we use for name disambiguation by clustering the documents using HAC in the embedding space similar to our approach. Graphlet based graph kernel methods (GL3, GL4) are existing state-of-the-art approaches for name disambiguation in anonymized graphs. More details of each of the competing methods are given below. For each method, for a given name reference, a list of documents need to be partitioned among L (user defined) different clusters.

(1) Rand: This naive method randomly assigns one of existing classes to the associated documents.

(2) AuthorList: Given the associated documents, we first aggregate the author-list of all documents in an author-array, then define a binary feature for each author, indicating his presence or absence in the author-list of that document. Finally we use HAC with the generated author-list as features for disambiguation task.

(3) AuthorList-NNMF: We perform Non-Negative Matrix Factorization (NNMF) on the generated author-list features the same way described above. Then the latent features from NNMF are used in a HAC framework for disambiguation task.

(4) Graph Factorization (GF) [16]: We first represent co-authorship network G_{pp} and the linked document network G_{dd} as affinity matrices, and then utilize matrix factorization technique to represent each document into low-dimensional vector. Note that GF is optimized via a point-wise regression model that minimizes a square loss function. However, in our proposed embedding approach, the objective aims to minimize a ranking loss function, which is substantially different from GF.

(5) DeepWalk [19]: DeepWalk is an approach recently proposed for network embedding, which is only applicable for homogeneous network with binary edges. Given G_{pp} and G_{dd}, we use uniform random walk to obtain the contextual information of its neighborhood for document embedding [6].

(6) LINE [24]: LINE aims to learn the document embedding that preserves both the first-order and second-order proximities [7]. Note that LINE can only handle the embedding of homogeneous network and the embedding formulation and optimization are quite different from the one proposed in our work.

(7) Node2Vec [9]: Similar to DeepWalk, Node2Vec designs a biased random walk procedure for document embedding. [8].

(8) PTE [23]: Predictive Text Embedding (PTE) framework aims to capture the relations of word-word, word-document, and word-label. However, such keyword and label based biographical features are not available in the anonymized setup. Instead we utilize local structural information of both G_{pp} and G_{pd} networks to learn the document embedding. However, this approach is not able to capture the linked information among documents.

(9) Graph Kernel [14]: In this work, size-3 graphlets (GL3) and size-4 graphlets (GL4) are used to build graph kernels, which measure the similarity between documents. Then the learned similarity metric is used as features in HAC for name disambiguation. As we see, both kernels only use network topological information. [9]

5.3 Experimental Setting and Implementation

For each of the 20 name references, we perform name disambiguation task using our proposed method and each of the competing methods to demonstrate that our proposed method is superior than the competing methods. For evaluation metric, we use Macro-F1 measure [28], which is the unweighted average of F1 measure of each class. The range of Macro-F1 measure is between 0 and 1, and a higher value indicates better disambiguation performance. Besides comparison with competing methodologies, we also perform experiments to show that our method is robust against the variation of user defined parameters (specifically, embedding dimension and the number of clusters) over a wide range of parameter values. Experiments are also performed to show how the embedding model performs with each of the three types of networks (person-person, person-document, and document-document) incrementally added. Finally, we show the convergence of the learning model while performing the document embedding phase.

There are a few user defined parameters in our proposed embedding model. The first among these is the embedding dimension k, which we set to be 20. For the regularization parameter in model inference (see Section 4.2), we perform grid search on the validation set in the following range: $\lambda = \{0.001, 0.005, 0.01, 0.1, 1, 10\}$. In addition to that, we fix the learning rate $\alpha = 0.02$. For the disambiguation stage, we use the actual number of classes L of each name reference as input to perform HAC. For both data processing and model implementation, we implement our own code in Python and use NumPy, SciPy, scikit-learn, and Networkx libraries for linear algebra, machine learning, and graph operations. We run all the experiments on a 2.1 GHz Machine with 8GB memory running Linux operating system.

5.4 Comparison among Various Name Disambiguation Methods

Table 3 and Table 4 show the performance comparison of name disambiguation between our proposed method and other competing methods for all 20 name references (one table for ArnetMiner names, and the other for CiteSeerX names). In both tables, the rows correspond to the name references and the columns (2 to 12) stand for various methods. The competing methods are grouped

[6] Code is available at http://www.perozzi.net/projects/deepwalk/

[7] Implementation Code is available at https://github.com/tangjianpku/LINE

[8] We use the code from https://github.com/aditya-grover/node2vec

[9] The kernel values are obtained by source code supplied by the original authors

Name Reference	**Our Method**	Rand	AuthorList	AuthorList-NNMF	GF [16]	DeepWalk [19]	LINE [24]	Node2Vec [9]	PTE [23]	GL3 [14]	GL4 [14]	Improv.
Jing Zhang	**0.734 (0.014)**	0.192	0.327	0.463	0.669	0.654	0.651	0.312	0.458	0.318	0.329	9.7%
Bin Yu	**0.804 (0.009)**	0.201	0.371	0.283	0.610	0.644	0.643	0.531	0.399	0.489	0.504	24.8%
Rakesh Kumar	**0.834 (0.012)**	0.226	0.305	0.404	0.448	0.617	0.641	0.372	0.219	0.434	0.407	30.1%
Lei Wang	**0.805 (0.021)**	0.198	0.502	0.424	0.633	0.419	0.639	0.263	0.447	0.291	0.321	26.0%
Bin Li	**0.848 (0.016)**	0.172	0.610	0.733	0.761	0.392	0.641	0.186	0.349	0.336	0.418	11.4%
Yang Wang	**0.798 (0.011)**	0.199	0.442	0.532	0.575	0.640	0.623	0.331	0.444	0.378	0.512	24.7%
Bo Liu	0.831 (0.022)	0.215	0.482	0.740	**0.850**	0.788	0.781	0.459	0.373	0.498	0.347	-2.2%
Yu Zhang	**0.820 (0.031)**	0.186	0.519	0.566	0.565	0.454	0.658	0.196	0.385	0.369	0.305	24.6%
David Brown	**1.00 (0.00)**	0.304	0.818	0.583	0.802	0.494	**1.00**	0.221	0.575	0.603	0.698	0%
Wei Xu	**0.793 (0.014)**	0.256	0.527	0.564	0.625	0.228	0.599	0.136	0.236	0.386	0.428	26.9%

Table 3: Comparison of Macro-F1 values between our proposed method and other competing methods for name disambiguation task in Arnetminer dataset (embedding dimension = 20). Paired t-test is conducted on all performance comparisons and it shows that all improvements are significant at the 0.05 level.

Name Reference	**Our Method**	Rand	AuthorList	AuthorList-NNMF	GF [16]	DeepWalk [19]	LINE [24]	Node2Vec [9]	PTE [23]	GL3 [14]	GL4 [14]	Improv.
K Tanaka	**0.706 (0.018)**	0.178	0.202	0.168	0.334	0.450	0.398	0.304	0.173	0.235	0.276	56.9%
M Jones	**0.743 (0.009)**	0.184	0.189	0.261	0.529	0.696	0.688	0.513	0.348	0.216	0.398	6.8%
J Smith	**0.503 (0.007)**	0.083	0.121	0.280	0.316	0.098	0.104	0.073	0.136	0.201	0.237	59.2%
Y Chen	0.367 (0.019)	0.069	0.325	0.355	**0.439**	0.118	0.193	0.058	0.199	0.334	0.385	-16.4%
J Martin	**0.898 (0.021)**	0.310	0.624	0.536	0.755	0.728	0.774	0.629	0.587	0.414	0.431	16.0%
A Kumar	**0.645 (0.006)**	0.166	0.251	0.375	0.319	0.407	0.395	0.424	0.247	0.192	0.234	52.1%
J Robinson	**0.796 (0.033)**	0.200	0.348	0.438	0.393	0.513	0.603	0.608	0.345	0.271	0.316	30.9%
M Brown	**0.741 (0.028)**	0.171	0.306	0.573	0.478	0.481	0.633	0.211	0.269	0.297	0.248	17.1%
J Lee	0.366 (0.038)	0.089	0.262	0.256	0.231	**0.387**	0.134	0.181	0.142	0.189	0.205	-5.4%
S Lee	**0.624 (0.015)**	0.057	0.214	0.248	0.345	0.194	0.109	0.044	0.074	0.215	0.268	80.9%

Table 4: Comparison of Macro-F1 values between our proposed method and other competing methods for name disambiguation task in CiteSeerX dataset (embedding dimension = 20). Paired t-test is conducted on all performance comparisons and it shows that all improvements are significant at the 0.05 level.

logically. The first group includes the baseline methods that we have designed such as random predictor (Rand) and methods using low-dimensional factorization of author-list for clustering. The second group includes various state-of-the-art network embedding methodologies, and the third group includes two methods using graphlet based graph kernels. The cell values are the performance of a method using Macro-F1 score for disambiguation of documents under a given name reference. The last column shows the overall improvement of our proposed method compared with the best competing method. Since SGD based optimization technique in our proposed embedding model is a randomized method, for each name reference we execute the method 10 times and report the average Macro-F1 score. For our method, we also show the standard deviation in the parenthesis. [10] For better visual comparison, we highlight the best Macro-F1 score of each name reference with bold-face font.

As we observe, our proposed embedding model performs the best for 9 and 8 name references (out of 10) in Table 3, and Table 4, respectively. Besides, the overall percentage improvement that our method delivers over the second best method is relatively large. For an example, consider the name "S Lee" shown in the last row of Table 4. This is a difficult disambiguation task; from Table 2, it has 1091 documents and 74 distinct real-life authors ! A random predictor (Rand) obtains a Macro-F1 of only 0.057 due to the large number of classes. Whereas our method achieves 0.624 Macro-F1 score for this name reference; the second best method for this name (GF) achieves only 0.345, indicating a substantial improvement (80.9%) by our method. The relatively good performance of our proposed method across all the name references is due to the fact that the method is able to learn document embedding, which is particularly suited for the name disambiguation task by facilitating information exchange among the three networks (see Section 3).

Among the competing methods, AuthorList based methods perform poorly because the binary features are not intelligent enough to disambiguate documents, even after using traditional low dimensional embedding by non-negative matrix factorization. Graph kernel based methods such as GL3 and GL4 also have similar fate; the possible reason could be that the size-3 and size-4 graphlet structures are not decisive patterns to distinguish documents authored by different persons. On the other hand, embedding based methods are much better as they are able to learn effective features, which bring the documents authored by the same real-life person in close proximity in the feature space. This finding justifies our approach of choosing a document embedding method for solving name disambiguation. Among the competing network embedding based approaches, as we can observe from all name references, no single method emerges as a clear winner. To be more

[10] Standard deviation for other competing methods are not shown due to the space limit.

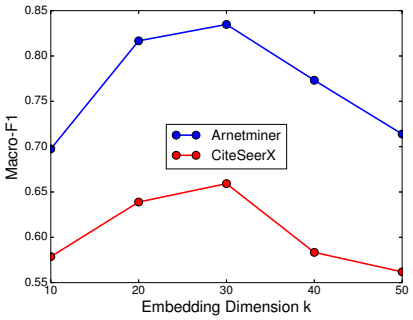


Figure 2: The effects of embedding dimension on the name disambiguation performance

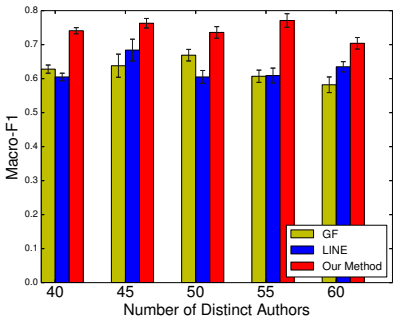


Figure 3: Macro-F1 results of multiple L values on name reference "Lei Wang" using Our Method, GF, and LINE (embedding dimension = 20).

precise, PTE performs poorly as it fails to incorporate linked structural information among the documents. Both GF and LINE outperform DeepWalk in majority of name references. This is because DeepWalk ignores the weights of the edges, which is considered to be very important in the linked document network. However, neither of embedding based competing methods could encode the document co-occurrence by exploiting the information from multiple networks, which is exploited by our proposed model. Besides, as mentioned earlier, our similarity ranking based objective function is better suited than the K-L divergence based objective functions for placing the nodes in the embedding space for facilitating a downstream clustering task. This is possibly a significant reason for our method to show superior performance over the existing network embedding based methods.

5.5 Parameter Sensitivity of Embedding Dimension

We also perform experiment to show how the embedding dimension k affects the disambiguation performance of our proposed method. Specifically, we vary the number of embedding dimension k as $\{10, 20, 30, 40, 50\}$. For the sake of space, in each of the datasets, we show the average results over all the 10 name references. The disambiguation results are given in Figure 2. As we observe, for both datasets, as the dimension of embeddings increases, the disambiguation performance in terms of Macro-F1 first increases and then decreases. The possible explanation could be that when the embedding dimension is too small, the embedding representation capability is not sufficient. However, when the embedding dimension is too large, the proposed embedding model may overfit the data, leading to the unsatisfactory disambiguation performance.

5.6 Performance Comparison over the Number of Clusters

One of the potential problems for name disambiguation is to determine the number of real-life persons L under a given name reference, because in real-life L is generally unknown a-priori. So a method whose performance is superior over a range of L values should be preferred. For this comparison, after learning the document representation, we use various L values as input in the HAC

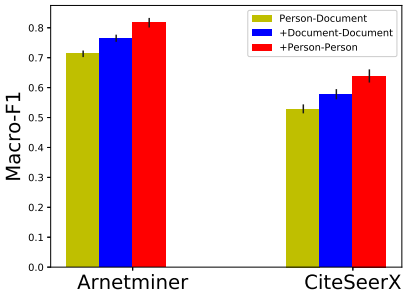
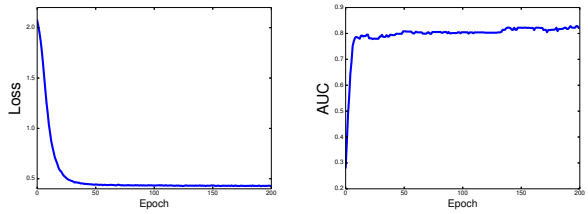


Figure 4: Component Contribution Analysis in terms of name disambiguation performance using Arnetminer and CiteSeerX as a whole source (embedding dimension = 20).

for name disambiguation and record the Macro-F1 score over different L for the competing methods. In our experiment, we compare Macro-F1 value of our method with two other best performing methods over several names, but due to space limitation, we show this result only for one name ("Lei Wang" in Arnetminer) using bar-charts in Figure 3. In this figure, we compare the performance differences between our method with two other best performing methods (GF and LINE) as we vary L as $\{40, 45, 50, 55, 60\}$. Note that the actual number of distinct authors under "Lei Wang" is 48 as shown in Table 1. As we can see, our proposed method always outperforms the state-of-the-art with all different L values, and the overall improvement of our method over these two methods is statistically significant with a p-value of less than 0.01. Because of the robustness of our proposed embedding method for name disambiguation regardless of L values, this is a better method for the real-life application.

5.7 Component Contribution Analysis

Our proposed network embedding model is composed of three types of networks, namely person-person, person-document, and linked document networks (explained in Section 3). In this section we study the contribution of each of the three components for the



(a) Loss vs Epoch (b) AUC vs Epoch

Figure 5: Convergence analysis in terms of both objective loss and AUC of name reference "Lei Wang" using our proposed network embedding model for name disambiguation.

task of name disambiguation by incrementally adding the components in the network embedding model. Specifically, we first rank each individual component by its disambiguation performance in terms of Macro-F1, then add the components one by one in the order of their disambiguation power. In particular, we first add person-document graph, followed by linked document graph, and person-person graph. Figure 4 shows the name disambiguation performance in terms of Macro-F1 value using our proposed network embedding model with different component combinations. As we see from the figure, after adding each component, we observe improvements for both datasets, in which the results are averaged out over all the 10 name references.

5.8 Convergence Analysis

We further investigate the convergence of proposed network embedding algorithm shown in Section 4. Figure 5 shows the convergence analysis of our method under the name reference "Lei Wang" from Arnetminer. For each epoch, we sample $\left(|E_{pp}| + \left|E_{pd}\right| + |E_{dd}|\right)$ training instances to update the corresponding model embedding vectors. We can observe that our proposed network embedding approach converges approximately within 50 epochs and achieves promising convergence results on both pairwise ranking based objective loss and AUC. However, as shown in Equation 7, the objective function in our proposed embedding model is not convex, thus reaching global optimal solution using SGD based optimization technique is a fairly challenging task. The possible remedy could be to decrease the learning rate α in SGD when number of epochs increases. Another strategy is to try multiple runs with different seeds initialization. Similar convergence patterns are observed for other name references as well.

6 CONCLUSION

To conclude, in this paper we propose a novel representation learning based solution to address the name disambiguation problem. Our proposed representation learning model uses a pairwise ranking objective function which clusters the documents belonging to a single person better than other existing network embedding methods. Besides, the proposed solution uses only the relational data, so it is particularly useful for name disambiguation in anonymized network, where node attributes are not available due to the privacy concern. Our experimental results on multiple datasets show that our proposed method significantly outperforms many of the existing state-of-the-arts for name disambiguation.

REFERENCES

[1] Razvan Bunescu and Marius Pasca. 2006. Using Encyclopedic Knowledge for Named Entity Disambiguation. In *ACL*. 9–16.

[2] Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2015. GraRep: Learning Graph Representations with Global Structural Information. In *CIKM*. 891–900.

[3] Lei Cen, Eduard C. Dragut, Luo Si, and Mourad Ouzzani. 2013. Author Disambiguation by Hierarchical Agglomerative Clustering with Adaptive Stopping Criterion. In *SIGIR*. 741–744.

[4] Shiyu Chang, Wei Han, Jiliang Tang, Guo-Jun Qi, Charu C. Aggarwal, and Thomas S. Huang. 2015. Heterogeneous Network Embedding via Deep Architectures. In *SIGKDD*. 119–128.

[5] P. Y. Chen, S. Choudhury, and A. O. Hero. 2016. Multi-centrality graph spectral decompositions and their application to cyber intrusion detection. In *ICASSP'16*.

[6] Pin-Yu Chen, Baichuan Zhang, Mohammad Al Hasan, and Alfred O Hero. 2016. Incremental Method for Spectral Clustering of Increasing Orders. In *KDD Workshop on Mining and Learning with Graphs*.

[7] Ting Chen and Yizhou Sun. Task-Guided and Path-Augmented Heterogeneous Network Embedding for Author Identification. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM'17)*. 295–304.

[8] Sutanay Choudhury, Khushbu Agarwal, Sumit Purohit, Baichuan Zhang, Meg Pirrung, Will Smith, and Mathew Thomas. 2017. NOUS: Construction and Querying of Dynamic Knowledge Graphs. In *8th International Workshop on Data Engineering meets the Semantic Web (DESWeb) under ICDE 2017*. IEEE.

[9] Aditya Grover and Jure Leskovec. 2016. Node2Vec: Scalable Feature Learning for Networks. In *SIGKDD*. 855–864.

[10] Hui Han, Lee Giles, Hongyuan Zha, Cheng Li, and Kostas Tsioutsiouliklis. 2004. Two Supervised Learning Approaches for Name Disambiguation in Author Citations. In *Joint Conf. on Digital Libraries*.

[11] Hui Han, Hongyuan Zha, and C. Lee Giles. 2005. Name Disambiguation in Author Citations Using a K-way Spectral Clustering Method. In *ACM Joint Conf. on Digital Libraries*. 334–343.

[12] Xianpei Han, Le Sun, and Jun Zhao. 2011. Collective Entity Linking in Web Text: A Graph-based Method. In *SIGIR*.

[13] Xianpei Han and Jun Zhao. 2009. Named Entity Disambiguation by Leveraging Wikipedia Semantic Knowledge. In *CIKM*. 215–224.

[14] Linus Hermansson, Tommi Kerola, Fredrik Johansson, Vinay Jethava, and Devdatt Dubhashi. 2013. Entity Disambiguation in Anonymized Graphs Using Graph Kernels. In *CIKM*. 1037–1046.

[15] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust Disambiguation of Named Entities in Text. In *EMNLP*.

[16] Da Kuang, Haesun Park, and Chris H. Q. Ding. 2012. Symmetric Non-negative Matrix Factorization for Graph Clustering. In *SDM*. 106–117.

[17] Bradley Malin. Unsupervised name disambiguation via social network similarity. In *SDM'05 Workshop on Link Analysis, Counterterrorism, and Security*. 93–102.

[18] Tomas Mikolov, Ilya Sutskever, K Chen, G S Corrado, and Jeff Dean. Distributed Representations of Words and Phrases and their Compositionality. In *NIPS'13*.

[19] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: Online Learning of Social Representations. In *SIGKDD*. 701–710.

[20] Tanay Kumar Saha, Baichuan Zhang, and Mohammad Al Hasan. 2015. Name disambiguation from link data in a collaboration graph using temporal and topological features. *Social Network Analysis and Mining* (2015), 1–14.

[21] Yang Song, Jian Huang, Isaac G. Councill, Jia Li, and C. Lee Giles. 2007. Efficient Topic-based Unsupervised Name Disambiguation. In *JCDL*. 342–351.

[22] Jie Tang, Alvis C. M. Fong, Bo Wang, and Jing Zhang. 2012. A Unified Probabilistic Framework for Name Disambiguation in Digital Library. *IEEE TKDE* (2012).

[23] Jian Tang, Meng Qu, and Qiaozhu Mei. 2015. PTE: Predictive Text Embedding Through Large-scale Heterogeneous Text Networks. In *SIGKDD*. 1165–1174.

[24] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale Information Network Embedding. In *WWW*. 1067–1077.

[25] Cunchao Tu, Han Liu, Zhiyuan Liu, and Maosong Sun. 2017. CANE: Context-Aware Network Embedding for Relation Modeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 1722–1731.

[26] Suhang Wang, Jiliang Tang, Charu Aggarwal, and Huan Liu. 2016. Linked Document Embedding for Classification. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM '16)*. 115–124.

[27] Xuezhi Wang, Jie Tang, Hong Cheng, and Philip S. Yu. 2011. ADANA: Active Name Disambiguation. In *ICDM*. 794–803.
[28] Mohammed J. Zaki and Wagner Meira Jr. 2014. *Data Mining and Analysis: Fundamental Concepts and Algorithms*. Cambridge University Press.
[29] Baichuan Zhang, Sutanay Choudhury, Mohammad Al Hasan, Xia Ning, Khushbu Agarwal, Sumit Purohit, and Paola Gabriela Pesntez Cabrera. 2016. Trust from the past: Bayesian Personalized Ranking based Link Prediction in Knowledge Graphs. In *SDM Workshop on Mining Networks and Graphs*.
[30] Baichuan Zhang, Murat Dundar, and Mohammad Al Hasan. 2016. Bayesian Non-Exhaustive Classification A Case Study: Online Name Disambiguation using Temporal Record Streams. In *CIKM'2016*. 1341–1350.
[31] Baichuan Zhang, Noman Mohammed, Vachik S. Dave, and Mohammad Al Hasan. 2017. Feature Selection for Classification under Anonymity Constraint. *Transactions on Data Privacy* 10, 1 (2017), 1–25.
[32] Baichuan Zhang, Tanay Kumar Saha, and Mohammad Al Hasan. Name disambiguation from link data in a collaboration graph. In *ASONAM'14*. 81–84.
[33] Duo Zhang, Jie Tang, Juanzi Li, and Kehong Wang. 2007. A Constraint-based Probabilistic Framework for Name Disambiguation. In *CIKM 07*. 1019–1022.