# SHINE+: A General Framework for Domain-Specific Entity Linking with Heterogeneous Information Networks

Wei Shen, Jiawei Han, *Fellow, IEEE*, Jianyong Wang, *Fellow, IEEE*, Xiaojie Yuan, and Zhenglu Yang

**Abstract**—Heterogeneous information networks that consist of multi-type, interconnected objects are becoming increasingly popular, such as social media networks and bibliographic networks. The task of linking named entity mentions detected from unstructured Web text with their corresponding entities in a heterogeneous information network is of practical importance for the problem of information network population. This task is challenging due to name ambiguity and limited knowledge existing in the network. Most existing entity linking methods focus on linking entities with Wikipedia and cannot be applied to our task. In this paper, we present SHINE+, a general framework for linking named entitieS in Web free text with a Heterogeneous Information NEtwork. We propose a probabilistic linking model, which unifies an entity popularity model with an entity object model. As the entity knowledge contained in the information network is insufficient, we propose a knowledge population algorithm to iteratively enrich the network entity knowledge by leveraging the context information of mentions mapped by the linking model with high confidence, which subsequently boosts the linking performance. Experimental results over two real heterogeneous information networks (i.e., DBLP and IMDb) demonstrate the effectiveness and efficiency of our proposed framework in comparison with the baselines.

**Index Terms**—Entity linking, heterogeneous information network, probabilistic linking model, knowledge population algorithm

## 1 Introduction

Heterogeneous information networks (HIN) that involve a large number of multi-type objects are becoming ubiquitous and prevalent, since real world physical and abstract data objects are all connected via different relations, forming diverse heterogeneous information networks [1]. For example, in a bibliographic dataset, objects of multiple types, such as papers (P), authors (A), publication venues (V), and title terms (T), and relations of multiple types, such as *write*, *publish*, and *contain* are interconnected together, providing rich information and forming a heterogeneous information network. However, object names in a heterogeneous network are potentially ambiguous: the same textual name may refer to several different entities. As the example shown in Fig. 1, in the DBLP network, the object name "Wei Wang" may refer to 119 different authors, including "Wei Wang" at University at Albany, SUNY, "Wei Wang" at Fudan Univ., China, "Wei Wang" at UCLA, and "Wei Wang" at UNSW, Australia. In the IMDb network, the object name "Chris Evans" can refer to an American actor known for his superhero role Captain America, a famous English presenter, or some other actors named "Chris Evans".

Although there are many large-scale heterogeneous networks in existence, information contained in them is limited. For example, there does not exist the *advisor* relation between authors in the DBLP network. Furthermore, as the world evolves, new facts come into existence and are digitally expressed on the Web. Therefore, populating the existing heterogeneous information networks with the newly extracted facts (such as relations between entities) becomes increasingly important. However, integrating the newly extracted facts derived from the information extraction systems into an existing heterogeneous information network inevitably needs a system to map the entity mentions associated with the extracted facts to their corresponding entities in the heterogeneous information network. For instance, we could extract the *graduateFrom* relation between the author name "Wei Wang" and the organization name "UCLA" from the Web document in Fig. 1. Before populating this relation into the DBLP network, we need to map the author name "Wei Wang" in this relation to its true mapping author (i.e., "Wei Wang" at UCLA) as there are 119 different authors having the same name "Wei Wang" in the DBLP network.

On the other hand, to some extent, some heterogeneous information networks could be regarded as domain-specific knowledge bases [2]. For example, the DBLP (or IMDb) network contains more interesting and diverse knowledge than Wikipedia with respect to the domain of computer science (or entertainment). In this case, we could regard this task as one type of *domain-specific entity linking*. In our task, we focus on linking entity mentions appearing in the domain-specific unstructured Web text, which pertains to

- *W. Shen, X. Yuan, and Z. Yang are with the College of Computer and Control Engineering, Nankai University, Tianjin 300071, China. E-mail: {shenwei, yuanxj, yangzl}@nankai.edu.cn.*
- *J. Han is with the Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801. E-mail: hanj@cs.uiuc.edu.*
- *J. Wang is with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China, and the Jiangsu Collaborative Innovation Center for Language Ability, Jiangsu Normal University, Xuzhou, Jiangsu 221009, China. E-mail: jianyong@tsinghua.edu.cn.*

*Manuscript received 7 June 2016; revised 29 Apr. 2017; accepted 5 July 2017. Date of publication 24 July 2017; date of current version 9 Jan. 2018. (Corresponding author: Jianyong Wang.)*
*Recommended for acceptance by T. Palpanas.*
*For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.*
*Digital Object Identifier no. 10.1109/TKDE.2017.2730862*

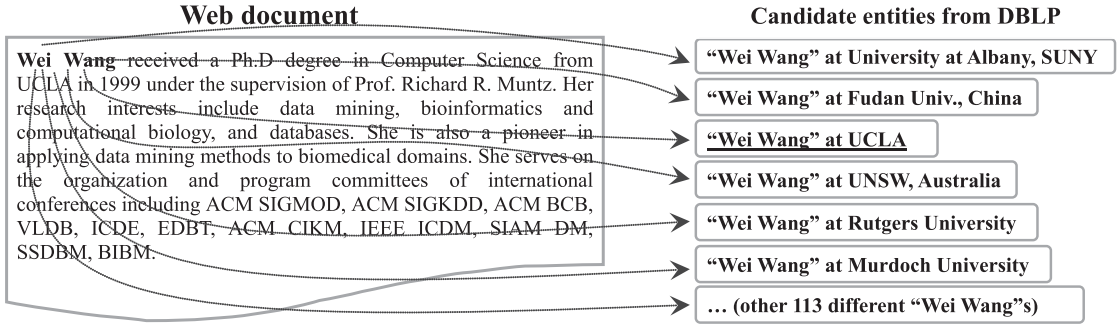



Fig. 1. An illustration for the task of linking an entity mention in Web document with the DBLP bibliographic network. Named entity mention detected from the Web document is in bold face; candidate entities from the DBLP network are shown on the right; true mapping entity is underlined.

the same domain as the heterogeneous information network. Therefore, this task is beneficial for bridging the unstructured documents and the semi-structured heterogeneous information networks, which can facilitate many tasks such as information retrieval and question answering. Most question answering systems leverage their supported knowledge bases to give the answer to the user's question. To answer the question such as "How many papers has Prof. Wei Wang at UCLA published in SIGMOD?", the system should first leverage the entity linking technique to map the queried "Wei Wang" to the professor at UCLA, instead of, for example, the professor at UNSW, Australia; and then it retrieves the number of her SIGMOD papers from the DBLP network directly.

Traditional entity linking methods mainly focus on linking entity mentions in text with their corresponding entities in Wikipedia or Wikipedia-derived knowledge bases (e.g., YAGO [3]), and are largely dependent on the special features associated with Wikipedia [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16]. Specifically, they rely on the context knowledge embedded in the Wikipedia article [4], [5], [6], [9], [10], [11], [12], [13], [14], Wikipedia-based semantic relatedness measures [6], [7], [8], [10], [11], [12], [16] (e.g., Wikipedia Link-based Measure [17]), and some special structures in Wikipedia [4], [5], [6], [8], [9], [11], [13], [14], [16] (e.g., disambiguation page and hyperlink in the Wikipedia article). In this paper, we instead study the problem of linking entities in Web text with a heterogeneous information network. Heterogeneous information networks do not have these specific features associated with Wikipedia. Thus, these traditional entity linking methods cannot be applied to our task. For example, an essential step in these previous approaches [4], [5], [6], [7], [9], [11], [13], [14] is to define a context similarity measure between the Wikipedia article associated with the candidate entity and the text around the entity mention, while for each author entity in the DBLP network, we do not have her descriptive article and cannot calculate the context similarity measure. In addition, the Wikipedia Link-based Measure [17] has been utilized to calculate the topical coherence between mapping entities in many existing entity linking methods [6], [7], [8], [10], [11], [12], [16]. However, this measure is based on the hyperlink structure among Wikipedia articles and cannot be used to calculate the topical coherence between entities in the DBLP or IMDb network.

To deal with this problem, we propose a probabilistic linking model, which combines an entity popularity model with an entity object model. The entity popularity model is context-independent and captures the popularity of an entity. For example, a famous professor named "Wei Wang" who has published many papers is usually considered more popular than a student who is also named "Wei Wang" and has published very few papers.

The entity object model captures the probability of multi-type objects appearing in the textual context of an entity. In a heterogeneous information network, multi-type objects are connected via different types of relations or sequences of relations, forming a set of *meta-paths* [18]. A meta-path is a path consisting of a sequence of relations between different object types (i.e., structural path at the meta level). Different meta-paths imply distinct semantic meanings, which may lead to diverse distributions over objects. For example, in a bibliographic network, A-P-A is a meta-path denoting a relation between an author and her coauthor, whereas A-P-V denotes a relation between an author and a venue where her paper is published. Random walks starting from one author along the meta-path A-P-A may generate the distribution of coauthors for that author, while the meta-path A-P-V may lead to the distribution of venues for that author. A question then arises: *which meta-paths are more important for the entity linking task?* The estimation problem of our model is to determine which meta-paths (or their weighted combination) are used for the specific entity linking task. It is difficult to ask a user to explicitly specify the weights for such sophisticated meta-paths. To address this problem, an effective weight learning algorithm is proposed to automatically learn the most appropriate weights of meta-paths based on the expectation-maximization (EM) algorithm without requiring any annotated training data. With regard to different meta-path sets for arbitrary heterogeneous information networks, our probabilistic linking model can automatically learn the proper meta-path weights, which makes our model general and flexible enough to accommodate various types of heterogeneous networks.

As stated above, the entity knowledge contained in the existing information network is limited. In some cases, the information network cannot provide enough useful knowledge to help link entity mentions correctly. For example, when the entity mention "Ke Chen" appears in the text "Ke Chen from Liverpool University is giving a talk to the students.", the existing entity linking methods cannot link it with DBLP correctly, because the DBLP network does not have the affiliated institution information for authors. To address this problem, we propose a knowledge population algorithm to iteratively enrich the network entity knowledge by leveraging the context information extracted from the text where linked mentions with high confidence appear. Subsequently, the linking model could leverage the enriched entity knowledge to link entity mentions

more accurately. For example, there is another entity mention "Ke Chen" in some text that has been linked by the linking model with high confidence. In its surrounding context, "Liverpool University" is extracted by the knowledge population algorithm to augment its corresponding entity knowledge in the DBLP network. Then this enriched entity knowledge could be leveraged by the linking model to link the aforementioned "Ke Chen" correctly.

This knowledge population algorithm performs entity linking and entity knowledge population jointly, and makes these two tasks mutually reinforce each other. So far, these two tasks have been investigated separately. The experimental results introduced in Section 5 verify that this algorithm boosts the entity linking accuracy significantly. Moreover, the idea of the knowledge population algorithm could be applied to other entity linking tasks or models, and might motivate further research on combining these two tasks to obtain more significant and interesting achievements.

*Contributions.* The main contributions of this paper are summarized as follows.

- We are among the first to explore the problem of linking entities with a heterogeneous information network, and propose a general unsupervised framework SHINE+ to address this problem effectively.
- We propose a probabilistic linking model, which unifies an entity popularity model with an entity object model. To solve the model estimation problem, a weight learning algorithm is proposed to learn the meta-path weights based on the EM algorithm without requiring any annotated training data.
- A knowledge population algorithm is proposed to iteratively enrich the network entity knowledge by leveraging the context of mentions mapped by the linking model with high confidence. This algorithm performs entity linking and entity knowledge population jointly, and makes them mutually reinforce each other.
- To verify the effectiveness and efficiency of SHINE+, we conducted experiments over two real heterogeneous information networks (i.e., DBLP and IMDb) and three manually annotated Web document collections. The experimental results show that SHINE+ significantly outperforms the baselines in terms of accuracy, and scales very well.

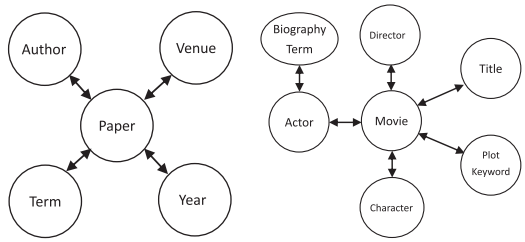
The remainder of this paper is organized as follows. Section 2 introduces some background concepts and the formal notation used throughout the paper. We present the probabilistic linking model in Section 3, and introduce the knowledge population algorithm in Section 4. Section 5 presents the experimental results and Section 6 discusses the related work. Finally, we conclude this paper in Section 7.

## 2 Preliminaries and Notation

In this section, we begin by introducing some concepts in heterogeneous information networks. Next, we define the task of linking entities in Web text with a heterogeneous information network (entity linking with a HIN for short).

### 2.1 Heterogeneous Information Network

A heterogeneous information network $G$ is an information network with multiple types of objects and multiple types of links [1], [18].




(a) DBLP (b) IMDb

Fig. 2. The DBLP and IMDb network schemas.

**Definition 1 (Heterogeneous information network).** *A heterogeneous information network is defined as a directed graph $G = (V, Z)$, where $V$ is the object set and $Z$ is the link set. Each object $v \in V$ belongs to a particular object type $T$, and each link $z \in Z$ belongs to a particular relation type $R$. Moreover, the number of object types $|\{T\}| > 1$ and the number of relation types $|\{R\}| > 1$.*

The DBLP bibliographic network[1] is a typical heterogeneous information network, containing five types of objects: papers (P), authors (A), publication venues (V), title terms (T), and publication years (Y). Links exist between authors and papers by the relations *write* and $write^{-1}$, between publication venues and papers by *publish* and $publish^{-1}$, between papers and title terms by *contain* and $contain^{-1}$, and between papers and publication years by *publishedIn* and $publishedIn^{-1}$. Network schema (i.e., meta-level description for the network) for the DBLP network is shown in Fig. 2a. The IMDb network[2] (see its schema in Fig. 2b) is also a heterogeneous information network, containing seven types of objects: actors (Ac), biography terms (B), movies (Mv), movie titles (MT), plot keywords (K), characters (C), and directors (Di). In a heterogeneous information network, two objects can be connected via different types of relations or sequences of relations, forming a set of meta-paths, which are defined as follows.

**Definition 2 (Meta-path).** *A meta-path $p$ is a path defined over the network schema of a given network $G$, and is denoted in the form of $T_1 \xrightarrow{R_1} T_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} T_{l+1} (l \geq 1)$, which defines a composite relation $R_1 \circ R_2 \circ \cdots \circ R_l$ between object types $T_1$ and $T_{l+1}$, where $\circ$ denotes the composition operator on relations.*

Meta-path $p$ can be also described as a sequence of relations (denoted by $R_1 - R_2 - \cdots - R_l$), or a sequence of object types (denoted by $T_1 - T_2 - \cdots - T_{l+1}$) if there exist no multiple relations between the same pair of object types for simplicity. The length of meta-path $p$ is the number of relations in $p$. For example, in the IMDb network, Ac-B is a length-1 meta-path denoting a relation between an actor and a biography term she has, and Ac-Mv-Di is a length-2 meta-path denoting a relation between an actor and a director who directs the movie she performs.

### 2.2 Entity Linking with a HIN

According to the task setting, we take (1) a collection of unstructured Web documents (denoted by $D$), (2) named entity mentions recognized in the given documents $D$ (denoted by $M$), and (3) a heterogeneous information

1. http://www.dblp.org/
2. http://www.imdb.com/

network $G$ as input. Each Web document $d \in D$ should pertain to the same domain as the heterogeneous information network $G$; otherwise, the document $d$ does not have any common knowledge with the information network, which makes entity linking meaningless. For example, if we link with the DBLP network, the Web document $d \in D$ should pertain to the domain of computer science (CS), such as CS researcher's homepage, news article in CS department website, CS talk/seminar introduction page, etc. Each entity mention $m \in M$ detected from document $d$ is a token sequence (or surface form) of a named entity that is potentially linked with an entity in the heterogeneous network $G$. $E$ is the set of entities in the heterogeneous network $G$ which have the same object type as the type of entity mentions $M$. Each entity in $E$ is denoted by $e$. Generally, the entity set $E$ is a subset of the object set $V$ in the network $G$ (i.e., $E \subset V$). For example, if we want to link author name mentions in Web text with the DBLP network, the entity set $E$ should be the object set of author type in the DBLP network. Here, we formally state the task of entity linking with a HIN as follows.

**Definition 3 (Entity linking with a HIN).** *Given a named entity mention set $M$ detected from a Web document collection $D$ and a heterogeneous information network $G$, the goal is to identify the mapping entity $e \in E$ in the heterogeneous information network $G$ for each entity mention $m \in M$ in a document $d \in D$.*

For illustration, we show a running example of the task of entity linking with a HIN.

**Example 1 (Entity linking with a HIN).** In this example, we consider the task of entity linking with a HIN shown in Fig. 1. Named entity mention "Wei Wang" in the Web document of Fig. 1 needs to be linked with its referring author in the DBLP bibliographic network. There are totally 119 different candidate author entities in the DBLP network according to Fig. 1. For the named entity mention "Wei Wang" in this example, we should output its true mapping author entity (i.e., "Wei Wang" at UCLA), which is underlined in Fig. 1.

In this paper, due to limited scope we assume that the heterogeneous information network $G$ contains all the mapping entities for all the named entity mentions $M$.

## 3 The Probabilistic Linking Model

In this section, we propose a probabilistic linking model to deal with the task of entity linking with a HIN. Given a named entity mention $m \in M$ detected from a document $d \in D$, we want to find its most likely mapping entity $e \in E$ in the heterogeneous information network $G$. This leads to the following inference problem.

**Problem 1 (Inference).** *Given a named entity mention $m$ appearing in a document $d$, compute*

$$\arg\max_{e \in E} \mathbf{P}(e|m, d), \tag{1}$$

*i.e., the most likely mapping entity $e$ given an entity mention $m$ in a document $d$.*

According to Formula (1), given a named entity mention $m$ in $d$, we could find its mapping entity $e$ as follows:

$$\arg\max_{e \in E} \mathbf{P}(e|m, d) = \arg\max_{e \in E} \frac{\mathbf{P}(m, d, e)}{\mathbf{P}(m, d)} = \arg\max_{e \in E} \mathbf{P}(m, d, e). \tag{2}$$

We assume that there is an underlying distribution $\mathbf{P}$ over the set $M \times D \times E$. Therefore, our goal is to model $\mathbf{P}(m, d, e)$. The probability of an entity mention $m$ whose context is the document $d$ referring to a specific entity $e$ could be expressed as the following formula (here we assume that $m$ and $d$ are independent given $e$):

$$\mathbf{P}(m, d, e) = \mathbf{P}(e)\mathbf{P}(m|e)\mathbf{P}(d|e). \tag{3}$$

The mapping entity $e$ should have the name of surface form $m$ and we denote entities that could be referred by the name $m$ as the candidate entities for entity mention $m$. For simplicity, we assume that the probability $\mathbf{P}(m|e)$ of observing the name $m$ given each candidate entity $e$ for mention $m$ is the same and defined as a constant $\eta$ where $0 < \eta \leq 1$. For example, given each of the 119 author entities named "Wei Wang" shown in Fig. 1, we assume that the likelihood of observing "Wei Wang" as her name is the same. Under this reasonable assumption, the complete model can be expressed as

$$\mathbf{P}(m, d, e) = \eta \cdot \mathbf{P}(e)\mathbf{P}(d|e), \tag{4}$$

where $e$ is a candidate entity for entity mention $m$. This probabilistic linking model as shown in Formula (4) mainly consists of two components:

(1) The entity popularity model $\mathbf{P}(e)$ captures the popularity of an entity $e$, which is the likelihood of observing an entity $e$ appearing in a document without knowing any context information.
(2) The entity object model $\mathbf{P}(d|e)$ denotes the probability of observing document $d$ as the textual context for entity $e$.

In the following, we present the entity popularity model in Section 3.1 and the entity object model in Section 3.2. We introduce the model estimation method in Section 3.3.

### 3.1 The Entity Popularity Model

We have the observation that each entity in the heterogeneous information network has different popularity. Some entities in the heterogeneous information network are obviously more prevalent than others. For example, a professor named "Rakesh Kumar" who has published many papers is usually regarded more popular than a Ph.D student who has published very few papers and has the same name "Rakesh Kumar".

Most previous entity linking systems estimate the popularity of an entity using the entity frequency in the Wikipedia article corpus [6], [7], [8], [9], [11]. However, this approach cannot be applied to our task of entity linking with a HIN. Entities in the heterogeneous information network are connected via different relations, and the popularity of an entity in the network relies on the visibility of other connected entities. For example, in the DBLP bibliographic network, the popularity of an author depends on some features, such as her coauthors' popularity, her publication quantity, and the authoritativeness of the venues where her papers are published.

TABLE 1
The Entity Popularity in Example 1

| Candidate entity | Entity popularity |
|---|---|
| "Wei Wang" at University at Albany, SUNY | $4.441*10^{-6}$ |
| "Wei Wang" at Fudan Univ., China | $7.373*10^{-6}$ |
| "Wei Wang" at UCLA | $1.08*10^{-5}$ |
| "Wei Wang" at UNSW, Australia | $6.202*10^{-6}$ |
| "Wei Wang" at Rutgers University | $7.675*10^{-7}$ |
| "Wei Wang" at Murdoch University | $4.180*10^{-7}$ |

As we know, PageRank [19] is a general-purpose network node importance measure which is fairly successful for many tasks. Here, we utilize the PageRank score of an entity in the network to indicate its popularity. For simplicity, we ignore the object types in the network $G$ when computing the PageRank score offline (the detailed method for computing PageRank can be seen in our previous paper [20]). As PageRank algorithm is computed over the entire object set $V$ in the network $G$, we focus on the popularity of entities in $E$, which is a subset of $V$. For each entity $e \in E$, let $pr(e)$ be its PageRank score. Our entity popularity model estimates the popularity $\mathbf{P}(e)$ of entity $e$ as follows:

$$\mathbf{P}(e) = \frac{pr(e)}{\sum_{e' \in E} pr(e')}. \quad (5)$$

For illustration, we show in Table 1 the entity popularity $\mathbf{P}(e)$ for each candidate entity $e$ in Example 1. From the results in Table 1, we can see that the popularity of the author entity "Wei Wang" at UCLA (i.e., $1.08*10^{-5}$) is the highest among all the candidates, which demonstrates that the author "Wei Wang" at UCLA is the most popular entity in the candidate entity set with respect to the entity mention "Wei Wang" that is consistent with our intuition, while the author "Wei Wang" at Murdoch University who has just published one paper in DBLP has the lowest entity popularity (i.e., $4.180*10^{-7}$). It can be seen that the entity popularity model suitably expresses the popularity of the candidate entity.

### 3.2 The Entity Object Model

The entity object model $\mathbf{P}(d|e)$ captures the probability of observing document $d$ as the textual context for entity $e$. That is to say, it will assign a high probability if the entity $e$ frequently appears in the context of document $d$, and will assign a low probability if the entity $e$ rarely appears in the context of document $d$.

Since we are dealing with heterogeneous information networks which involve a large number of multi-type objects, we assume the document $d$ consists of various multi-type objects $v$'s from the heterogeneous information network and the observation of these different objects given the entity is independent. In Example 1, the Web document where the entity mention "Wei Wang" appears as shown in Fig. 1 consists of an object of author type (i.e., *Richard R. Muntz*), some objects of venue type (such as *SIGMOD*, *SIGKDD*, *BCB*, *VLDB*, etc.), some objects of term type (such as *computer*, *data*, *mining*, *bioinformatics*, *computational*, etc.), and an object of year type (i.e., *1999*). The approach to recognizing multi-type objects in the document is introduced in Section 5.1.

Then the entity object model $\mathbf{P}(d|e)$ can be expressed as the product of the probabilities $\mathbf{P}(v|e)$ under the assumption that the document $d$ is composed of various multi-type objects $v$'s from the heterogeneous information network and the observation of these different objects $v$'s given the entity $e$ is independent, which is similar to unigram language modeling [21]. Thus we have

$$\mathbf{P}(d|e) = \prod_{v \in d} \mathbf{P}(v|e). \quad (6)$$

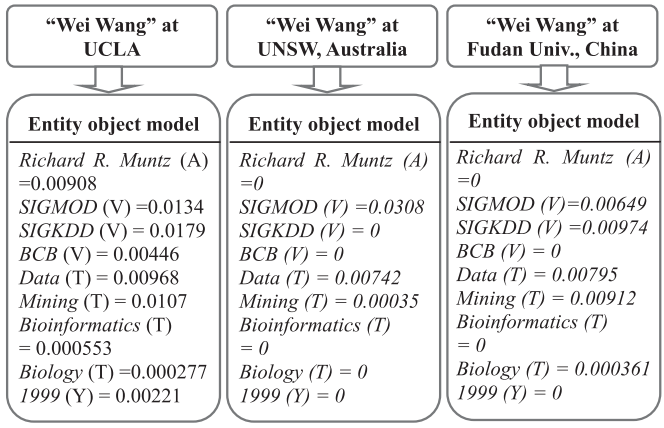



Fig. 3. The entity object model for three candidate entities in Example 1. The letter in parentheses after each object represents its object type.

From Formula (6), we can see that the entity object model captures the probability of multi-type objects $v$'s appearing in the textual context of entity $e$. The distribution $\mathbf{P}(v|e)$ encodes the probability of observing object $v$ given entity $e$, which can be estimated from the associated network about entity $e$ in the heterogeneous information network. For example, with respect to the entity "Wei Wang" at UCLA, the probability of observing venue object *SIGMOD* should be higher than the probability of observing venue object *VLDB*, because the author "Wei Wang" at UCLA has published much more papers in the *SIGMOD* conference (i.e., six papers) than the *VLDB* conference (i.e., one paper) in DBLP.

Fig. 3 shows parts of the entity object model for three candidate entities with the highest entity popularity in Example 1 (i.e., "Wei Wang" at UCLA, "Wei Wang" at UNSW, Australia, and "Wei Wang" at Fudan Univ., China), which is generated using *meta-path constrained random walks* (its definition is given in Formula (9)). From Fig. 3, we can see that the probability $\mathbf{P}(d|$"Wei Wang" at UCLA$)$ of observing the Web document $d$ in Fig. 1 given the entity "Wei Wang" at UCLA is likely to be significantly higher than the probability $\mathbf{P}(d|$"Wei Wang" at Fudan Univ., China$)$ and the probability $\mathbf{P}(d|$"Wei Wang" at UNSW, Australia$)$, because the probabilities of observing most of the representative objects appearing in the document $d$ (e.g., author object *Richard R. Muntz*, venue objects *SIGKDD* and *BCB*, term objects *Data*, *Mining*, *Bioinformatics*, and year object *1999*) given the entity "Wei Wang" at UCLA are higher than the probabilities of observing these objects given each of the other two candidate entities.

From Fig. 3, we can also see that the probability of observing some object given an entity is equal to 0 (e.g., the probability of observing author object *Richard R. Muntz* given the entity "Wei Wang" at Fudan Univ., China, $\mathbf{P}($Richard R. Muntz $|$"Wei Wang" at Fudan Univ., China$)$) due to the sparse data problem. This leads to that the product of probabilities in Formula (6) equals zero. To avoid this problem, we further smooth $\mathbf{P}(v|e)$ using a generic object model for the domain.

Formally, given a document $d$ that is the textual context for entity $e$, each object $v \in d$ is drawn randomly from a mixture of two object models: an entity-specific object model $\mathbf{P_e}(v)$ which is a distribution over objects with respect to entity $e$ and can be generated using *meta-path constrained random walks* (its definition is given in Formula (9)), and a generic object model for the domain $\mathbf{P_g}(v)$ which is independent of entity $e$ and can be estimated from the whole collection. Thus, we could further define the entity object model $\mathbf{P}(d|e)$ as

$$\mathbf{P}(d|e) = \prod_{v \in d} (\theta \cdot \mathbf{P_e}(v) + (1 - \theta) \cdot \mathbf{P_g}(v)), \tag{7}$$

where $\theta \in (0, 1)$ is a parameter that balances the two parts (i.e., the entity-specific object model $\mathbf{P_e}(v)$ and the generic object model for the domain $\mathbf{P_g}(v)$). The generic object model for the domain $\mathbf{P_g}(v)$ can be learned by counting the frequencies of multi-type objects appearing in the document collection $D$. The approach to recognizing multi-type objects in the document is introduced in Section 5.1.

In a heterogeneous information network, an object could link to many different types of objects by multiple meta-paths. Different meta-paths imply different semantic meanings, which may lead to rather diverse distributions over objects. Thus, we explore meta-paths to guide the random walks over the heterogeneous network $G$. In this paper, we propose to use *meta-path constrained random walks* [22] to estimate the entity-specific object model $\mathbf{P_e}(v)$. Formally, let meta-path $p = R_1 - R_2 - \cdots - R_l$, and each relation $R_k$ be a binary relation. We define $R_k(v', v) = 1$ if object $v'$ and object $v$ are linked by relation $R_k$, and $R_k(v', v) = 0$ otherwise. We also define $R_k(v') = \{v | R_k(v', v)\}$, which is the set of objects that are linked with object $v'$ via the relation $R_k$. Given the meta-path $p = R_1 - R_2 - \cdots - R_l$ which starts with the same object type as entity $e$, we define $\mathbf{P_e}(v|p)$, i.e., the distribution of observing object $v$ given entity $e$ and meta-path $p$, as follows. First, if meta-path $p$ is an empty path, we define

$$\mathbf{P_e}(v|p) = \begin{cases} 1 & \text{if object } v \text{ is entity } e, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

If $p = R_1 - R_2 - \cdots - R_l$ is a nonempty path, then let $p' = R_1 - R_2 - \cdots - R_{l-1}$, and define

$$\mathbf{P_e}(v|p) = \sum_{v' \in V} \mathbf{P_e}(v'|p') \cdot \frac{R_l(v', v)}{|R_l(v')|}, \tag{9}$$

where $|R_l(v')|$ is the number of objects that are linked with the object $v'$ via the relation $R_l$. This definition (Formula (9)) is in a recursive form and is called *meta-path constrained random walks*, i.e., random walks starting from entity $e$ along meta-path $p$. Given each meta-path, we could calculate the distribution of observing objects for each entity using Formula (9).

For example, given the meta-path A-P-V in DBLP, with respect to the entity "Wei Wang" at UCLA, the probability of observing venue object *SIGMOD* is 0.0536, while the probability of observing venue object *VLDB* or venue object *SIGMETRICS* is the same (i.e., 0.00893) in the DBLP network, because the author "Wei Wang" at UCLA has published six papers in the *SIGMOD* conference, and has published just one paper in the *VLDB* conference and the *SIGMETRICS* conference respectively, from the DBLP network. Additionally, given the meta-path A-P-A-P-V, with respect to the entity "Wei Wang" at UCLA, the probability of observing venue object *VLDB* (i.e., 0.00863) is much higher than the probability of observing venue object *SIGMETRICS* (i.e., 0.00471) in DBLP, since the coauthors of author "Wei Wang" at UCLA have published much more papers in the *VLDB* conference than the *SIGMETRICS* conference. It can be seen that different meta-paths may imply different semantic meanings, which may lead to rather diverse distributions over objects.

Therefore it is desirable to learn the relative importance for each meta-path for the specific entity linking task. In order to quantify the importance for each meta-path $p$, we give a meta-path weight $w_p$ for each meta-path $p$. Given a set of meta-paths, the entity-specific object model $\mathbf{P_e}(v)$ could be the weighted sum of the probabilities of observing object $v$ given entity $e$ along each meta-path $p$. We define the entity-specific object model $\mathbf{P_e}(v)$ as follows:

$$\mathbf{P_e}(v) = \sum_p w_p \mathbf{P_e}(v|p), \tag{10}$$

where $\sum_p w_p = 1$. A larger $w_p$ indicates a higher importance for the meta-path $p$ with respect to the entity linking task. We define the meta-path weight vector as $\overrightarrow{W}$, in which each item $w_p$ is the weight for meta-path $p$. Note that, we do not consider negative $w_p$ in this model, which means relationships with a negative impact to the entity linking process are not considered, and the extreme case of $w_p = 0$ means the relationships in this meta-path are totally irrelevant to the entity linking process.

A set of meta-paths starting from the same object type as entity $e$, which might be useful for the entity linking task, should be provided as the input of this model. We define this input set of meta-paths as $MP$. These meta-paths could be determined either according to users' expert knowledge, or by traversing the network schema starting from the same object type as entity $e$ with a length constraint using standard traversal methods such as the BFS (breadth-first search) algorithm. A very long meta path that will propagate relationships to remote neighborhoods may not carry much meaningful semantic meaning [18] and is not very useful in entity linking. The meta-paths we use in our experiments are shown in Section 5.2.2.

Note that the meta-path weights $w_p$'s for each meta-path $p$ in the meta-path set $MP$ are the only parameters which need to be learned in our model. The estimation problem of our model could be solved as follows.

**Problem 2 (Estimation).** *Given a heterogeneous information network $G$ and a set of named entity mentions $M$ recognized in the given document collection $D$, determine parameters (i.e., meta-path weights $w_p$'s for each meta-path $p$) that maximize the likelihood of observing named entity mentions $M$ in the document collection $D$.*

Once we learn the model, we could use it to link entity mentions with a heterogeneous information network according to Formula (2). In the following section, we introduce the model estimation method (i.e., the weight learning algorithm).

### 3.3 The Weight Learning Algorithm

Given a set of named entity mentions $M$ recognized in the given document collection $D$, we want to estimate parameters (i.e., meta-path weights $w_p$'s for each meta-path $p$) that

maximize the likelihood of observing these named entity mentions $M$ in the document collection $D$. Thus, we want

$$\arg\max_{w_p} \prod_{(m,d)} \mathbf{P}(m,d). \tag{11}$$

We have

$$\mathbf{P}(m,d) = \sum_{e \in E} \mathbf{P}(m,d,e), \tag{12}$$

where $\mathbf{P}(m,d,e)$ is given in Formula (4). Then, we have

$$\arg\max_{w_p} \prod_{(m,d)} \sum_{e} \eta \cdot \mathbf{P}(e)\mathbf{P}(d|e) = \arg\max_{w_p} \prod_{(m,d)} \sum_{e} \mathbf{P}(e)\mathbf{P}(d|e). \tag{13}$$

Since this objective function is in a product of sum form that is difficult to optimize directly, we define a hidden random variable $\pi(m,d,e)$ for each triple $(m,d,e)$ to simplify its form as follows:

$$\pi(m,d,e) = \begin{cases} 1 & \text{if mention } m \text{ in } d \text{ refers to } e, \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

Then our optimization function could be written as

$$\arg\max_{w_p} \prod_{(m,d)} \prod_{e} (\mathbf{P}(e)\mathbf{P}(d|e))^{\pi(m,d,e)}. \tag{15}$$

Now we can apply the expectation-maximization (EM) method iteratively to optimize this objective function. In the initialization step, we assume some initial values of the parameters (i.e., give some initial values to the meta-path weight vector $\overrightarrow{W}$).

*E-Step.* In the expectation step, using the current values of the parameters, we could find the expected values of the hidden variables using the following formula:

$$\mathbf{E}(\pi(m,d,e)) = \mathbf{P}(e|m,d) = \frac{\mathbf{P}(m,d,e)}{\mathbf{P}(m,d)} = \frac{\mathbf{P}(m,d,e)}{\sum_{e'} \mathbf{P}(m,d,e')}. \tag{16}$$

As in Formula (4), entity $e$ is defined as the candidate entity for entity mention $m$. Therefore, for each entity mention $m$ in a document $d$, we maintain a candidate entity set (denoted by $E_m$), and assume the probability of linking with other entities to be 0. Thus, this expression can be calculated by iterating over each candidate entity in the candidate entity set $E_m$ for each given mention-document tuple.

*M-Step.* In the maximization step, we use the value $f(m,d,e) = \mathbf{E}(\pi(m,d,e))$ calculated in the E-step, and find the parameters $w_p$'s that maximize the following function:

$$\prod_{(m,d),e} (\mathbf{P}(e)\mathbf{P}(d|e))^{f(m,d,e)} = \prod_{(m,d),e} (\mathbf{P}(e))^{f(m,d,e)} \cdot \prod_{(m,d),e} (\mathbf{P}(d|e))^{f(m,d,e)}. \tag{17}$$

We can see that the first product $\prod_{(m,d),e}(\mathbf{P}(e))^{f(m,d,e)}$ does not involve the parameters $w_p$'s and does not depend on these parameters. Therefore, we just need to find the optimal parameters $w_p$'s that maximize the following function:

$$\prod_{(m,d),e} (\mathbf{P}(d|e))^{f(m,d,e)}. \tag{18}$$

**Algorithm 1.** *The Learning Algorithm*

**Input:** Heterogeneous information network $G$, named entity mentions $M$, document collection $D$, and meta-path set $MP$.
**Output:** The meta-path weight vector $\overrightarrow{W}$.

```
for each meta-path p do
    Initialize the weight w_p = 0
end for
repeat
    E-step: update E(π(m,d,e)) by Formula (16)
    M-step:
    W^0 = W
    t = 1
    repeat
        for each meta-path p do
            Update w_p^(t) by Formula (21)
        end for
        Normalize w_p^(t) to satisfy Σ_p w_p^(t) = 1
        t = t + 1
    until objective function J (Formula (20)) converges
    W = W^(t-1)
until meta-path weight vector W stabilizes within some threshold
```

By obtaining the logarithm of the above objective function, we get the objective function:

$$J = \sum_{(m,d),e} f(m,d,e) ln \mathbf{P}(d|e). \tag{19}$$

By substituting Formulas (7) and (10), the objective function of Formula (19) can be derived as

$$J = \sum_{(m,d),e} f(m,d,e) \sum_{v} ln\left(\theta \cdot \sum_{p} w_p \mathbf{P}_{\mathbf{e}}(v|p) + (1-\theta) \cdot \mathbf{P}_{\mathbf{g}}(v)\right). \tag{20}$$

We use gradient descent approach to solve this optimization problem. The basic idea of gradient descent is to find the direction (gradient) so that the objective function climbs up and makes a small step towards the direction via iteratively updating the meta-path weights $w_p$'s in the vector $\overrightarrow{W}$. Specifically, it is an iterative algorithm with the updating formula as

$$w_p^{(t)} = w_p^{(t-1)} + \alpha \cdot \frac{\partial J}{\partial w_p}|_{w_p = w_p^{(t-1)}}, \tag{21}$$

where $\alpha$ is the learning rate, which decides the step size towards the increasing direction and is usually set to a small enough number to guarantee the increase of the objective function $J$. The partial derivative of $w_p$ can be derived as

$$\frac{\partial J}{\partial w_p} = \sum_{(m,d),e} f(m,d,e) \sum_{v} \frac{\theta * \mathbf{P}_{\mathbf{e}}(v|p)}{\mathbf{P}(v|e)} \tag{22}$$

After each iteration of updating the weights for meta-paths using Formula (21), we normalize the meta-path weights to satisfy the constraint $\sum_p w_p = 1$.

This learning algorithm is summarized in Algorithm 1. Overall, it is an iterative algorithm based on the expectation-maximization (EM) method. The optimization of meta-path weights $w_p$'s contains an inner loop of gradient descent

algorithm (lines 9-15). This learning algorithm can automatically learn the weights of meta-paths by maximizing the likelihood of observing named entity mentions $M$ in the given document collection $D$ without requiring any annotated training data, which makes our framework SHINE+ unsupervised.

We analyze the time complexity for this learning algorithm. Formally, for the inner gradient descent algorithm, the time complexity is $O(t_1|M| \cdot |E_m| \cdot |V_d| \cdot |\overrightarrow{W}|)$, where $t_1$ is the number of iterations, $|M|$ is the number of entity mentions in $M$, $|E_m|$ is the number of candidate entities for entity mention $m$, $|V_d|$ is the number of objects involved in the document $d$ where mention $m$ appears, and $|\overrightarrow{W}|$ is the number of meta-paths. The time complexity for the whole learning algorithm is $O(t(t_1|M| \cdot |E_m| \cdot |V_d| \cdot |\overrightarrow{W}| + |M| \cdot |E_m| \cdot |V_d| \cdot |\overrightarrow{W}|)) = O(t(t_1|M| \cdot |E_m| \cdot |V_d| \cdot |\overrightarrow{W}|))$ where $t$ is the number of iterations for the EM algorithm. Therefore, we can see that the inner gradient descent algorithm consumes most of running time of the whole learning algorithm. Though we do not know the upper bound on the number of iterations this EM learning algorithm may run until it terminates, in our experiments, we observe that it converges quickly and typically takes only a few iterations. Moreover, as $|E_m|$, $|V_d|$, and $|\overrightarrow{W}|$ are usually small constants, the running time of this weight learning algorithm and the inner gradient descent algorithm is linear to the number of entity mentions in $M$, which has been confirmed by our experiments shown in Section 5.3. When the number of entity mentions in $M$ is enormous, our weight learning algorithm becomes a little expensive. At that time, we could use stochastic gradient descent method that is very effective for large-scale learning problem, which samples a subset of entity mentions at each iteration and updates the parameters $w_p$'s on the basis of these sampled entity mentions only [23]. Then, the running time of our weight learning algorithm is linear to the number of sampled entity mentions.

## 4 The Knowledge Population Algorithm

In many cases, the information network cannot provide enough useful entity knowledge to help the entity linking model make correct linking decisions. It is very necessary to enrich the entity knowledge in the information network to improve the entity linking performance. To deal with this problem, we propose a knowledge population algorithm to iteratively enrich the network entity knowledge without the requirement of any labeled data. Specifically, we first run the entity linking model to link the entity mentions in the data set. After the linking process, our knowledge population algorithm regards each mention mapped by the linking model with a high confidence score as the *golden mapped mention*. For each golden mapped mention, our knowledge population algorithm adds the context information extracted from its appearing text into the information network to enrich the corresponding entity knowledge. In the subsequent iteration, the linking model could leverage the added entity knowledge to link the same set of entity mentions more accurately. Then a new set of golden mapped mentions can be generated and new entity knowledge can be added into the information network. This iterative process will continue until no new entity knowledge is added. It can be seen that this proposed algorithm performs entity linking and entity knowledge population jointly, and makes them mutually reinforce each other. In this paper, our choice for the linking model is the probabilistic linking model introduced in Section 3.

First, we define a confidence score for each mention mapped by the linking model, which is similar to the technique utilized in [24]. In our probabilistic linking model, the probability $\mathbf{P}(m, d, e)$ of an entity mention $m$ detected from a document $d$ referring to a specific entity $e$ (computed by Formula (4)) expresses the linking confidence. However, this probability computed in this way could be very small and not easy to interpret. We consider transforming these computed probabilities into normalized confidence scores. Specifically, for each triple $(m, d, e)$, its normalized confidence score $NS(m, d, e)$ is computed as

$$NS(m, d, e) = \frac{\mathbf{P}(m, d, e)}{\sum_{e_i \in E_m} \mathbf{P}(m, d, e_i)}, \tag{23}$$

where $E_m$ is the candidate entity set for a mention $m$. Intuitively, a mention $m$ is mapped by an entity linking model with high confidence if the highest score of some entity in its candidate entity set $E_m$ is far larger than the scores of other candidate entities. Therefore, we define the confidence score $CS(m)$ for each mention $m$ in a document $d$ as the highest score in its candidate entity set

$$CS(m) = NS(m, d, \arg\max_{e \in E_m} NS(m, d, e)). \tag{24}$$

We regard the mentions whose confidence scores are larger than a threshold $\gamma \in (0, 1)$ as the golden mapped mentions.

Once the golden mapped mentions are discovered, the remaining problem is how to enrich the corresponding entity knowledge that could be easily leveraged by the linking model. An intuitive method is to use the text where the golden mapped mention appears to construct a term-based representation for the corresponding entity in the information network. This method is suitable for many entity linking systems that utilize term-based representation to describe the entity existing in a knowledge base. Such kind of notable entity linking systems include AIDA [7], Illinois Wikifier [11], Kulkarni et al. [6], and Cucerzan [5]. Therefore, it is noted that the knowledge population algorithm developed in this paper can work with any of these entity linking systems to augment the entity knowledge and enhance the linking power.

In our probabilistic linking model, the entity-specific object model $\mathbf{P_e}(v)$ (Formula (10)) encodes the entity knowledge existing in the information network in the form of distributions over objects, and is generated using meta-path constrained random walks (Formula (9)). In order to make our linking model leverage the enriched entity knowledge easily, we add two new types of objects (i.e., documents (Dc) and document objects (DO)) to the network. Here, the object type of document means the document where the golden mapped mention appears, and the object type of document object means the object which constitutes the document, as we have the assumption that each document is composed of various objects from the heterogeneous information network. Links exist between objects with the same type as the golden mapped mention and documents by the relations *have* and *have* $^{-1}$, and between documents and document objects by the relations *contain* and *contain*$^{-1}$. The meta-path connecting the object type of golden mapped mentions with these two new

object types is called the *population meta-path*. For example, when we link author (or actor) name mentions with the DBLP (or IMDb) network, the population meta-path is A-Dc-DO (or Ac-Dc-DO). To leverage the enriched entity knowledge to link mentions, we add the population meta-path into the meta-path set $MP$ used by the probabilistic linking model. For each golden mapped mention, we add the document where this mention appears and its document objects into the information network. In this way, we represent the enriched entity knowledge as distributions over objects generated using meta-path constrained random walks along the population meta-path. Then, our linking model can automatically learn the relative importance for the population meta-path using the weight learning algorithm, which makes our model seamlessly take into account the enriched entity knowledge for entity linking.

The knowledge population algorithm is described in Algorithm 2. It is noted that in the first iteration of Algorithm 2, Algorithm 1 learns the meta-path weights without using the enriched entity knowledge, since at that time no document or document object is added into the information network and the distribution of objects given the population meta-path is empty. In the following iterations, Algorithm 1 learns the meta-path weights by leveraging the enriched entity knowledge.

**Algorithm 2.** *The Knowledge Population Algorithm*

**Input:** Heterogeneous information network $G$, named entity mentions $M$, document collection $D$, meta-path set $MP$, and threshold $\gamma$.

Add the population meta-path into $MP$
**repeat**
  Apply Algorithm 1 to learn meta-path weights
  **for** each mention $m \in M$ **do**
    Compute confidence score $CS(m)$ by Formula (24)
    **if** $CS(m) > \gamma$ **then**
      Set mention $m$ as the golden mapped mention
      Add the document where $m$ appears and its document objects to the information network $G$
    **end if**
  **end for**
**until** no new entity knowledge is added

Our proposed framework SHINE+ first utilizes the knowledge population algorithm (Algorithm 2) to enrich the entity knowledge until no new entity knowledge is added in the information network. Then SHINE+ runs the learning algorithm (Algorithm 1) to learn the final meta-path weights by leveraging the enriched entity knowledge. Lastly SHINE+ outputs the final linking results using Formula (2).

## 5 Experimental Study

To evaluate the effectiveness and efficiency of our framework SHINE+, we present a thorough experimental study in this section. We first describe the experimental setting in Section 5.1 and then study the effectiveness of SHINE+ in Section 5.2. In Section 5.3, we evaluate the efficiency and scalability of SHINE+. In Section 5.4, we study the impact of parameters to the performance of SHINE+. Lastly, we give a case study on the knowledge population algorithm. All the programs were implemented in JAVA and all the experiments were conducted on a server with 2.67 GHz CPU, 48 GB memory, and 64-bit Windows.

### 5.1 Experimental Setting

To the best of our knowledge, there is no publicly available benchmark data set for the task of entity linking with a HIN. In this paper, we choose two real heterogeneous information networks (i.e., the DBLP network and the IMDb network) as the underlying heterogeneous information networks, and link author/actor names in Web documents with their corresponding author/actor entities in the DBLP/IMDb network. For the DBLP network, we created two gold standard Web document data sets. For the IMDb network, we created one gold standard Web document data set. We make the three data sets online available for future research.[3] The annotation task for entity linking with a HIN consists of generating test Web documents that pertain to the same domain as the information network, detecting named entity mentions in them, and identifying their corresponding mapping entities existing in the network.

We downloaded the March 2013 version of the DBLP data set and built the DBLP network according to the network schema in Fig. 2a. This DBLP network contains over 1.2 M authors, 2.1 M papers, and 7 K venues (conferences/journals). The terms in the paper titles are filtered by a stop word list of size 667 and stemmed by Porter Stemmer.[4] We finally got around 408 K terms. According to our task setting, the entities in the network which would be linked with should be disambiguated. The DBLP network has some highly ambiguous author names (such as "Wei Wang", "Eric Martin", etc.) that have been disambiguated (i.e., determine which author names in publication records refer to the same author entity), and these ambiguous names are followed by a space character and a four digit number (e.g., "Wei Wang 0010" and "Eric Martin 0001") to uniquely represent each distinct author [25]. In addition, we combined the DBLP network with the author disambiguation results from a publicly available data set used in [26], which contains 110 author names and their gold standard disambiguation results, to create a partially disambiguated DBLP network.

For the IMDb network, we downloaded its January 2015 version and built it according to its network schema in Fig. 2b. This IMDb network contains over 2.6 M actors/actresses, 3.3 M movies, 0.3 M directors, and 3.5 M characters. The terms in actors' biographies and movies' plot keywords are also filtered by a stop word list and stemmed. Actor names in the IMDb network have been disambiguated and the ambiguous names are followed by a space character and a roman number in parentheses (e.g., Chris Evans (V) and Peter Alexander (XIV)) to uniquely represent each distinct actor.

To generate the test Web documents where mentions appear, we focus on Web documents that pertain to the same domain as the underlying information network (i.e., the DBLP or IMDb network). Given any Web document, we could develop a highly accurate classifier to predict whether it pertains to the same domain as the DBLP or IMDb network. Since the main focus of this paper is to investigate the effectiveness of our framework for entity linking, we consider developing such classifiers as an orthogonal effort to our task, and opted for querying Web search engine (i.e., Google) to generate a test document collection $D$ for each network. We formed the Web search queries by including randomly selected ambiguous author/actor names, as well as some domain

3. https://sites.google.com/site/weishen09/LinkHINdata.rar
4. http://tartarus.org/martin/PorterStemmer/

representative phrases (such as "computer science", "database", "actor", "movie" etc.). Each returned Web document, along with all candidate entities with respect to the ambiguous author/actor name in this document, is presented to annotators, and the documents which contain ambiguous names referring to the entities existing in the DBLP/IMDb network are collected. This yields a collection of 709 Web documents for the DBLP network which we refer to as the DBLP1 data set and a collection of 561 Web documents for the IMDb network which we refer to as the IMDb data set.

Besides leveraging search engine to generate data sets, we collected a large corpus of domain-specific Web documents from the websites of some CS/ECE departments, labs, and conferences. We filtered out the documents which do not contain ambiguous author names in the DBLP network, since we could output linking results for unambiguous author names directly without entity linking. From the remaining Web documents, we manually annotated 400 documents which contain ambiguous author names referring to the entities existing in the DBLP network. We refer to this data set as the DBLP2 data set. Each Web document in the three data sets introduced above has one author/actor name mention that needs to be linked.

To generate the candidate entities for each author/actor name mention, we use a method based on string comparison between names of the author/actor mention and the author/actor entity in the network. For each test HTML Web document, we first extracted the full text of the article and removed the author/actor name mention itself. As we assume each Web document $d$ consists of various multi-type objects $v$'s from the heterogeneous information network, for the test Web documents for DBLP, we recognized objects of author type and objects of venue type in them using dictionary-based exact matching method, while for the test Web documents for IMDb, we recognized objects of actor type, objects of director type, objects of character type, and objects of movie title type in them using the same matching method. For simplicity, when recognizing these objects using the above method, we regarded all object names are unambiguous (i.e., regarded the same object name representing the same object). For the test Web documents for DBLP, we identified objects of year type using regular expression. All remaining terms in the documents (removing all punctuation symbols) are filtered by a stop word list and stemmed. We regarded these stemmed terms as the object set of various term types.

In Section 5.4, we evaluate how accuracy changes by varying $\theta$ from 0.1 to 0.9 and varying $\gamma$ from $1\text{-}10^{-1}$ to $1\text{-}10^{-11}$ in the three data sets. The parameter $\theta$ is set to 0.2 and the threshold $\gamma$ is set to $1\text{-}10^{-9}$ in the other experiments. The learning rate $\alpha$ in Formula (21) decides the step size towards the increasing direction. When $\alpha$ gets too big, the gradient descent algorithm would fail to converge. $\alpha$ is set to 0.000003 in all experiments. To evaluate the performance of SHINE+, in this paper we adopt the evaluation measure *accuracy*, which is calculated as the number of correctly linked entity mentions divided by the total number of all mentions. All the operations introduced in this section are regarded as preprocessing.

## 5.2 Effectiveness Study

In this section, we study the effectiveness of our framework SHINE+ under different configurations, and compare them with several baselines.

### 5.2.1 Baselines

Since no previous work deals with the task of entity linking with a HIN, we created four baselines in this paper. The first one (POP) is entity popularity-based method. The feature of entity popularity has been found to be very useful in previous entity linking systems [7], [8], [11], [27]. In this POP baseline method, we used our entity popularity model (Formula (5)) introduced in Section 3.1 to estimate the popularity for each candidate entity. The entity with the highest popularity among all the candidate entities for each entity mention is considered as the mapping entity for this entity mention.

The second baseline (VSim) is vector similarity-based method. In this VSim method, we constructed a context vector for each entity mention and a profile vector for each candidate entity. Specifically, for each entity mention, we used the object sets of different types which compose the document where this entity mention appears to construct the context vector. For each candidate author entity, we obtained all her publication records from our partially disambiguated DBLP network, and added objects of different types (i.e., her coauthors, venues, title terms, and publication years) in her publications into the profile vector. For each candidate actor entity, we obtained all her movie records and her biography from the IMDb network, and added objects of different types (i.e., her biography terms, co-actors, movie titles, movie directors, characters, and movie plot keywords) into the profile vector. Then we measure the cosine similarity of the two vectors for each mention-entity pair. Finally, the entity with the highest similarity is considered as the mapping entity for the entity mention. Items in these vectors can be weighted by TF or TF-IDF, and we define the corresponding methods as $\text{VSim}_{\text{TF}}$ and $\text{VSim}_{\text{IDF}}$ respectively.

The third baseline (Tradi) leverages the features of entity popularity and context similarity, similar to the main idea of most traditional entity linking systems [28]. Specifically, it multiplies the popularity by the vector similarity from the above two baselines as the final score for each mention-entity pair. The entity with the highest score is output as the mapping entity for the entity mention. Due to the two different term weighting strategies of the baseline VSim, the baseline Tradi has two versions (i.e., $\text{Tradi}_{\text{TF}}$ and $\text{Tradi}_{\text{IDF}}$) as well.

The fourth baseline (SHINE) is the framework we proposed in our previous paper [20]. Compared with the SHINE+ framework present in this paper, the SHINE framework does not utilize the knowledge population algorithm to enrich the network entity knowledge and just leverages the probabilistic linking model (Section 3) for entity linking.

### 5.2.2 The SHINE+ Framework

The meta-paths used for entity linking with the DBLP network include: A-P-A, A-P-V, A-P-T, A-P-Y, A-P-A-P-A, A-P-V-P-A, A-P-A-P-V, A-P-T-P-V, A-P-A-P-T, and A-P-V-P-T. Among them, there are four length-2 meta-paths and six length-4 meta-paths. To analyze the effectiveness of different meta-path sets, we refer to our SHINE+ framework that just utilizes the four length-2 meta-paths as $\text{SHINE+}_{\text{part}}$, and refer to our SHINE+ framework that utilizes all ten meta-paths as $\text{SHINE+}_{\text{all}}$. The meta-paths used for linking with the IMDb network include: Ac-B, Ac-Mv-Ac, Ac-Mv-MT, Ac-Mv-Di, Ac-Mv-C, and Ac-Mv-K. Among them, there are one length-1 meta-path and five length-2 meta-paths. We also refer to SHINE+ that just utilizes the length-1 meta-path

TABLE 2
Experimental Results over the DBLP1, DBLP2, and IMDb Data Sets

| | DBLP1 | | DBLP2 | | IMDb | |
|---|---|---|---|---|---|---|
| Method | # | Accu. | # | Accu. | # | Accu. |
| POP | 345 | 0.487 | 212 | 0.53 | 304 | 0.542 |
| VSim $_{\text{TF}}$ | 604 | 0.852 | 346 | 0.865 | 406 | 0.724 |
| VSim $_{\text{IDF}}$ | 630 | 0.889 | 356 | 0.89 | 506 | 0.902 |
| Tradi $_{\text{TF}}$ | 610 | 0.860 | 350 | 0.875 | 418 | 0.745 |
| Tradi $_{\text{IDF}}$ | 638 | 0.900 | 362 | 0.905 | 515 | 0.918 |
| SHINE $_{\text{part}}$ | 655 | 0.924 | 366 | 0.915 | 483 | 0.861 |
| SHINE $_{\text{all}}$ | 668 | 0.942 | 375 | 0.938 | 537 | 0.957 |
| SHINE+$_{\text{part}}$ | 680 | 0.959 | 376 | 0.94 | 497 | 0.886 |
| SHINE+$_{\text{all}}$ | **692** | **0.976** | **387** | **0.968** | **555** | **0.989** |

as SHINE+$_{\text{part}}$, and refer to SHINE+ that utilizes all six meta-paths as SHINE+$_{\text{all}}$. For the baseline SHINE, we refer to it leveraging different meta-path sets in the same way.

The experimental results of all methods over the DBLP1, DBLP2, and IMDb data sets are shown in Table 2. Besides the accuracy, we also show the number of correctly linked entity mentions for all methods. From the results, we can see that our proposed framework SHINE+$_{\text{all}}$ significantly outperforms all the baseline methods over the three data sets (paired $t$-tests, $p < 0.05$), which demonstrates the effectiveness of our framework. We can also see that SHINE+$_{\text{all}}$ and SHINE+$_{\text{part}}$ significantly outperform the methods SHINE $_{\text{all}}$ and SHINE $_{\text{part}}$ respectively ($p < 0.05$), which means the knowledge population algorithm in the SHINE+ framework effectively enriches the network entity knowledge and greatly boosts the entity linking accuracy. Moreover, it can be also seen from Table 2 that the methods SHINE+$_{\text{all}}$ and SHINE $_{\text{all}}$ that leverage all meta-paths significantly outperform SHINE+$_{\text{part}}$ and SHINE $_{\text{part}}$ respectively over the three data sets ($p < 0.05$). The more useful meta-paths, the better the entity linking accuracy, which is consistent with our intuition, since our linking model can obtain more related knowledge about the candidate entity from the information network by leveraging more useful meta-paths.

### 5.3 Efficiency and Scalability Study

In this section, we study the scalability of SHINE+ using different subsets of the entity mention set over the DBLP1, DBLP2, and IMDb data sets. Fig. 4 plots the average running time for one iteration of the EM algorithm and one iteration of the inner gradient descent algorithm in the weight learning algorithm (Algorithm 1) with varied size of the entity mention set over the three data sets. From the results, we can see that the average running time for one iteration of both the EM algorithm and the inner gradient descent algorithm is about linear to the number of entity mentions in the data set, which is consistent with the time complexity analysis of Algorithm 1 described in Section 3.3. This evaluation demonstrates the scalability of SHINE+.

Fig. 5 depicts the accuracy performance of SHINE+$_{\text{all}}$ with varied size of the entity mention set over the three data sets. We can see that our proposed framework SHINE+$_{\text{all}}$ can achieve relatively stable and high accuracy with different size of data set, which demonstrates the robustness of our framework. SHINE+ can automatically learn the meta-path weights by maximizing the likelihood of observing named entity mentions in a given document collection.

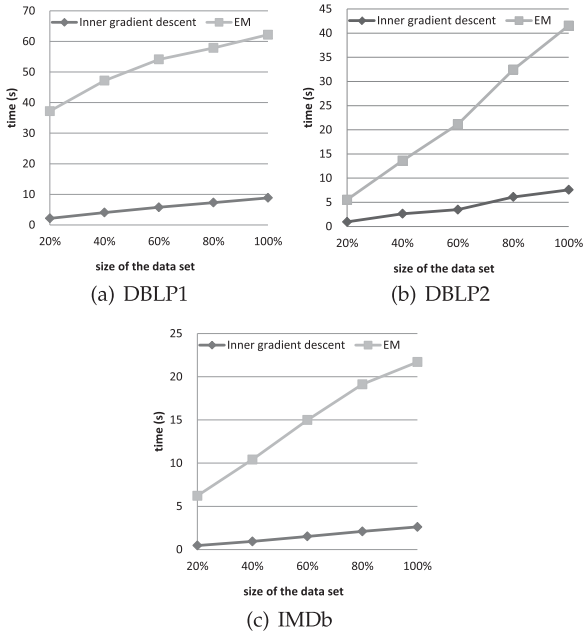



Fig. 4. Scalability evaluation.

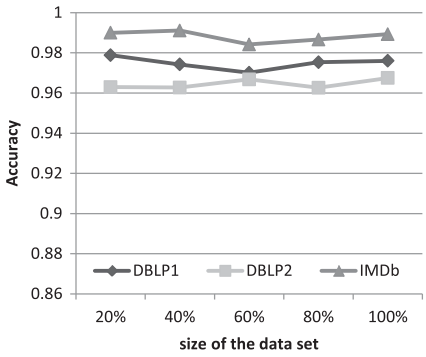



Fig. 5. Performance varying the size of the entity mention set.

Over various entity mention sets, although the learned meta-path weights are different, the final entity linking accuracy of our framework is stable and high, which means SHINE+ can select the most appropriate meta-paths according to different entity linking tasks.

### 5.4 Sensitivity Analysis

To better understand the performance characteristics of our proposed framework, we conducted sensitivity analysis to understand the impact of the parameters $\theta$ and $\gamma$ to SHINE +'s performance. $\theta$ in Formula (7) balances the two parts (i.e., the entity-specific object model and the generic object model for the domain). Fig. 6 shows the performance of SHINE+$_{\text{all}}$ with varied parameter $\theta$ from 0.1 to 0.9 over the three data sets. From the trend plotted in Fig. 6, it can be seen that when $\theta \in [0.1, 0.5]$, the accuracy achieved by SHINE+$_{\text{all}}$ is greater than 0.965, 0.955, and 0.975 over the DBLP1, DBLP2, and IMDb data sets respectively. Thus, we can say that when $\theta$ is varied from 0.1 to 0.5, the performance of SHINE+ is not very sensitive to the parameter $\theta$.

The threshold $\gamma \in (0, 1)$ in Algorithm 2 controls the quality of the golden mapped mention. The closer the threshold is to 1.0, the more likely the golden mapped mention is to be mapped correctly. Fig. 7 shows the performance of SHINE +$_{\text{all}}$ with varied threshold $\gamma$ from 1-$10^{-1}$ to 1-$10^{-11}$ over the

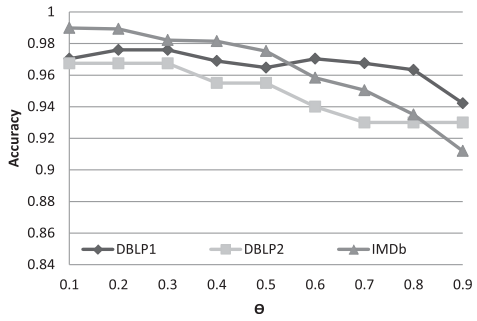



Fig. 6. Parameter study of $\theta$.

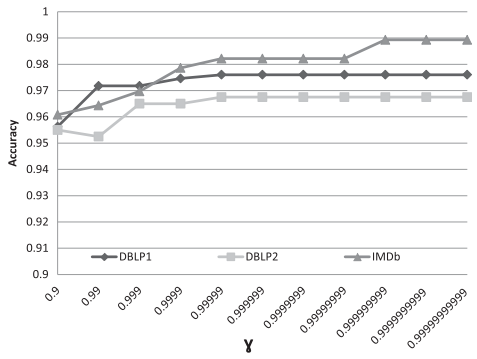



Fig. 7. Parameter study of $\gamma$.

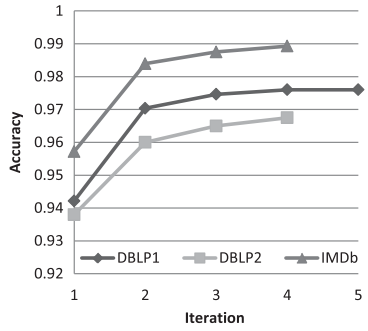



Fig. 8. Performance varying iterations of Algorithm 2.

three data sets. As shown in Fig. 7, we achieve better performance with larger values of $\gamma$. When $\gamma \geq 1 - 10^{-5}$, the accuracy achieved by SHINE+$_{\text{all}}$ over the three data sets is quite stable and high, and is insensitive to the parameter $\gamma$.

### 5.5 Case Study on Knowledge Population Algorithm

To illustrate the effectiveness of the knowledge population algorithm (Algorithm 2), we show how the performance of our SHINE+$_{\text{all}}$ method changes with respect to the number of iterations of Algorithm 2 over the three data sets in Fig. 8. From this figure, we can see that Algorithm 2 terminates after five iterations over the DBLP1 data set and four iterations over the DBLP2 and IMDb data sets. It demonstrates that the entity linking accuracy increases as the number of iterations increases, since more iterations bring more useful entity knowledge into the information network. Additionally, the increasing speed of the linking accuracy slows down as the number of iterations increases.

Table 3 shows parts of the enriched entity knowledge for several entities, which is represented as distributions over objects generated using meta-path constrained random walks along the population meta-path. The number within the parentheses after each object (term objects are stemmed) represents its probability. We can see that our knowledge population algorithm can provide complementary knowledge for entity linking, especially for the entities that are not very famous and do not have sufficient information in the network. For example, with respect to the entity "Ke Chen 0002" in DBLP, "imag, model, mathemat, comput, applic" indicate his research interests, and "liverpool, cmit, prof" correspond to his affiliated institution and his title. For the entity "Peter Alexander (XIV)" in IMDb, "nbc, todai, correspond, report" indicate that he is an NBC Correspondent known for the daily live broadcast Today, and "starl, alison, abc7" describe information on his spouse. All this entity knowledge is not contained in the DBLP or IMDb network, but available in the Web documents where the corresponding mentions appear. Our proposed knowledge population algorithm can find it out and enrich it into the network, which subsequently helps entity linking.

TABLE 3
The Enriched Entity Knowledge for Several Entities

| Entity | Enriched Entity Knowledge |
|---|---|
| Bin Yu 0000 | statist(0.0081), model(0.0039), berkelei(0.0037) data(0.0032), mathemat(0.0022), uc(0.0016) lasso(0.0015), professor(0.0015), california(0.0014) learn(0.0013), spars(0.0012), dimension(0.0012) |
| Ke Chen 0002 | imag(0.0079), model(0.0051), liverpool(0.0049) mathemat(0.0033), segment(0.003), comput(0.0028) cmit(0.0021), prof(0.0020), applic(0.0020) |
| John Clayton (V) | espn(0.0082), nfl(0.0065), sport(0.0032) seahawk(0.0029), seattl(0.0026), game(0.0024) report(0.0024), team(0.0023), radio(0.0017) |
| Peter Alexander (XIV) | nbc(0.0071), todai(0.0058), starl(0.0057) alison(0.0055), correspond(0.004), abc7(0.0039) report(0.0037), marri(0.0033), washington(0.0021) |

## 6 Related Work and Discussion

In recent years, the advent of knowledge sharing communities such as Wikipedia and the development of information extraction techniques have facilitated the automated construction of large scale machine-understanding knowledge bases. Knowledge bases contain rich information about the world's entities, their semantic classes, and their mutual relationships. Such kind of notable endeavors include DBpedia [29], YAGO [3], Freebase [30], ReadTheWeb [31], and Probase [32].

Most traditional entity linking methods focus on linking entities with Wikipedia or Wikipedia-derived knowledge bases (e.g., YAGO) [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], and are largely dependent on the special features associated with Wikipedia (e.g., Wikipedia articles or Wikipedia-based relatedness measures), which have been introduced in Section 1. Some of these systems leverage probabilistic methods. Specifically, Kulkarni et al. [6] started with an SVM-based supervised learner for local context similarity, and modeled it in combination with pairwise document-level topical coherence of candidate entities using a probabilistic graphical model. Han and Sun [9] proposed a generative probabilistic entity-mention model, by incorporating three types of knowledge (i.e., popularity knowledge, name knowledge, and context knowledge). In our SHINE+, we propose a probabilistic linking model,

which unifies an entity popularity model with an entity object model. You could refer to our survey paper [28] for more information about entity linking techniques.

Recently, some work has been proposed to deal with the domain-specific entity linking problem. Pantel and Fuxman [33] associated search engine queries with entities from a large product catalog, and Dalvi et al. [34] exploited the geographic aspects of tweets to infer the matches between tweets and restaurants from a list. D'souza and Ng [35] associated disease mentions in the biomedical text (e.g., clinical reports) with the corresponding concepts in a biomedical ontology. Our task is different from these existing entity linking problems, and no previous method can be applied to address it.

As object (or entity) names in information networks (such as bibliographic networks) are inherently ambiguous, considerable progresses have been made in the task of name disambiguation for these networks [26], [36], [37], [38], [39]. Given a set of entity names appearing in a network, the task is to determine which entity names refer to the same underlying entity. Essentially, this task is to cluster entity names referring to the same entity in a network into one cluster, which is different from our entity linking task addressed in this paper. For a comprehensive survey of the approaches for author name disambiguation, you could refer to the survey paper [40].

## 7 Conclusion

In this paper, we have studied the problem of entity linking with a heterogeneous information network and propose a general unsupervised framework SHINE+ to address it. We present a probabilistic linking model which combines an entity popularity model with an entity object model to link entities in text with the network. To further boost the entity linking performance, we propose a knowledge population algorithm which iteratively enriches the network entity knowledge by exploiting the results of the linking model. The experimental results over two real heterogeneous information networks (i.e., DBLP and IMDb) and three manually annotated Web document collections have shown that SHINE+ can output much more accurate linking results compared with the baselines, and is efficient and scalable. Our future work will consider entity linking for other domains, such as the biomedical domain and the music domain. Additionally, to develop more efficient entity linking techniques is also a promising direction for future research.

## Acknowledgments

This work was supported in part by the National Basic Research Program of China (973 Program) under Grant No. 2014CB340505, the National Natural Science Foundation of China under Grant No. 61532010, 61502253, U1636116 and 11431006, the National 863 Program of China under Grant No. 2015AA015401, the Fundamental Research Funds for the Central Universities, Research Fund for International Young Scientists under Grant No. 61650110510, the U.S. Army Research Lab. under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), U.S. National Science Foundation IIS-1320617, IIS 16-18481, and NSF IIS 17-04532, and grant 1U54GM114838 awarded by NIGMS through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative (www.bd2k.nih.gov).

## References

[1] Y. Sun, B. Norick, J. Han, X. Yan, P. S. Yu, and X. Yu, "Integrating meta-path selection with user-guided object clustering in heterogeneous information networks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 1348–1356.

[2] O. Deshpande, et al., "Building, maintaining, and using knowledge bases: A report from the trenches," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2013, pp. 1209–1220.

[3] F. Suchanek, G. Kasneci, and G. Weikum, "YAGO: A core of semantic knowledge unifying WordNet and Wikipedia," in *Proc. 16th Int. Conf. World Wide Web*, 2007, pp. 697–706.

[4] R. Bunescu and M. Pasca, "Using encyclopedic knowledge for named entity disambiguation," in *Proc. 11th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2006, pp. 9–16.

[5] S. Cucerzan, "Large-scale named entity disambiguation based on Wikipedia data," in *Proc. Joint Conf. Empirical Methods Natural Language Process. Comput. Natural Language Learn.*, 2007, pp. 708–716.

[6] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti, "Collective annotation of Wikipedia entities in Web text," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 457–466.

[7] J. Hoffart, et al., "Robust disambiguation of named entities in text," in *Proc. Conf. Empirical Methods Natural Language Process.*, 2011, pp. 782–792.

[8] W. Shen, J. Wang, P. Luo, and M. Wang, "LINDEN: Linking named entities with knowledge base via semantic knowledge," in *Proc. 21st Int. Conf. World Wide Web*, 2012, pp. 449–458.

[9] X. Han and L. Sun, "A generative entity-mention model for linking entities with knowledge base," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2011, pp. 945–954.

[10] W. Shen, J. Wang, P. Luo, and M. Wang, "Linking named entities in tweets with knowledge base via user interest modeling," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 68–76.

[11] L. Ratinov, D. Roth, D. Downey, and M. Anderson, "Local and global algorithms for disambiguation to Wikipedia," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2011, pp. 1375–1384.

[12] W. Shen, J. Wang, P. Luo, and M. Wang, "LIEGE: Link entities in web lists with knowledge base," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 1424–1432.

[13] S. Guo, M.-W. Chang, and E. Kiciman, "To link or not to link? a study on end-to-end tweet entity linking," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2013, pp. 1020–1030.

[14] R. Blanco, G. Ottaviano, and E. Meij, "Fast and space-efficient entity linking in queries," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2015, pp. 179–188.

[15] Y. Li, C. Wang, F. Han, J. Han, D. Roth, and X. Yan, "Mining evidences for named entity disambiguation," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 1070–1078.

[16] P. Ferragina and U. Scaiella, "TAGME: On-the-fly annotation of short text fragments (by Wikipedia entities)," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2010, pp. 1625–1628.

[17] D. Milne and I. H. Witten, "An effective, low-cost measure of semantic relatedness obtained from Wikipedia links," in *Proc. AAAI Workshop Wikipedia Artif. Intell.*, 2008, pp. 25–30.

[18] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "PathSim: Meta path-based top-k similarity search in heterogeneous information networks," *Proc. VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, 2011.

[19] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," in *Proc. 7th Int. Conf. World Wide Web*, 1998, pp. 107–117.

[20] W. Shen, J. Han, and J. Wang, "A probabilistic model for linking named entities in web text with heterogeneous information networks," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2014, pp. 1199–1210.

[21] C. D. Manning, P. Raghavan, and H. Schütze, Eds., *An Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[22] N. Lao and W. W. Cohen, "Relational retrieval using a combination of path-constrained random walks," *Mach. Learn.*, vol. 81, no. 1, pp. 53–67, Oct. 2010.

[23] L. Bottou and O. Bousquet, "The tradeoffs of large scale learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2008, pp. 161–168.

[24] J. Hoffart, Y. Altun, and G. Weikum, "Discovering emerging entities with ambiguous names," in *Proc. 23rd Int. Conf. World Wide Web*, 2014, pp. 385–396.

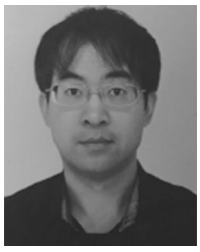
[25] M. Ley, "DBLP: Some lessons learned," *Proc. VLDB Endowment*, vol. 2, no. 2, pp. 1493–1500, Aug. 2009.
[26] X. Wang, J. Tang, H. Cheng, and P. S. Yu, "ADANA: Active name disambiguation," in *Proc. IEEE 11th Int. Conf. Data Mining*, 2011, pp. 794–803.
[27] H. Ji and R. Grishman, "Knowledge base population: Successful approaches and challenges," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2011, pp. 1148–1158.
[28] W. Shen, J. Wang, and J. Han, "Entity linking with a knowledge base: Issues, techniques, and solutions," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 2, pp. 443–460, Feb. 2015.
[29] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, and Z. Ives, "DBpedia: A nucleus for a web of open data," in *Proc. 6th Int. Semantic Web 2nd Asian Conf. Asian Semantic Web Conf.*, 2007, pp. 722–735.
[30] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2008, pp. 1247–1250.
[31] A. Carlson, J. Betteridge, R. C. Wang, E. R. Hruschka, Jr., and T. M. Mitchell, "Coupled semi-supervised learning for information extraction," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2010, pp. 101–110.
[32] W. Wu, H. Li, H. Wang, and K. Q. Zhu, "Probase: A probabilistic taxonomy for text understanding," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2012, pp. 481–492.
[33] P. Pantel and A. Fuxman, "Jigs and Lures: Associating web queries with structured entities," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2011, pp. 83–92.
[34] N. Dalvi, R. Kumar, and B. Pang, "Object matching in tweets with spatial models," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2012, pp. 43–52.
[35] J. D'souza and V. Ng, "Sieve-based entity linking for the biomedical domain," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2015, pp. 297–302.
[36] X. Yin, J. Han, and P. S. Yu, "Object distinction: Distinguishing objects with identical names," in *Proc. IEEE Int. Conf. Data Eng.*, 2007, pp. 1242–1246.
[37] P. Kanani, A. McCallum, and C. Pal, "Improving author coreference by resource-bounded information gathering from the Web," in *Proc. Int. Joint Conf. Artif. Intell.*, 2007, pp. 429–434.
[38] L. Shu, B. Long, and W. Meng, "A latent topic model for complete entity resolution," in *Proc. IEEE Int. Conf. Data Eng.*, 2009, pp. 880–891.
[39] P. Li, X. L. Dong, A. Maurino, and D. Srivastava,"Linking temporal records," *Proc. VLDB Endowment*, vol. 4, no. 11, pp. 956–967, Aug. 2011.
[40] A. A. Ferreira, M. A. Gonçalves, and A. H. Laender, "A brief survey of automatic methods for author name disambiguation," *ACM SIGMOD Rec.*, vol. 41, no. 2, pp. 15–26, 2012.

**Wei Shen** received the BS degree from Beihang University, China, in 2009 and the PhD degree in computer science from Tsinghua University, China, in 2014. He is an assistant professor in the College of Computer and Control Engineering, Nankai University, China. His research interests include entity linking, knowledge base population, and text mining. He is a recipient of the CAAI Outstanding Doctoral Dissertation Award and the Google PhD Fellowship.

**Jiawei Han** is Abel Bliss professor in engineering in the Department of Computer Science, University of Illinois. He has been researching into data mining, information network analysis, and database systems, with more than 600 publications. He served as the founding editor-in-chief of the *ACM Transactions on Knowledge Discovery from Data* and on the editorial boards of several other journals. He received the ACM SIGKDD Innovation Award (2004), IEEE Computer Society Technical Achievement Award (2005), IEEE Computer Society W. Wallace McDowell Award (2009), and Daniel C. Drucker Eminent Faculty Award at UIUC (2011). He is currently the director of the Information Network Academic Research Center (INARC) supported by the Network Science-Collaborative Technology Alliance (NS-CTA) program of U.S. Army Research Lab. His book *Data Mining: Concepts and Techniques* (Morgan Kaufmann) has been used worldwide as a textbook. He is a fellow of the ACM and the IEEE.

**Jianyong Wang** received the PhD degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, in 1999. He is currently a professor in the Department of Computer Science and Technology, Tsinghua University, Beijing, China. He was an assistant professor with Peking University, and visited Simon Fraser University, the University of Illinois at Urbana-Champaign, and the University of Minnesota at Twin Cities before joining Tsinghua University in December 2004. His research interests mainly include data mining and Web information management. He has co-authored more than 60 papers in some leading international conferences and some top international journals. He is serving or has served as a PC member for some leading international conferences, such as SIGKDD, VLDB, ICDE, WWW, and an associate editor of the *IEEE Transactions on Knowledge and Data Engineering* and the *ACM Transactions on Knowledge Discovery from Data*. He is a fellow of the IEEE and a member of the ACM.

**Xiaojie Yuan** received the BS, MS, and PhD degrees in computer science from Nankai University. She is currently working as a professor of the College of Computer and Control Engineering, Nankai University. She leads a research group working on topics of database, data mining, and information retrieval.



**Zhenglu Yang** received the BS degree from Tsinghua University, and the MS and PhD degrees from the University of Tokyo. He is currently working as a professor in the College of Computer and Control Engineering, Nankai University, China. His main research interests include data mining and web search.