




# Unsupervised meta-path selection for text similarity measure based on heterogeneous information networks

Chenguang Wang<sup>1</sup> · Yangqiu Song<sup>2</sup>  · Haoran Li<sup>3</sup> · Ming Zhang<sup>3</sup> · Jiawei Han<sup>4</sup>

Received: 3 July 2017 / Accepted: 2 July 2018  
© The Author(s) 2018

## Abstract

Heterogeneous information network (HIN) is a general representation of many different applications, such as social networks, scholar networks, and knowledge networks. A key development of HIN is called PathSim based on meta-path, which measures the pairwise similarity of two entities in the HIN of the same type. When using PathSim in practice, we usually need to handcraft some meta-paths which are paths over entity types instead of entities themselves. However, finding useful meta-paths is not trivial to human. In this paper, we present an unsupervised meta-path selection approach to automatically find useful meta-paths over HIN, and then develop a new similarity measure called KnowSim which is an ensemble of selected meta-paths. To solve the high computational cost of enumerating all possible meta-paths, we propose to use an approximate personalized PageRank algorithm to find useful subgraphs to allocate the meta-paths. We apply KnowSim to text clustering and classification problems to demonstrate that unsupervised meta-path selection can help improve the clustering and classification results. We use Freebase, a well-known world knowledge base, to conduct semantic parsing and construct HIN for documents. Our experiments on 20Newsgroups and RCV1 datasets show that KnowSim results in impressive high-quality document clustering and classification performance. We also demonstrate the approximate personalized PageRank algorithm can efficiently and effectively compute the meta-path based similarity.

**Keywords** Heterogeneous information network · Similarity · Text categorization

## 1 Introduction

Heterogeneous information network (HIN) (Han et al. 2010) is a general representation of multi-typed network, such as scholar or social networks (Sun et al. 2011, 2012)

---

Responsible editor: Hanghang Tong

Extended author information available on the last page of the article

Published online: 14 July 2018

and knowledge graph networks (Wang et al. 2015a). It has attracted increasing attention recently due to its flexibility of representation power (Han et al. 2010; Sun et al. 2011; Zhang et al. 2014; Wang et al. 2015a). A key concept in HIN is called meta-path, which is a path defined over entities types instead of entities themselves. For example, we have documents, words, and related named entities, which form an HIN. A meta-path defined over this HIN can be  $Document \xrightarrow{\text{contain}} Word \xrightarrow{\text{contain}^{-1}} Document$  and  $Document \xrightarrow{\text{contain}} Politician \xrightarrow{\text{contain}^{-1}} Document$ . Then given the designed meta-path, we can define similarities among entities based on the meta-path. One of the most important similarities is called PathSim (Sun et al. 2011), which considers how many path instances of a given meta-path can be found from the HIN, normalized by the numbers of the self-reachable path instances. For example, for the meta-path  $Document \xrightarrow{\text{contain}} Word \xrightarrow{\text{contain}^{-1}} Document$ , the PathSim relies on the shared words between two documents normalized by both documents' word counts.

However, PathSim only computes over one meta-path, and the meta-path should be hand-crafted by human. This can be very challenging when there are many entity types. For example, in a knowledge base, Freebase (Bollacker et al. 2008), there are about 1500+ entity types and 3500+ relation types. Then it would be very difficult for a human to find a meaningful meta-path even longer than three relations. Thus, we need an automatic way to find interesting and useful meta-paths. Previously, a semi-supervised learning approach has been proposed to find useful meta-paths (Sun et al. 2012, 2013). However, the supervision is still task-dependent, and when there are very little labeled information, the learned meta-path could be biased. Thus, if we can develop an unsupervised meta-path selection algorithm to automatically find useful meta-paths, many real-world applications can benefit from it. Another challenge is that to compute the overall meta-path based similarity, we need to produce a lot of meta-path based similarity matrices. This also introduces computation and storage problems when the number of meta-paths is large.

In this paper, we propose an unsupervised meta-path selection and ensemble approach, KnowSim, to derive an HIN based similarity measure that explores the explicit structural information from knowledge bases to compute document similarities. We use document similarity as an illustration but the idea can be generalizable to all other networks that can be regarded as HINs.

To construct an HIN from texts, we use the source of world knowledge, Freebase (Bollacker et al. 2008), which is a collaboratively collected knowledge base about entities and their organizations. We follow Wang et al. (2015a) to use the world knowledge specification framework including a semantic parser to ground any text to the knowledge bases, and a conceptualization-based semantic filter to resolve the ambiguity problem when adapting world knowledge to the corresponding document. By the specification of world knowledge, we have the documents as well as the extracted entities and their relations. For example, named entities (“Clinton” and “Obama”) and their types (*Person* and *Politician*), as well as the documents and the words can be used to form the HIN.

Given a constructed HIN, we can use meta-path based similarity to measure the similarity between two documents in the network. Rather than asking users to provide meaningful meta-path(s), we propose an automatic way to generate meta-paths for a given set of documents. In this case, an efficient mechanism should be developed to enumerate all the possible meta-paths of interests and locate the best ones. Based on the PageRank-Nibble algorithm (Andersen et al. 2006) that can conduct efficient graph pruning for a single node, we develop *Meta-path Dependent PageRank-Nibble* algorithm to locally partition the large-scale HIN (in our case, consisting of 108,722 entities and 9,655,466 relations) given a meta-path, and then based on the local partition to approximate commuting matrices for all meta-paths. We then store all the commuting matrices generated based on the local partition, which saves up to 15% space compared to that based on the original network. Thus, the meta-path generation process can be approximated in time independent of the size of the underlying network with low accuracy loss and high space saving. Then we perform meta-path selection based on feature selection algorithms [i.e., maximal spanning tree (Sahami 1998) and Laplacian score (He et al. 2006) based methods] by defining the meta-path similarities based on document-meta-path co-occurrences. We define an unsupervised knowledge-driven document similarity measure, *KnowSim*, which incorporates the selected meta-paths to represent the links between documents. The computation of KnowSim can be done in nearly linear time using the precomputed commuting matrices.

The contributions of this work are highlighted as follows:

- We formulate the document similarity problem as a graph base similarity problem over heterogeneous information networks.
- We propose a personalized PageRank based algorithm to automatically compute the meta-path based commuting matrices efficiently and effectively.
- An HIN-based document similarity measure, KnowSim, is introduced for better use of the link information (meta-paths) in an unsupervised way.
- Experiments on two datasets (20newsgroups and RCV1) show that our approach performs 12.3% better in comparison with the state-of-the-art document similarity measures.

A preliminary version of this work appeared in the proceedings of ICDM 2015 short paper (Wang et al. 2015b) and AAAI 2016 (Wang et al. 2016a). Here, we made several major improvements. First, we add basic concepts of HINs before introducing the document HIN construction, and add more details on how to construct HIN for documents, including two steps: semantic parsing and semantic filtering. Second, we add more details of the efficient Meta-path Dependent PageRank-Nibble algorithm that is used to generate meta-paths in the document HIN, and show the proof of how the algorithm satisfies the efficiency bound. Third, we show the mathematical details of two meta-path selection methods: (1) maximal spanning tree based selection; and (2) Laplacian score based selection. Fourth, we also add more computational details of KnowSim. Finally, in addition to demonstrate the effectiveness of the proposed document similarity measure, we add more experiments on parameter study and quantitative evaluation of the performance of meta-path generation algorithm. The code is available at <https://github.com/cgraywang/TextHIN> and the datasets used in this paper are available at <https://github.com/cgraywang/TextHINData>.

The remainder of the paper is organized as follows. We first introduce the basic concepts in HIN in Sect. 2. Then we formulate our KnowSim framework and introduce how the similarities over a set of meta-paths can be computed in Sect. 3. In Sect. 4, we provide a fast approximate algorithm to efficiently compute the meta-path based similarities over HIN with many meta-paths. Then in Sect. 5, we show our example of texts as HIN and introduce how to use the similarities derived in this paper in text clustering and classification problems. In Sect. 6, we show our experimental results on how our approach is effective and efficient on the benchmark datasets. Finally, we introduce the related work in Sect. 7 followed by the conclusion in Sect. 8.

## 2 Heterogeneous information networks

In this section, we review some key concepts related to heterogeneous information network (HIN).

**Definition 1** A *heterogeneous information network* (HIN) is a graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  with an entity type mapping  $\phi: \mathcal{V} \rightarrow \mathcal{A}$  and a relation type mapping  $\psi: \mathcal{E} \rightarrow \mathcal{R}$ , where  $\mathcal{V}$  denotes the entity set,  $\mathcal{E}$  denotes the link set,  $\mathcal{A}$  denotes the entity type set, and  $\mathcal{R}$  denotes the relation type set, and the number of entity types  $|\mathcal{A}| > 1$  or the number of relation types  $|\mathcal{R}| > 1$ .

**Definition 2** Given the HIN  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  with the entity type mapping  $\phi: \mathcal{V} \rightarrow \mathcal{A}$  and the relation type mapping  $\psi: \mathcal{E} \rightarrow \mathcal{R}$ , the *network schema* for network  $G$ , denoted as  $\mathcal{T}_G = (\mathcal{A}, \mathcal{R})$ , is a graph with nodes as entity types from  $\mathcal{A}$  and edges as relation types from  $\mathcal{R}$ .

The network schema provides a high-level description of a given heterogeneous information network. Another important concept, meta-path (Sun et al. 2011), is proposed to systematically define relations between entities at the schema level.

**Definition 3** A *meta-path*  $\mathcal{P}$  is a path defined on the graph of network schema  $\mathcal{T}_G = (\mathcal{A}, \mathcal{R})$ , and is denoted in the form of  $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \dots \xrightarrow{R_L} A_{L+1}$ , which defines a composite relation  $R = R_1 \cdot R_2 \cdot \dots \cdot R_L$  between types  $A_1$  and  $A_{L+1}$ , where  $\cdot$  denotes relation composition operator, and  $L$  is the length of  $\mathcal{P}$ .

We say a meta-path is *symmetric* if the relation  $R$  is symmetric. For simplicity, we use type names connected by “-” to denote the meta-path when there exist no multiple relations between a pair of types:  $\mathcal{P} = (A_1 - A_2 - \dots - A_{L+1})$ . For example, in the Freebase network, the composite relation *two Person co-founded an Organization* can be described as *Person*  $\xrightarrow{\text{found}}$  *Organization*  $\xrightarrow{\text{found}^{-1}}$  *Person*, or *Person-Organization-Person* for simplicity. We say a path  $p = (v_1 - v_2 - \dots - v_{L+1})$  between  $v_1$  and  $v_{L+1}$  in network  $\mathcal{G}$  follows the meta-path  $\mathcal{P}$ , if  $\forall l, \phi(v_l) = A_l$  and each edge  $e_l = \langle v_l, v_{l+1} \rangle$  belongs to each relation type  $R_l$  in  $\mathcal{P}$ . We call these paths as *path instances* of  $\mathcal{P}$ , denoted as  $p \in \mathcal{P}$ .  $R_l^{-1}$  represents the reverse order of relation  $R_l$ .

For calculation based on meta-paths, commuting matrix (Sun et al. 2011) is defined as follows.

**Definition 4 (Commuting matrix)** Given a network  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  and its network schema  $\mathcal{T}_{\mathcal{G}}$ , a commuting matrix  $\mathbf{M}_{\mathcal{P}}$  for a meta-path  $\mathcal{P} = (A_1 - A_2 - \dots - A_{L+1})$  is defined as  $\mathbf{M}_{\mathcal{P}} = \mathbf{W}_{A_1 A_2} \mathbf{W}_{A_2 A_3} \dots \mathbf{W}_{A_L A_{L+1}}$ , where  $\mathbf{W}_{A_k A_{k+1}}$  is the adjacency matrix between types  $A_k$  and  $A_{k+1}$ .  $\mathbf{M}_{\mathcal{P}}(i, j)$  represents the number of path instances between objects  $v_i$  and  $v_j$ , where  $\phi(v_i) = A_1$  and  $\phi(v_j) = A_{L+1}$ , under meta-path  $\mathcal{P}$ .

### 3 KnowSim: unsupervised ensemble of meta-path based similarities

In this section, we introduce an HIN-based similarity measure, *KnowSim*. We present our meta-path weighting methodology based on two feature selection techniques which can speed up the similarity computation using the precomputed commuting matrices.

Given an HIN, meta-paths can be used to compute the similarity between entities. For example, two documents can be connected by different types of entities, e.g., the link between entities “Obama” and “Clinton” is given by entity type *Politician*, and such link and type information can be useful to define the similarity according to semantic “Politics”. Following Sun et al. (2011), we use PathSim to define the similarity along a meta-path.

**Definition 5 (PathSim: A meta-path based similarity measure)** Given a symmetric meta path  $\mathcal{P}$ , PathSim between two entities  $v_i$  and  $v_j$  of the same entity type  $\phi(v_i) = A$  and  $\phi(v_j) = A$  is:

$$PS(v_i, v_j) = \frac{2 \times |\{p_{v_i \rightsquigarrow v_j} \in \mathcal{P}\}|}{|\{p_{v_i \rightsquigarrow v_j} \in \mathcal{P}\}| + |\{p_{v_j \rightsquigarrow v_i} \in \mathcal{P}\}|}. \tag{1}$$

PathSim takes the link information via meta-path  $\mathcal{P}$  between two entities  $v_x$  and  $v_y$  into consideration. Besides, PathSim also satisfies *symmetric*, *self-maximum* and *balance of visibility* properties (Sun et al. 2011). Obviously, PathSim measures entity similarity based on one meta-path. Previous approaches require human to define the meta-path(s). Here we should have multiple meta-paths useful for finding similar entities. Therefore, it is necessary to provide an automated mechanism to select the most meaningful meta-paths to define similarity between entities.

#### 3.1 Meta-path selection

We first define the entity-meta-path representation, and then use two feature selection methods to perform automatic meta-path selection.

##### 3.1.1 Entity-meta-path representation

For each meta-path  $\mathcal{P}_j$ , we have a commuting matrix  $\mathbf{M}_{\mathcal{P}_j}$ . Suppose we have  $N$  interested entities and  $M$  interested (automatically generated) meta-paths. Then we can use a tensor  $\mathbf{T} \in \mathbb{R}^{M \times N \times N}$  to encode all the numbers of meta-paths, where  $\mathbf{T}_{j,i,k} = \mathbf{M}_{\mathcal{P}_j}(i, k)$ . Based on this tensor representation, we can have different similarities

between entities or between meta-paths. Here we propose to use a simplest way based on entity-meta-path co-occurrence representation. We generate an entity meta-path representation matrix  $\mathbf{D} \in \mathbb{R}^{N \times M}$  where  $\mathbf{D}_{i,j} = \sum_k \mathbf{T}_{j,i,k}$ , which means that  $\mathbf{D}_{i,j}$  is the row sum of  $\mathbf{M}_{\mathcal{P}_j}$ . Summing the  $i$ -th row of  $\mathbf{M}_{\mathcal{P}_j}$  represents the density degree of this meta-path  $j$  for entity  $i$  (or short for  $v_i$ ). If the meta-path  $j$  is dense for entity  $i$  in the HIN, then most pairs related to entity  $i$  should have value in  $\mathbf{M}_{\mathcal{P}_j}$ . Then  $\mathbf{D}_{i,j}$  will be large. Then we can use the distribution of density over all the entities to evaluate the meta-path similarity. Specifically, we can define  $\text{sim}(\mathbf{D}_{\cdot,j_1}, \mathbf{D}_{\cdot,j_2})$  where  $\mathbf{D}_{\cdot,j_1}$  is the  $j_1$ -th column of  $\mathbf{D}$ . For example, we can use cosine score of two vectors or kernels to define the similarity. Moreover, we can define the entity similarity based on all the meta-path densities for the entities. Specifically, we can define  $\text{sim}(\mathbf{D}_{i_1,\cdot}, \mathbf{D}_{i_2,\cdot})$  where  $\mathbf{D}_{i_1,\cdot}$  is the  $i_1$ -th row of  $\mathbf{D}$ . Note that we do not use this entity similarity as our final similarity between two entities because it is only based on meta-path density. What we need is more elaborate entity similarity based on each entity meta-path pair. We will introduce the meta-path specific semantically meaningful similarity in the next section.

Given the similarities defined above and inspired by Song et al. (2009), we introduce two feature selection methods based on them to select the most meaningful meta-paths.

### 3.1.2 Maximal spanning tree based selection

Inspired by the mutual information-based feature selection (Sahami 1998), we propose to use maximal spanning tree (MST) to select only the meta-paths with the largest dependencies with others. The motivation behind using MST is that “features that only weakly influence the remaining domain variables are candidates for elimination” for mixture models (Sahami 1998). Since we have represented each document as its meta-path features where entity  $v_i$  is represented as meta-path  $\mathcal{P}_j$  as  $\mathbf{D}_{i,j}$ , we can leverage this feature selection method to find the most important meta-paths for a set of documents. Intuitively, if two meta-paths have similar density distributions over all the entities, then these two meta-paths are dependent. Therefore, we replace the mutual information in the original one with cosine similarity due to the consideration of the computational cost. We follow the steps below to find the best meta-paths to use.

1. We construct a complete graph  $\mathcal{G}_{\mathcal{M}}$  where the weight of the edge between  $\mathcal{P}_{j_1}$  and  $\mathcal{P}_{j_2}$  is  $\cos(\mathbf{D}_{\cdot,j_1}, \mathbf{D}_{\cdot,j_2})$ .
2. Build the MST based on  $\mathcal{G}_{\mathcal{M}}$ .
3. Define the relevant meta-path set  $\mathbf{P}_{rel} = \{\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_M\}$ .
4. While  $|\mathbf{P}_{rel}| > 0$ ,
  - (a) if  $\exists$  a node where  $\mathcal{P}_j \in \mathbf{P}_{rel}$  that is not connected to the others in  $\mathbf{P}_{rel}$ , remove this meta-path:  $\mathbf{P}_{rel} \leftarrow \mathbf{P}_{rel} - \mathcal{P}_j$ .
  - (b) otherwise remove the least weighted edge from MST.
5. Rank the meta-paths according to the order in which they were removed. Rank the last removed meta-path the highest. We weight each meta-path by the average similarity with the other meta-paths, i.e.,  $\frac{\sum_{j \neq i} \cos(\mathbf{D}_{\cdot,j_1}, \mathbf{D}_{\cdot,j_2})}{M-1}$ .

To compute the cosine similarity for meta-path re-ranking needs  $O(M^2N)$ . We use the Prime’s algorithm to construct the MST. By using a heap to construct a priority queue, we can build an MST in  $O(M^2 \log M)$  time.

### 3.1.3 Laplacian score based selection

We also use the Laplacian score to select meta-paths (He et al. 2006). Different from the MST based method that reflects the dependency between meta-paths, the Laplacian score represents the power of a meta-path in discriminating entities from different clusters. It consists of following steps.

1. Construct the  $K$ -nearest neighbor graph based on a similarity matrix  $\mathbf{S}$  where  $\mathbf{S}_{ij} = \exp\{-d_{ij}^2/2\sigma^2\}$ . Here,  $d_{ij}$  is the Euclidian distance between  $\mathbf{D}_{i,\cdot}$  and  $\mathbf{D}_{j,\cdot}$ ,  $\sigma$  is control parameter. We use the self-tuning method to compute  $\sigma$  (Zelnik-manor and Perona 2005).
2. Compute graph Laplacian  $\mathbf{L} = \mathbf{\Lambda} - \mathbf{S}$  where  $\mathbf{\Lambda}$  is a diagonal matrix and  $\mathbf{\Lambda}_{ii} = \sum_{j=1}^N \mathbf{S}_{ij}$  is the degree of the  $i$ -th vertex.
3. For each meta-path  $\mathcal{P}_j$  we have a column vector  $\mathbf{D}_{\cdot,j} \in \mathbb{R}^N$ , let  $\tilde{\mathbf{D}}_{\cdot,j} = \mathbf{D}_{\cdot,j} - \frac{\mathbf{D}_{\cdot,j}^T \mathbf{\Lambda} \mathbf{1}}{\mathbf{1}^T \mathbf{\Lambda} \mathbf{1}} \mathbf{1}$  where  $\mathbf{1} = [1, 1, \dots, 1]^T$ .
4. Compute the Laplacian score of the  $j$ -th meta-path:

$$L_j = \frac{\tilde{\mathbf{D}}_{\cdot,j}^T \mathbf{L} \tilde{\mathbf{D}}_{\cdot,j}}{\tilde{\mathbf{D}}_{\cdot,j}^T \mathbf{\Lambda} \tilde{\mathbf{D}}_{\cdot,j}}. \tag{2}$$

To find the  $K$ -nearest neighbors of a meta-path, we keep a  $K$ -size heap. For each meta-path, we compute its distances to all the other meta-paths and then check whether to insert it to the heap. Thus, the main time complexity is in graph Laplacian construction which is  $O(N^2M + N^2 \log K)$ .

### 3.2 KnowSim ensemble

Given the selected meta-paths, we now define our similarity measure, *KnowSim*. KnowSim is an extension of PathSim, which can take multiple selected meta-paths into account. Intuitively, if two entities are more strongly connected by the important (i.e., highly weighted) meta-paths, they tend to be more similar. Formally, we have the following definition.

**Definition 6** (*KnowSim: unsupervised ensemble of HIN based similarity*) Given a collection of symmetric meta-paths, denoted as  $\mathbf{P} = \{\mathcal{P}_m\}_{m=1}^{M'}$ , KnowSim between two entities  $d_i$  and  $d_j$  is defined as:

$$KS(d_i, d_j) = \frac{2 \times \sum_m^{M'} \omega_m |\{p_{i \rightsquigarrow j} \in \mathcal{P}_m\}|}{\sum_m^{M'} \omega_m |\{p_{i \rightsquigarrow i} \in \mathcal{P}_m\}| + \sum_m^{M'} \omega_m |\{p_{j \rightsquigarrow j} \in \mathcal{P}_m\}|}, \tag{3}$$

where we use  $d_i$  and  $d_j$  to denote the interested entities to distinguish with other types of entities  $v_i$ 's,  $p_{i \rightsquigarrow j} \in \mathcal{P}_m$  is a path instance between  $d_i$  and  $d_j$  following meta-path  $\mathcal{P}_m$ ,  $p_{i \rightsquigarrow i} \in \mathcal{P}_m$  is that between  $d_i$  and  $d_i$ , and  $p_{j \rightsquigarrow j} \in \mathcal{P}_m$  is that between  $d_j$  and  $d_j$ . We have  $|\{p_{i \rightsquigarrow j} \in \mathcal{P}_m\}| = \mathbf{M}_{\mathcal{P}_m}(i, j)$ ,  $|\{p_{i \rightsquigarrow i} \in \mathcal{P}_m\}| = \mathbf{M}_{\mathcal{P}_m}(i, i)$ , and  $|\{p_{j \rightsquigarrow j} \in \mathcal{P}_m\}| = \mathbf{M}_{\mathcal{P}_m}(j, j)$ . We use a vector  $\omega = [\omega_1, \dots, \omega_m, \dots, \omega_{M'}]$  to denote the meta-path weights, where  $\omega_m$  is the weight of meta-path  $\mathcal{P}_m$ .  $M'$  is the number of selected meta-paths.

$KS(d_i, d_j)$  is defined in two parts: (1) the *semantic overlap* in the numerator, which is defined by the number of meta-paths between entities  $d_i$  and  $d_j$ ; and (2) the *semantic broadness* in the denominator, which is defined by the number of total meta-paths between themselves. We can see that the larger number of meta-paths between  $d_i$  and  $d_j$ , the more similar the two entities are, which is further normalized by the semantic broadness of  $d_i$  and  $d_j$ .

For example, for two entities  $d_i$  (the upper document) and  $d_j$  (the lower document), the meta-path set  $\mathbf{P}$  includes two meta-paths:

$$\mathcal{P}_1 = Document \xrightarrow{\text{contain}} Politician \xrightarrow{\text{presidentOf}} Country \xrightarrow{\text{presidentOf}^{-1}} Politician \xrightarrow{\text{contain}^{-1}} Document,$$

and

$$\mathcal{P}_2 = Document \xrightarrow{\text{contain}} State \xrightarrow{\text{contain}^{-1}} Country \xrightarrow{\text{contain}} State \xrightarrow{\text{contain}^{-1}} Document.$$

By looking at the HIN, we can find  $|\{p_{d_i \rightsquigarrow d_j} \in \mathcal{P}_1\}| = 1$ ,  $|\{p_{d_i \rightsquigarrow d_j} \in \mathcal{P}_2\}| = 1$ ,  $|\{p_{d_i \rightsquigarrow d_i} \in \mathcal{P}_1\}| = 1$ ,  $|\{p_{d_i \rightsquigarrow d_i} \in \mathcal{P}_2\}| = 2$ ,  $|\{p_{d_j \rightsquigarrow d_j} \in \mathcal{P}_1\}| = 2$ , and  $|\{p_{d_j \rightsquigarrow d_j} \in \mathcal{P}_2\}| = 3$ . Besides, we are also given the corresponding meta-path weights  $\omega = [0.8, 0.2]$ . The KnowSim between these two documents is:

$$KS(d_i, d_j) = \frac{2 \times (0.8 \cdot 1 + 0.2 \cdot 1)}{(0.8 \cdot 1 + 0.2 \cdot 2) + (0.8 \cdot 2 + 0.2 \cdot 3)} = 0.588,$$

which has shown that the two documents are rather similar given the meta-path collection according to semantic ‘‘Politics’’ that we may be interested in.

KnowSim satisfies several nice properties as indicated from properties (1) to (3). The proof is similar to the proof of Theorem 1 in Sun et al. (2011).

- (1) Range:  $\forall d_i, d_j, 0 \leq KS(d_i, d_j) \leq 1$ . This is because  $\omega_m \cdot |\{p_{i \rightsquigarrow j} \in \mathcal{P}_m\}|, |\{p_{i \rightsquigarrow i} \in \mathcal{P}_m\}|, |\{p_{j \rightsquigarrow j} \in \mathcal{P}_m\}| \geq 0$ , and  $2 \times |\{p_{i \rightsquigarrow j} \in \mathcal{P}_m\}| \leq (|\{p_{i \rightsquigarrow i} \in \mathcal{P}_m\}| + |\{p_{j \rightsquigarrow j} \in \mathcal{P}_m\}|) \forall m$ .
- (2) Symmetric:  $KS(d_i, d_j) = KS(d_j, d_i)$ . This is because  $p_{i \rightsquigarrow j}$  is symmetric.
- (3) Self-maximum:  $KS(d_i, d_i) = 1$ . This is because  $2 \times |\{p_{i \rightsquigarrow j} \in \mathcal{P}_m\}| \leq (|\{p_{i \rightsquigarrow i} \in \mathcal{P}_m\}| + |\{p_{j \rightsquigarrow j} \in \mathcal{P}_m\}|)$ .

Note that if KnowSim only contains a single meta-path, it degenerates to PathSim.



## 4 Offline meta-path calculation

It is costly to compute the commuting matrix for a meta-path involving multiple entity types since it requires a matrix multiplication to compute two consecutive relations connecting entity types in the path (Sun et al. 2011). It is unnecessary to use the full HIN constructed in the previous section, since not all the entities are related. Inspired by Lao et al.' work (Lao and Cohen 2010; Lao et al. 2011), we use a meta-path dependent random walk to reduce the complexity of the HIN inference. We adopt a similar random walk algorithm which is based on personalized random walk (Andersen et al. 2006) with stops to enumerate all the meta-path relevant nodes in the HIN. We employ the modified version of approximate personalized PageRank called *PageRank-Nibble* algorithm (Andersen et al. 2006). The advantage of using this algorithm is that we can have a theoretical guarantee of the random walk approximation to the original HIN in the sense of the network structure. The goal of *PageRank-Nibble* algorithm is to find a small, low-conductance component  $\hat{\mathcal{G}}$  of a large graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  that contains a given node  $v$ . In our setting, instead of a single given node, we need  $\hat{\mathcal{G}}$  that contains a node set  $\hat{\mathcal{V}}$ . Specifically, in our case, we need the set of documents so that  $\mathcal{D} = \hat{\mathcal{V}} \subseteq \mathcal{V}$ .

The *PageRank-Nibble* algorithm starting with a node set  $\hat{\mathcal{V}}$  is called *Meta-path Dependent PageRank-Nibble*. We maintain a pair of distributions: an approximate personalized PageRank vector  $\mathbf{p}$ , and a residual error vector  $\mathbf{r}$  related to  $\mathbf{p}$ . Initially,  $\mathbf{p} = \mathbf{0}$  and  $\mathbf{r} = \mathcal{X}_{\hat{\mathcal{V}}}$ , where  $\mathbf{0}$  is a zero vector and  $\mathcal{X}_{\hat{\mathcal{V}}}$  is a function defined as

$$\mathcal{X}_{\hat{\mathcal{V}}} = \begin{cases} \frac{1}{|\hat{\mathcal{V}}|} & \text{if } v \in \hat{\mathcal{V}}, \forall v \in \mathcal{V} \\ 0 & \text{otherwise.} \end{cases}$$

We show the outline of the Meta-path Dependent PageRank-Nibble algorithm in Algorithm 1. The algorithm repeatedly picks a node  $u$  with a large residual error/degree ratio  $\frac{\mathbf{r}[u]}{d[u]}$ , where  $\mathbf{r}[u]$  is the residual error of  $u$  and  $d[u]$  denotes the degree of  $u$ . Afterwards, the algorithm uses a *push* operation which reduces this ratio by distributing a fraction  $\alpha$  of it to  $\mathbf{p}[u]$  (the approximate PageRank of  $u$ ), and the remaining fraction back to  $\mathbf{r}[u]$  and the residual error of neighbors of  $u$ . In line 2,  $\mathcal{S}_j^{\mathbf{p}}$  is defined as the node set which contains first  $j$  nodes ranked by the residual error/degree ratio  $\frac{\mathbf{r}[u]}{d[u]}$  in descending order. *Conductance*  $\Phi(\mathcal{S}_j^{\mathbf{p}})$  is the percentage of the linked number of nodes (i.e.,  $|S_j^{\mathbf{p}}|$ ) in  $S_j^{\mathbf{p}}$  normalized by the linked number of nodes in the rest of graph  $\mathcal{G}$  (i.e.,  $|\mathcal{V}| - |S_j^{\mathbf{p}}|$ ).  $|Supp(\mathbf{p})| = \{v | \mathbf{p}(v) \neq 0\}$  is the support of the distribution  $\mathbf{p}$ . Finally, we aim to find  $\mathcal{S}_j^{\mathbf{p}}$  that could minimize the conductance and return  $\hat{\mathcal{G}}$  consisting of the set of nodes  $v \in \mathcal{S}_j^{\mathbf{p}}$ . Based on the proof in Andersen et al. (2006), when *Meta-path Dependent PageRank-Nibble* terminates, for any  $u$ , the error  $\mathbf{q}[u] - \mathbf{p}[u]$  is bounded by  $\epsilon |\mathcal{N}(u)|$ , where  $\mathbf{q}[u]$  is the PageRank vector of  $u$  and  $\mathcal{N}(u)$  denotes the neighbors of  $u$ . For any graph, a good approximation can thus be guaranteed if  $\mathcal{N}(u)$  is bounded.

We also satisfy the following efficiency bound proved in Andersen et al. (2006):

**Theorem 1** *Let  $u_i$  be the  $i$ -th node pushed by Meta-path Dependent PageRank-Nibble. Then,  $\sum_i |\mathcal{N}(u_i)| < \frac{1}{\alpha \epsilon}$ .*

**Input** : A graph  $\mathcal{G}$ , a meta-path  $\mathcal{P}$ , a node set  $\hat{\mathcal{V}}$ , and two parameters:  $\alpha$  and  $\epsilon$ .  
**Output**: A compact graph  $\hat{\mathcal{G}}$  of a large graph  $\mathcal{G}$  that contains the given node set  $\hat{\mathcal{V}}$ .

- 1 Compute an approximate PageRank vector  $\mathbf{p}$  with residual vector  $\mathbf{r}$  initialized with function  $\mathcal{X}_{\hat{\mathcal{V}}}$  according to the given node set  $\hat{\mathcal{V}}$ , satisfying  $\max_{u \in \mathcal{V}} \frac{\mathbf{r}[u]}{d[u]} \leq \epsilon$  following (Andersen et al. 2006). The random walk terminates when meeting the entities not included in the given meta-path  $\mathcal{P}$ .
- 2 Check each set  $\mathcal{S}_j^{\mathbf{p}}$  with  $j \in [1, |\text{Supp}(\mathbf{p})|]$ , to see if the *conductance*:  $\Phi(\mathcal{S}_j^{\mathbf{p}})$  is the smallest one.
- 3 Return  $\hat{\mathcal{G}}$  that contains all the nodes  $v \in \mathcal{S}_j^{\mathbf{p}}$ . Otherwise, return  $\emptyset$ .

**Algorithm 1:** *Meta-path Dependent PageRank-Nibble*( $\mathcal{G}, \mathcal{P}, \hat{\mathcal{V}}, \alpha, \epsilon$ ).

Notice that initially  $\|\mathbf{r}\|_1 = 1$  and  $\|\mathbf{r}\|_1$  is decreased at  $i$ -th push by  $\alpha \epsilon |\mathcal{N}(u_i)|$ . Thus Theorem 1 can be proved. Then the following corollary holds.

**Corollary 1**  $\hat{\mathcal{G}}$  is generated by *Meta-path Dependent PageRank-Nibble* with no more than  $\frac{1}{\alpha \epsilon}$  edges.

Consequently, the bound holds independent of the size of the network. The complexity of this algorithm to find a cut is  $O(|\mathcal{E}| \log^4 |\mathcal{E}| / \Phi)$  where  $|\mathcal{E}|$  is the number of edges in the graph (Andersen et al. 2006). So this pruning strategy will work on very large networks, such as our specified world knowledge HIN.

After generating the local graph  $\hat{\mathcal{G}}_{\mathcal{P}}$  for meta-path  $\mathcal{P}$ , we compute the commuting matrix  $\mathbf{M}_{\mathcal{P}}$  for each meta-path  $\mathcal{P}$  based on the local graph. Notice that we only consider the symmetric meta-paths, it is easy to see that the commuting matrix can be decomposed. For example, suppose the meta-path is  $\mathcal{P} = (\mathcal{P}_l \mathcal{P}_l^{-1})$  where  $\mathcal{P}_l^{-1}$  is the reverse path of  $\mathcal{P}_l$ . Then the commuting matrix is  $\mathbf{M}_{\mathcal{P}} = \mathbf{M}_{\mathcal{P}_l} \mathbf{M}_{\mathcal{P}_l^{-1}}$ , where  $\mathbf{M}_{\mathcal{P}_l}$  and  $\mathbf{M}_{\mathcal{P}_l^{-1}} = \mathbf{M}_{\mathcal{P}_l}^T$  are the commuting matrices for  $\mathcal{P}_l$  and  $\mathcal{P}_l^{-1}$ . Thus, only  $\mathbf{M}_{\mathcal{P}_l}$  is needed to be precomputed and stored.

The meta-paths are then generated in the following steps.

1. Given a maximum length  $L$  of the symmetric meta-path  $\mathcal{P} = (\mathcal{P}_l \mathcal{P}_l^{-1})$ , enumerate all  $\mathcal{P}_l$  within  $\lceil L/2 \rceil$  consisting of different orders of entity types in  $\{\mathcal{E}^1\}_{l=1}^T$  connected. The resulting meta-path set is denoted as  $\mathbf{P} = \{\mathcal{P}\}$ .
2. For each meta-path  $\mathcal{P} \in \mathbf{P}$ :
  - (a) Generate the corresponding local graph  $\hat{\mathcal{G}}_{\mathcal{P}}$  based on the *Meta-path Dependent PageRank-Nibble* given the node set  $\hat{\mathcal{V}} = \{d \in \mathcal{D}\}$ .
  - (b) Compute the commuting matrices for  $\mathcal{P}_l$  and store the commuting matrices.

To summarize, we have outlined an efficient local partitioning method based on personalized PageRank. Besides, we precompute (i.e., offline) and store all the commuting matrices for  $\mathcal{P}_l$  where  $\mathcal{P} = (\mathcal{P}_l \mathcal{P}_l^{-1}) \in \mathbf{P}$ . We will see that the precomputed commuting matrices based on the local graph are very useful for meta-path selection and for computation of meta-path based similarity measure.

## 5 Application: texts as HINs

In this section, we introduce our application of how to generate HIN for the documents based on world knowledge bases and the text clustering and classification settings and algorithms.

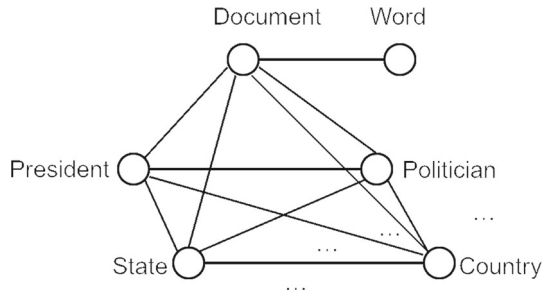
### 5.1 Construction of document HIN

Document similarity is a fundamental task, and can be used in many applications such as document classification, clustering and, ranking. Traditional approaches use bag-of-words (BOW) as document representation and compute the document similarities using different measures such as cosine, Jaccard, and dice. However, the entity phrases rather than just words in documents can be critical for evaluating the relatedness between texts. For example, “New York” and “New York Times” represent different meanings. “George Washington” and “Washington” are similar if they both refer to person, but can be rather different otherwise. If we can detect their names and types (coarse-grained types such as person, location, and organization; fine-grained types such as politician, musician, country, and city), they can help us better evaluate whether two documents are similar. Moreover, the links between entities or words are also informative. For example, the similarity between two documents can be zero if we use BOW representation since there is no identical word shared by them. However, the two documents are related in contents. If we can build a link between “Obama” of type *Politician* in one document and “Clinton” of type *Politician* in another, then the two documents become similar in the sense that they both talk about politicians and connect to “United States.” Therefore, we can use the structural information in the unstructured documents to further improve document similarity computation.

#### 5.1.1 Unsupervised semantic parsing

Semantic parsing is the task of mapping a piece of natural language text to a formal meaning representation (Mooney 2007). In our application, semantic parsing is used to ground a piece of text to a knowledge base with entities and their relations.

As an example, given the text “Obama is the president of United States of America,” “Obama” and “United States of America” are mapped to a knowledge base, resulting in two unary logic forms *People.BarackObama* and *Country.USA*, where *People* and *Country* are the type information in Freebase. Then it uses some grammar to combine the basic logic forms to generate the restricted logic forms below (Berant et al. 2013). For this example, *People.BarackObama*  $\wedge$  *President.USA* is generated to represent its semantic meaning. Notice that, *President.USA* is generated by joining the unary *Country.USA* with the binary *PresidentofCountry*, where *PresidentofCountry* is also a fact in the knowledge base. When there is more than one candidate semantic meaning for a sentence being generated, we constrain the entities to the spanning phrase with the maximum length recognized by a state-of-the-art named-entity recognition tool (Ratinov and Roth 2009).



**Fig. 1** The schema of a document HIN where the specified knowledge is represented in the form of a heterogeneous information network. The schema contains multiple entity types: document  $\mathcal{D}$ , word  $\mathcal{W}$ , named entities  $\{\mathcal{E}^1\}_{I=1}^T$ , and the relation types connecting the entity types. In this figure, we have entity types: *President*, *State*, *Country*, and *Politician*

### 5.1.2 Semantic filtering

For each sentence in a given document, the output of semantic parsing is a set of logic forms that represent the semantic meaning. However, the extracted entities can be ambiguous. For example, “apple” may be associated with type *Company* or *Fruit*. Therefore, we should filter out the noisy entities and their types to ensure that the knowledge we have is clean. We assume that in the domain specific tasks, given the context, the entities seldom have multiple meanings. Three methods have been introduced for semantic filtering (Wang et al. 2015a). We use conceptualization-based semantic filter in our experiment, because it performs the best among the methods. We assume that the type that can best fit the context is the correct semantic meaning. Motivated by the approaches of conceptualization (Song et al. 2011, 2015) and entity disambiguation (Li et al. 2013), we represent each entity with a feature vector of entity types, and use standard  $k$ -means to cluster the entities. In each cluster, we use the intersection operation to find the most likely entity type for the entities in the cluster. In this case, different entities can be used to disambiguate each other, and the entities that conflict with others will be removed.

## 5.2 An overview of document HIN

The output of semantic parsing and semantic filtering is the document associated with not only the entities but also the types and relations. In addition to the named entities, document and word are also regarded as two types. Following Wang et al. (2015a), we use the network schema the data. The network contains multiple entity types: *document*  $\mathcal{D}$ , *word*  $\mathcal{W}$ , *named entities*  $\{\mathcal{E}^1\}_{I=1}^T$ , and *relation types* connecting the *entity types*. Different from Wang et al. (2015a) which uses coarse-grained entity types such as *Person*, *Location*, and *Organization* to construct HIN, we prefer to use more fine-grained entity types, such as *Politician*, *Musician*, and *President* since they provide refined semantics to represent document similarity. We show an example of the document HIN schema in Fig. 1. However, in Freebase, there are about 1500+ entity types and 3500+ relation types, which will generate an exponential number of

meta-paths. In previous work (Sun et al. 2011, 2012), meta-paths are provided by users, which is doable for networks with simple schema consisting of several types of entities and relations, such as the DBLP network (five entity types and four relation types). It is unrealistic to ask a user to specify meta-paths for a network with a large number of entities and relations. An automatic mechanism should be developed to generate all the interested meta-paths.

By representing the world knowledge in HIN, two documents can be linked together via many meta-paths. For example, if two documents are linked by the meta-path  $Document \xrightarrow{\text{contain}} Politician \xrightarrow{\text{presidentOf}} Country \xrightarrow{\text{presidentOf}^{-1}} Politician \xrightarrow{\text{contain}^{-1}} Document$ , the number of the corresponding meta-path instances can be used to measure the similarity between the two documents, which cannot be captured by the original *bag-of-words* feature. Assuming that similar documents are structurally similar defined by symmetric meta-paths, we only explore symmetric meta-paths. The calculation based on the meta-paths is to compute all the corresponding commuting matrices of interests. Consequently, the size of network brings a critical issue since it is impossible to compute all the commuting matrices and load them into memory. To make the method practical, we propose two ways to prune this computation: (i) prune the large network to generate a more compact graph for the interested commuting matrices calculation (Sect. 4), and (ii) use unsupervised feature selection approaches to select semantically meaningful meta-paths for final document similarity computation (Sect. 3).

### 5.3 Spectral clustering of texts

To check the quality of different similarity measures in the real application scenario, we further use similarity matrices generated above as the weight matrix in the spectral clustering (Zelnik-manor and Perona 2005) for document clustering task. We compare the performance of clustering results of using three different KnowSim-based similarity matrices with using the similarity matrices generated by other similarity measures. We set the number of clusters as 20 and 16 for 20NG and GCAT according to their ground-truth labels, respectively. We employ the widely-used normalized mutual information (NMI) (Strehl and Ghosh 2003) as the evaluation measure. The NMI score is 1 if the clustering results match the category labels perfectly and 0 if the clusters are obtained from a random partition. In general, the larger the scores, the better the clustering results.

### 5.4 HIN-kernel SVM classification using KnowSim matrix

We also apply the state-of-the-arts classification models in the document classification tasks, including Naive Bayes (NB) and SVM. In particular, we use the HIN-links based text classification framework proposed in Wang et al. (2016a), to incorporate HIN links into the traditional classification models as  $NB^{HIN}$  and  $SVM^{HIN}$  respectively. We denote a set of training examples as  $\mathcal{X} = \{\mathbf{x}_i : i \in \{1, 2, \dots, n\}\}$ , and the corresponding labels as  $\mathbf{y} = \{y_i \in \mathcal{Y} : i \in 1, 2, \dots, n\}$ .

$NB^{HIN}$ . Traditional Naive Bayes classifier for text classification is formulated as:

$$P(y|\mathbf{x}^{\mathcal{V}}) = \frac{P(y) \prod P(x^{\mathcal{V}}|y)}{\sum P(y) \prod P(x^{\mathcal{V}}|y)}. \quad (4)$$

where  $x^{\mathcal{V}}$  represent a feature in entity<sup>1</sup> feature vector  $\mathbf{x}^{\mathcal{V}}$  of document  $d$ .

We also incorporate the links into Naive Bayes model:

$$P(y|\mathbf{x}^{\mathcal{E}}) = \frac{P(y) \prod P(x^{\mathcal{E}}|y)}{\sum P(y) \prod P(x^{\mathcal{E}}|y)}. \quad (5)$$

Then the combined estimation function is:

$$\hat{y} = \arg \max_{y \in \mathcal{Y}} P(y) \prod P(x^{\mathcal{V}}|y) \prod P(x^{\mathcal{E}}|y). \quad (6)$$

$SVM^{HIN}$ . Let matrix  $\mathbf{X}$  be the matrix where  $\mathbf{X}_i = \mathbf{x}_i^T$ , matrix  $\mathbf{Y} = \text{diag}(\mathbf{Y})$ , vector  $\mathbf{1}$  be an  $n$ -dimensional vector of all ones and  $C$  be a positive trade-off parameter. Then, the dual formulation of 1-norm soft margin SVM is given by

$$\begin{aligned} \max_{\alpha} \mathbf{1}^T \alpha - \frac{1}{2} \alpha^T \mathbf{Y} (\mathbf{X}^T \mathbf{X}) \mathbf{Y} \alpha \\ \text{s.t. } \mathbf{Y}^T \alpha = 0, 0 \leq \alpha \leq C \mathbf{1}. \end{aligned} \quad (7)$$

Here we have  $\mathbf{X}_i$  equals to  $[\mathbf{x}_i^{\mathcal{V}T}, \mathbf{x}_i^{\mathcal{E}T}]^T$  in  $SVM^{HIN}$ . By doing so,  $SVM^{HIN}$  provides a simple way to combine the structured information with traditional features. To learn the  $SVM^{HIN}$ , we use a convex quadratic programming to solve the dual problem in Eq. (7).

We follow Wang et al. (2016a) to encode the KnowSim matrix as kernel in SVM as below. We use  $\mathbf{K}$  to present the KnowSim kernel matrix. Suppose that  $\mathbf{K}$  is positive semi-definite (PSD). Let matrix  $\mathbf{Y} = \text{diag}(\mathbf{Y})$ , vector  $\mathbf{1}$  be an  $n$ -dimensional vector of all ones and  $C$  be a positive trade-off parameter. Then the dual formulation of 1-norm soft margin SVM is given by

$$\begin{aligned} \max \mathbf{1}^T \alpha - \frac{1}{2} \alpha^T \mathbf{Y} \mathbf{K} \mathbf{Y} \alpha \\ \text{s.t. } \mathbf{Y}^T \alpha = 0, 0 \leq \alpha \leq C \mathbf{1}. \end{aligned} \quad (8)$$

When  $\mathbf{K}$  is PSD, the above problem is a convex quadratic program and solved effectively.

However, the KnowSim matrix  $\mathbf{K}$ , where  $\mathbf{K}_{ij} = K S(d_i, d_j)$ , may not be PSD (Berg et al. 1984). We use  $\mathbf{K}_0$  ( $\mathbf{K}_{0ij} = K S(d_i, d_j)$ ) to present the indefinite kernel matrix generated by KnowSim. Luss and d'Aspremont (2008) proposed a saddle (min-max)

<sup>1</sup> Note that in the HIN for text, entity features include both tf of words and named entities.

approach to simultaneously learn a proxy PSD kernel matrix  $\mathbf{K}$  for the indefinite matrix  $\mathbf{K}_0$  and the SVM classification as follow:

$$\min_{\mathbf{K}} \max_{\alpha} \mathbf{1}^T \alpha - \frac{1}{2} \alpha^T \mathbf{Y} \mathbf{K} \mathbf{Y} \alpha + \rho \|\mathbf{K} - \mathbf{K}_0\|_F^2$$

$$s.t. \quad \mathbf{Y}^T \alpha = 0, 0 \leq \alpha \leq C \mathbf{1}, \mathbf{K} \succeq 0. \quad (9)$$

Let  $\mathcal{Q} = \{\alpha \in \mathbb{R}^n : \mathbf{Y}^T \alpha = 0, 0 \leq \alpha \leq C \mathbf{1}\}$ ,  $F(\alpha, \mathbf{K}) = \mathbf{1}^T \alpha - \frac{1}{2} \alpha^T \mathbf{Y} \mathbf{K} \mathbf{Y} \alpha + \rho \|\mathbf{K} - \mathbf{K}_0\|_F^2$ . The parameter  $\rho > 0$  controls the magnitude of the penalty on the distance between  $\mathbf{K}$  and  $\mathbf{K}_0$ . If any matrix  $\mathbf{A}$  is PSD, we write it as  $\mathbf{A} \succeq 0$ . Based on the min-max theorem (Boyd and Vandenberghe 2004), Eq. (9) equals to  $\max_{\alpha \in \mathcal{Q}} \min_{\mathbf{K} \succeq 0} F(\alpha, \mathbf{K})$ . Thus the objective function is represented as

$$J(\alpha) = \min_{\mathbf{K} \succeq 0} F(\alpha, \mathbf{K}). \quad (10)$$

To learn about how to establish the differentiability of the objective function and what optimization algorithm is used for the objective function, please refer to Wang et al. (2016a) for more details.

## 6 Experiments

This section reports our experiments which demonstrate the effectiveness and efficiency of our approach to measuring document similarity.

### 6.1 Datasets

We use the following two benchmark datasets to evaluate the document similarity task. *20Newsgroups (20NG)* The 20newsgroups dataset (Lang 1995) contains about 20,000 newsgroups documents evenly distributed across 20 newsgroups.<sup>2</sup>

*RCV1* The RCV1 dataset is a dataset containing manually labeled newswire stories from Reuter Ltd (Lewis et al. 2004). The news documents are categorized with respect to three controlled vocabularies: industries, topics and regions. There are 103 categories including all nodes except for root in the topic hierarchy. The maximum depth is four, and 82 nodes are leaves. We select top category GCAT (Government/Social) to form the document similarity task. In total, we have 60,608 documents with 16 leaf categories.

The ground-truth of document similarity is generated as follows: If two documents are in the same group or the same leaf category, their similarity equals to 1; otherwise, it is 0.

<sup>2</sup> <http://qwone.com/~jason/20Newsgroups/>.

**Table 1** Statistics of entities in different datasets with semantic parsing and filtering using Freebase

	#(Document)	#(Word)	#(FBEntity)	#(Total)	#(Types)
20NG	19,997	60,691	28,034	108,722	2615
GCAT	60,608	95,001	110,344	265,953	2665

#(Document) is the number of all documents; similar for #(Word) (# of words), #(FBEntity) (# of Freebase entities), #(Total) (the total # of entities), and #Types (the total # of entity types)

## 6.2 World knowledge base

Then we introduce the knowledge base we use. In Wang et al. (2015a), the authors have demonstrated that Freebase is more effective compared to YAGO2, so we only use Freebase as our world knowledge source in this experiment.

*Freebase* Freebase<sup>3</sup> is a publicly available knowledge base consisting of entities and relations collaboratively collected by its community members. So far, it contains over 2 billions relation expressions between 40 millions entities. Moreover, there are 1500+ entity types and 3500+ relation types in Freebase. We convert a logical form generated by unsupervised semantic parser into a SPARQL query and execute it on our copy of Freebase using the Virtuoso engine.

After performing semantic parsing and filtering, the numbers of entities in different document datasets with Freebase are summarized in Table 1. The numbers of relations (logical forms parsed by semantic parsing and filtering) in 20NG and GCAT are 9,655,466 and 18,008,612, respectively. We keep 20 and 43 entity types for 20NG and GCAT respectively, because they have relatively larger number of instances. Then 325 and 1682 symmetric meta-paths are generated based on the MDPN algorithm, for 20NG and GCAT respectively. We can save around 3.8 and 19.6 h for the corresponding datasets. The reason is that MDPN shares the similar nature with PageRank-Nibble, which is that the running time is independent of the size of the graph. Similar result is found when comparing the space usage. By using MDPN, we can save up to 1.4G storage (15.2%) compared to storing the exact commuting matrices. In our real setting, we can save 45.5G and 235.5G storage for 20NG and GCAT datasets, respectively, because MDPN only saves the nodes that have relatively high degree, which is important in sparse matrix.

## 6.3 Similarity results

In this experiment, we compare the performance of our document similarity measure, KnowSim, with three representative similarity measures: cosine, Jaccard, and dice. We denote *KnowSim* + *UNI*, *KnowSim* + *MST* and *KnowSim* + *LAP* as KnowSim with uniform weights, weights determined by MST and Laplacian score-based methods introduced in Sect. 3.1. Following (Wang et al. 2015a), we use the specified world knowledge as features to enhance cosine, Jaccard, and dice. The feature settings are defined as follows.

<sup>3</sup> <https://developers.google.com/freebase/>.



- BOW: Traditional bag-of-words model with the tf weighting mechanism.
- BOW + TOPIC: BOW integrated with additional features from topics generated by LDA (Blei et al. 2003). According to the number of domains that 20NG and GCAT have, we assign 20 topics and 16 topics to 20NG and GCAT, respectively.
- BOW + ENTITY: BOW integrated with additional features from entities in specified world knowledge from Freebase.
- BOW + TOPIC + ENTITY: BOW integrated with additional features from both topics generated by LDA and entities in specified world knowledge from Freebase.

We employ the widely-used correlation coefficient as the evaluation measure. The correlation coefficient is defined as  $r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}$ , where  $\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$  and  $\bar{y} = \frac{\sum_{i=1}^n y_i}{n}$ , and  $n$  equals to the total number of  $x_i$  or  $y_i$ . The correlation score is 1 if the similarity results match the ground-truth perfectly and 0 if the similarity results are random. In general, the larger the scores, the better the similarity results.

In Table 2, we show the performance of all the similarity measures with different experimental settings on both 20NG and GCAT datasets. Overall, among all the methods we test, KnowSim + LAP consistently performs the best. The reason is that Laplacian score could discriminate documents from different clusters, which is strongly correlated to our similarity task. We can also see that KnowSim + UNI, KnowSim + MST, and KnowSim + LAP outperform all the other similarity measures, including the similarity measures with specified world knowledge as flat features (BOW + ENTITY). This means that by using structural information in HIN extracted from the world knowledge, we can improve the document similarity, especially comparing with just using them as flat features. Also, KnowSim-based similarity measures perform better than the similarity measures with feature setting “BOW + TOPIC.” The reason is again world knowledge could provide the structural information between documents rather than using the flat topic distribution. In addition, one can also see that KnowSim + UNI performs relatively weaker than the other two KnowSim with weighted meta-paths. This means that our meta-path weighting methods do help find the important link information (i.e., meta-paths) related to certain domains. Moreover, we find the improvement of KnowSim on GCAT is more than that on 20NG. As Table 1 shows, GCAT has more entities and associated types specified by the world knowledge. This means that the more world knowledge we can find or use in the documents, the better improvement in the document similarity task. This also hints us that if we could improve the precision and coverage of the world knowledge base, we could further improve the performance.

## 6.4 Analysis of meta-path length

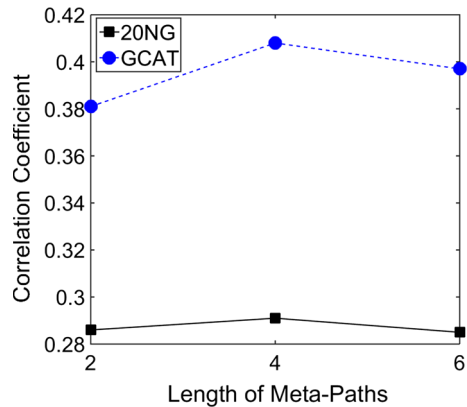
We also evaluated the effect of varying the length of meta-paths in  $\mathbf{P}$  of KnowSim + LAP on the document similarity task since KnowSim + LAP performs the best among all the KnowSim-based similarity measures. Figure 2 shows the results of similarity with different lengths of meta-paths in KnowSim+LAP on both 20NG and GCAT datasets.

**Table 2** Correlation coefficient of different similarity measures on 20NG and GCAT

Similarity measures	Datasets Settings	20NG	GCAT
Cosine	BOW	0.2400	0.3490
	BOW + TOPIC	0.2713	0.3639
	BOW + ENTITY	0.2473	0.2737
	BOW + TOPIC + ENTITY	0.2768	0.3128
Jaccard	BOW	0.2352	0.3313
	BOW + TOPIC	0.2632	0.3460
	BOW + ENTITY	0.2369	0.2319
	BOW + TOPIC + ENTITY	0.2650	0.2991
Dice	BOW	0.2400	0.3490
	BOW + TOPIC	0.2712	0.3638
	BOW + ENTITY	0.2474	0.2776
	BOW + TOPIC + ENTITY	0.2767	0.3156
KnowSim	KnowSim + UNI	0.2860	0.3815
	KnowSim + MST	0.2891	0.3833
	KnowSim + LAP	0.2913 (+ 5.2%)	0.4086 (+ 12.3%)

“BOW” represents bag-of-words as features; “BOW + TOPIC” represents bag-of-words plus topics generated by LDA as features; “BOW + ENTITY” represents bag-of-words plus entities as features; “BOW + TOPIC + ENTITY” represents bag-of-words plus topics plus entities as features

**Fig. 2** Effect of the length of meta-paths on document similarity on KnowSim + LAP for 20NG and GCAT datasets



The length of symmetric meta-paths equals to 2, 4 and 6.<sup>4</sup> It is shown that for both datasets, longer meta-path may not result in better correlation coefficient. We achieve the best performance as shown in Table 2 with a weighted combination of meta-paths with length 2 and length 4. We can see that longer meta-path may cause semantic drift. Moreover, the information carried in the meta-paths need to be combined to express the similarity between documents, which is captured by KnowSim and matches human intuition.

<sup>4</sup> We do not test longer meta-paths, because Theorem 2 in Sun et al. (2011) has demonstrated their limit on effective propagation of semantic knowledge.

**Table 3** NMI of clustering on 20NG and GCAT using the similarity matrix generated by different similarity measures

Similarity matrix sources	Datasets Settings	20NG	GCAT
Cosine	BOW	0.3440	0.3932
	BOW + TOPIC	0.3461	0.4352
	BOW + ENTITY	0.3896	0.4294
	BOW + TOPIC + ENTITY	0.4247	0.4106
Jaccard	BOW	0.3547	0.3887
	BOW + TOPIC	0.3517	0.4292
	BOW + ENTITY	0.3850	0.4197
	BOW + TOPIC + ENTITY	0.4293	0.4159
Dice	BOW	0.3440	0.3932
	BOW + TOPIC	0.3457	0.4355
	BOW + ENTITY	0.3894	0.4291
	BOW + TOPIC + ENTITY	0.4248	0.4112
KnowSim	KnowSim + UNI	0.4304	0.4463
	KnowSim + MST	0.4412	0.4653
	KnowSim + LAP	0.4461 (+ 3.9%)	0.4736 (+ 8.8%)

“BOW” represents bag-of-words as features; “BOW + TOPIC” represents bag-of-words plus topics generated by LDA as features; “BOW + ENTITY” represents bag-of-words plus entities as features; “BOW + TOPIC + ENTITY” represents bag-of-words plus topics plus entities as features

## 6.5 Spectral clustering using KnowSim matrix

As shown in Table 3, we illustrate the performance of all the clustering results with different similarity matrices on both 20NG and GCAT datasets. The NMI is the average NMI of five random trials per experiment setting. Among all the methods we tested, spectral clustering with KnowSim + LAP matrix performs the best, which is consistent with the similarity correlation results (Table 2). Moreover, all of the KnowSim similarity matrix-based clustering results consistently outperform the other methods. Note that the three KnowSim-based matrices produce higher NMI compared to that with “BOW + ENTITY,” which means using the meta-path as link information in the similarity matrix, the link information can be passed to the clustering results, where the link information can be very useful to group similar documents in the same cluster. We can infer that KnowSim could have positive impact on other similarity-based applications, e.g., document classification and ranking.

## 6.6 HIN-kernel SVM classification using KnowSim matrix

In order to show the effectiveness of the KnowSim kernel, besides the above clustering experiments, we show experiments on using KnowSim in document classification task here. To conduct deeper analysis of the effects of KnowSim to the task, as well as show the scalability of the KnowSim usage in similarity based applications, in the

spirit of Basu et al. (2004), we develop more document sub-datasets based on 20NG and GCAT as below. From 20NG, *20NG-SIM* consists of three newsgroups on similar topics (comp.graphics, comp.sys.mac.hardware, and comp.os.ms-windows.misc) with significant overlap among the groups; *20NG-DIF* consists of articles in three newsgroups that cover different topics (rec.autos, comp.os.ms-windows.misc, and sci.space) with well separated categories. From GCAT, *GCAT-SIM* consists of articles from three leaf categories of similar topics [GWEA (Weather), GDIS (Disasters), and GENV (Environment and Natural World)] with significant overlap among the categories. We have 1014, 2083, and 499 documents for the three categories respectively. *GCAT-DIF* consists of three leaf categories that cover different topics [GENT (Arts, Culture, and Entertainment), GODD (Human Interest), and GDEF (Defense)] with well separated categories. We have 1062, 1096, and 542 documents for the three categories respectively.

We analyze the performance of our classification methods here.

We first evaluate the effectiveness of the HIN-links based classification by comparing  $NB^{HIN}$  and  $SVM^{HIN}$  with traditional Naive Bayes and SVM. The feature settings regarding to the NB and SVM are defined as follows. “BOW” and “BOW + ENTITY” are the same as the setting in similarity results. Additionally, we add  $WE_{Avg}$ : we use Word2Vec (Mikolov et al. 2013) to train the word embedding based on the 20NG and GCAT respectively. We then use the average word vectors as features to feed them to the classifiers. We set the window size as 5, and the learned word representation is of 400 dimensions using CBOW model and hierarchical softmax for fast training.

$NB^{HIN}$  and  $SVM^{HIN}$  are the HIN-links based text classification algorithms. The entity features and relation features are constructed the same as Wang et al. (2016a). We experiment on the four datasets above. Each data split has three binary classification tasks. For each task, the corresponding data is randomly divided into 80% training and 20% testing data. We apply 5-fold cross validation on the training set to determine the optimal hyperparameter  $C$  for SVM and  $SVM^{HIN}$ . Then all the classification models are trained based on the full training set (SVM based methods with  $C$ ), and tested on the test set. We employ classification accuracy as the evaluation measure.

In Table 4, we show the performance of all the classification models with different settings on all the four datasets. We report the average classification accuracy of the three binary classification results in each dataset of the four. Notice that here we focus on  $NB^{HIN}$  versus NB and  $SVM^{HIN}$  versus SVM to directly test our general classification framework. From the results, we find that  $NB^{HIN}$  and  $SVM^{HIN}$  are competitive with NB and SVM with  $WE_{Avg}$ , and outperform NB and SVM with other settings. This means that by using link information in HIN extracted from the world knowledge (specifically refer to relation features), we can improve the text classification, especially comparing with the ones only using entity as additional features (BOW+ENTITY). The results are even competitive with the state-of-the-art word embedding approach trained based on 20NG and GCAT data respectively. Also, we find the improvement of  $SVM^{HIN}$  and  $NB^{HIN}$  on GCAT-SIM and GCAT-DIF are more than that on 20NG-SIM and 20NG-DIF. This is because that GCAT-SIM and GCAT-DIF have more entities and associated types grounded from Freebase.

**Table 4** Performance of different classification algorithms on 20NG-SIM, 20NG-DIF, GCAT-SIM, and GCAT-DIF datasets

Methods	Datasets Settings	20NG-SIM (%)	20NG-DIF (%)	GCAT-SIM (%)	GCAT-DIF (%)
NB	BOW	86.95	96.37	88.49	86.73
	BOW + ENTITY	89.76	96.94	89.12	88.08
	WE <sub>Avg</sub>	90.82	97.16	91.87	91.56
NB <sup>HIN</sup>		90.83	97.37	90.02	88.65
SVM	BOW	90.81	96.66	94.15	88.98
	BOW + ENTITY	91.11	96.90	94.29	90.18
	WE <sub>Avg</sub>	91.67	98.27	96.81	90.64
SVM <sup>HIN</sup>		91.60	97.20	94.82	91.19
SVM <sup>HIN</sup> + KnowSim	DWD	92.32	97.83	95.29	90.70
	DWD + MP	92.68	98.01	96.04	91.88
	DWD	92.65	98.13	95.63	91.63
IndefSVM <sup>HIN</sup> + KnowSim	DWD	92.65	98.13	95.63	91.63
	DWD+MP	93.38	98.45	98.10	93.51

BOW and ENTITY represent bag-of-words feature and the entities generated by the world knowledge specification framework based on Freebase, respectively. NB<sup>HIN</sup> and SVM<sup>HIN</sup> are the variant of traditional Naive Bayes and SVM under our HIN-links based text classification framework. SVM<sup>HIN</sup> + KnowSim represents the 1-norm soft margin SVM defined in Eq. (8) with indefinite KnowSim based kernel. IndefSVM<sup>HIN</sup> + KnowSim represents the SVM with a proxy PSD kernel for the indefinite KnowSim matrix as shown in Eq. (9). DWD and DWD + MP represent the kernel matrix that is constructed based on KnowSim with a single DWD meta-path and all kinds of meta-paths generated based on the text HIN, respectively

We next test the performance of the KnowSim kernel methods by comparing them with the other classification methods (i.e., SVM<sup>HIN</sup> and NB<sup>HIN</sup>). We follow Wang et al. (2016a) to derive two SVM with KnowSim kernel methods.

- One is denoted as “SVM<sup>HIN</sup> + KnowSim” using the 1-norm soft margin SVM defined in Eq. (8) by setting the negative eigenvalues of the KnowSim matrix being zeros.
- The other is denoted as “IndefSVM<sup>HIN</sup> + KnowSim.” It learns a proxy PSD kernel for the indefinite KnowSim matrix as shown in Eq. (9). The parameters  $C$  and  $\rho$  for indefinite SVM are tuned based on the 5-fold cross validation and the Nesterov’s efficient smooth optimization method (Nesterov 2005) is terminated if the value of the object function changes less than  $10^{-6}$  following (Ying et al. 2009).

We also explore what should be the best way to use KnowSim (Definition 6) as kernel matrix for the text classification. We particularly explore two different KnowSim computation settings.

- DWD. Kernel matrix is constructed based on KnowSim using only meta-path instances belonging to  $\mathcal{P}_{DWD} = Document \xrightarrow{\text{contain}} Word \xrightarrow{\text{contain}^{-1}} Document$  meta-path [i.e.,  $M' = 1$  in Eq. (3)]. This setting aims to test whether kernel methods themselves are

still effective, even with the simplest structural information in the HIN, when we have almost the same amount of information compared to bag-of-words features.

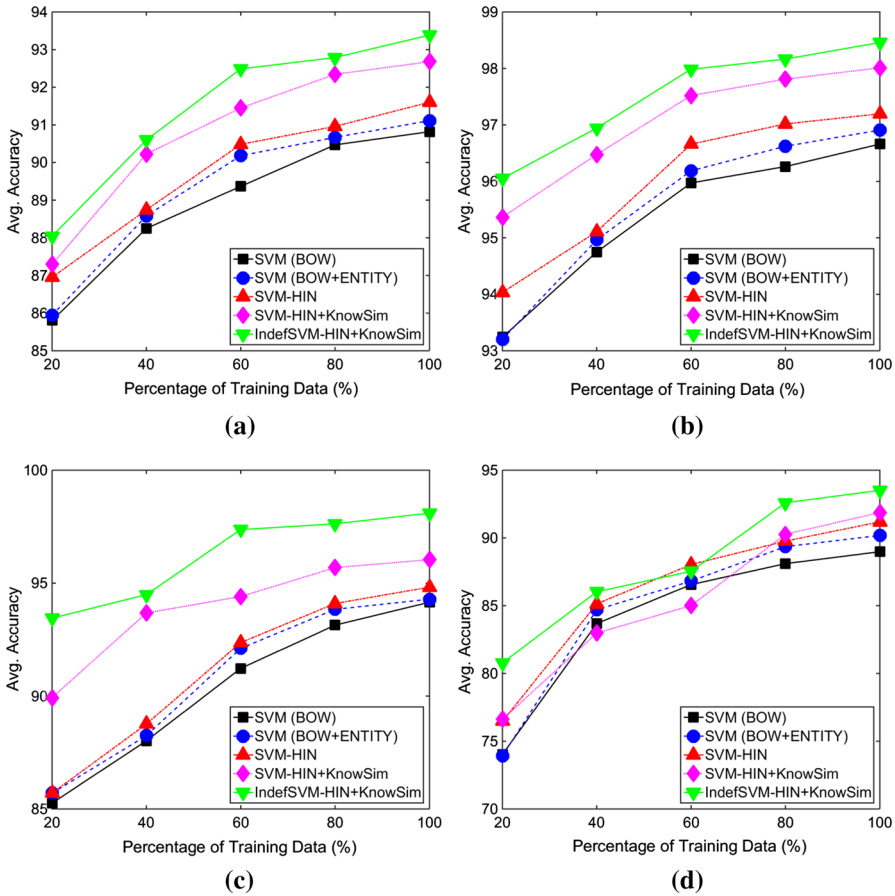
- DWD + MP. Kernel matrix is constructed based on KnowSim using meta-path instances belong to all kinds of meta-paths in the text HIN. This setting aims to test that how good can kernel based SVM leverage the specified world knowledge for text classification.

As shown in Table 4, IndefSVM<sup>HIN</sup> + KnowSim with DWD+MP consistently performs the best on all datasets. With t-test, we find the improvements are significant at 0.05 significance level. Especially, we can draw the following observations and conclusions.

- (1) The performance of SVM<sup>HIN</sup> + KnowSim with DWD is better than SVM with BOW. This is because in Eq. (3), there is a normalization term for the values in commuting matrix which is  $\mathbf{W}_{DW}\mathbf{W}_{DW}^T$  and  $\mathbf{W}_{DW}$  is the matrix between documents and words. The normalization terms in Eq. (3),  $|\{p_{i \rightsquigarrow i} \in \mathcal{P}_m\}|$  and  $|\{p_{j \rightsquigarrow j} \in \mathcal{P}_m\}|$ , correspond to the degree for the document node in the information network. Compare to Eq. (7) where no normalization is performed, it shows normalization indeed helps to formulate a better similarity. Note that, cosine similarity is another widely used approach for normalizing document length, but it cannot be applied to information network.
- (2) Both kernel methods with DWD + MP outperform NB<sup>HIN</sup> and SVM<sup>HIN</sup>. The reason is by considering the meta-path information as a whole and using some weighting mechanisms to select the more important meta-paths it indeed helps encode more informative information for text classification.
- (3) In both SVM<sup>HIN</sup> + KnowSim and IndefSVM<sup>HIN</sup> + KnowSim, DWD + MP is better than DWD. This indicates that meta-paths in HIN with knowledge (e.g., entities and relations) capture more similarity information for documents than just the links between documents via words.
- (4) IndefSVM<sup>HIN</sup> + KnowSim always works better than SVM<sup>HIN</sup> + KnowSim. The reason can be denoising the non-PSD kernel by removing the negative eigenvalues can lose some useful information about the similarity.
- (5) IndefSVM<sup>HIN</sup> + KnowSim with DWD+MP consistently outperforms classifiers with WE<sub>Avg</sub>. This means that KnowSim kernel with world knowledge carries more semantics about the similarities between texts compared to that the implicit embedding implies.

Moreover, we test the effectiveness of world knowledge for improving classification performance. We test on all the four classification datasets and vary the size of training data (20, 40, 60, 80, 100%) for each algorithm. The results are summarized in Fig. 3. In all the datasets, it seems that with less training data, the external knowledge encoded by document HIN similarity measure can consistently help improving the classification accuracy.

Besides using KnowSim, we also use the knowledge-based graph semantic similarity (GSim) proposed in Schuhmacher and Ponzetto (2014) to measure the document similarity. We use the indefinite SVM to encode the GSim similarity in the kernel.



**Fig. 3** Effects of the size of training data on the four classification datasets. SVM-HIN + KnowSim and IndefSVM-HIN + KnowSim denote  $SVM^{HIN} + KnowSim$  and  $IndefSVM^{HIN} + KnowSim$  with DWD + MP. **a** Effects of the size of training data on 20NG-SIM, **b** effects of the size of training data on 20NG-DIF, **c** effects of the size of training data on GCAT-SIM, **d** effects of the size of training data on GCAT-DIF

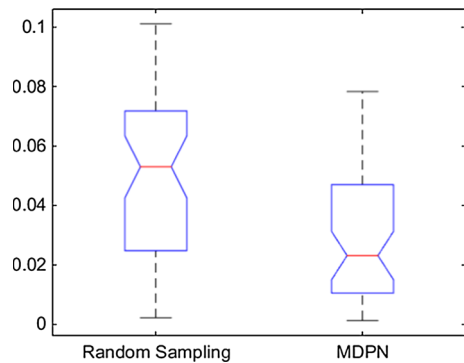
Instead of using DBpedia, we use Freebase. However, the time complexity of computing GSim is  $O(n^2 \cdot m^3)$ , where  $n$  denotes the size of the document dataset and  $m$  denotes the subgraph size which is proportional to the average number of entities in one document. In Freebase,  $m$  is much larger compared to DBpedia. So it is infeasible to run with the whole Freebase network. We thus implement GSim based on the text HIN, which is a subgraph of Freebase. We then achieve the accuracy of 50.44% on 20NG-SIM dataset. This indicates that (1) GSim may be not very suitable to be used in indefinite SVM; and (2) implementing GSim upon the text HIN, may lose important knowledge regarding to the documents.

We thus conclude that document HIN similarity measure: KnowSim is effective for document classification task.

**Table 5** Ten meta-paths sampled from 20NG dataset

$\mathcal{P}_1$	<i>Document</i> → <i>Baseball</i> → <i>Sports</i> → <i>Baseball</i> → <i>Document</i>
$\mathcal{P}_2$	<i>Document</i> → <i>Baseball</i> → <i>Olympics</i> → <i>Baseball</i> → <i>Document</i>
$\mathcal{P}_3$	<i>Document</i> → <i>Ice_hockey</i> → <i>Sports</i> → <i>Ice_hockey</i> → <i>Document</i>
$\mathcal{P}_4$	<i>Document</i> → <i>Ice_hockey</i> → <i>Olympics</i> → <i>Ice_hockey</i> → <i>Document</i>
$\mathcal{P}_5$	<i>Document</i> → <i>Astronomy</i> → <i>Location</i> → <i>Astronomy</i> → <i>Document</i>
$\mathcal{P}_6$	<i>Document</i> → <i>Computer</i> → <i>Cvg</i> → <i>Computer</i> → <i>Document</i>
$\mathcal{P}_7$	<i>Document</i> → <i>Religion</i> → <i>Government</i> → <i>Religion</i> → <i>Document</i>
$\mathcal{P}_8$	<i>Document</i> → <i>Religion</i> → <i>Organization</i> → <i>Religion</i> → <i>Document</i>
$\mathcal{P}_9$	<i>Document</i> → <i>Military</i> → <i>Government</i> → <i>Military</i> → <i>Document</i>
$\mathcal{P}_{10}$	<i>Document</i> → <i>Medicine</i> → <i>Government</i> → <i>Medicine</i> → <i>Document</i>

**Fig. 4** Notched Boxplot of Frobenius norm of approximation made by two commuting matrices (i.e., based on random sampling or Meta-path Dependent PageRank-Nibble) to the exact commuting matrices on 20NG dataset. The notch indicates the confidence interval of the median. With t-test, we found the difference between two methods is significant at 0.05 significance level



## 6.7 Approximate commuting matrix for meta-path

In order to verify the effectiveness of the Meta-path Dependent PageRank-Nibble (MDPN) algorithm shown in Sect. 4, we first sample ten meta-paths from the symmetric meta-path set  $\mathbf{P}$  as listed in Table 5 from 20NG. Second, for each meta-path, we use MDPN to generate the approximate commuting matrix. Besides, we also generate the exact commuting matrix  $\mathbf{M}_{\mathcal{P}}$  based on the original graph, and use random sampling to randomly select  $Z$  ( $1 \leq Z \leq |\mathbf{M}_{\mathcal{P}}|$ ) elements and set them to 0 based on the exact commuting matrix. Finally, we have 10 exact commuting matrices corresponding to 10 meta-paths, and for each matrix, we randomly sample 5 trials (in total, 50 iterations of sampling).

We first directly evaluate the effectiveness of MDPN by comparing the difference between the two approximate commuting matrices (i.e., generated based on our MDPN or by random sampling) and the exact commuting matrix. We use the Frobenius norm  $\|\mathbf{A} - \mathbf{B}\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |\mathbf{A}_{ij} - \mathbf{B}_{ij}|^2}$  to evaluate the difference between two matrices. As shown in Fig. 4, MDPN consistently performs well to reproduce the commuting matrix. The distribution of MDPN is relatively more stable and the standard deviation is smaller compared to that of random sampling.



**Table 6** The performance loss ratio of using KnowSim+LAP with the Meta-path Dependent PageRank-Nibble based commuting matrices compared to that with the exact matrices

Correlation coefficient	NMI
- 0.23%	- 1.36%

Next, we indirectly evaluate the effectiveness of MDPN algorithm, by comparing the correlation coefficient of KnowSim + LAP using the above selected ten approximate commuting matrices generated by MDPN and the corresponding exact commuting matrices. The difference of correlation coefficient is shown in Table 6. We can see the difference is relatively small (- 0.23%), compared to the improvement (+ 12.3%) we have achieved (Table 2) against the best one among the other similarity measures (Cosine with “BOW + TOPIC”). We find that such PageRank based local graph carries the necessary information for KnowSim, because we have the nodes with more linked nodes (neighbor nodes) in the local graph, where the advantage of KnowSim is actually the ability to use such link information.

Then we use spectral clustering results that applying the similarity matrix generated by KnowSim to demonstrate the difference between the approximate commuting matrix generated by MDPN and the exact commuting matrix. As also shown in Table 6, we can find the similar results with the correlation results. The performance loss of using similarity matrix generated by KnowSim as approximate matrix can be ignored (- 1.36% loss) when compared to the gain of our method (8.8%) as shown in Table 3.

Finally, we compare the execution time and space usage of using MDPN based local graph, and using original graph for commuting matrix generation and storage. Given the meta-paths in Table 5, the execution time is shown in Table 7. We can see that we can save around seven minutes (- 14.6%) with the ten meta-paths. In our experiment, 325 meta-paths and 1682 meta-paths are generated for 20NG and GCAT,<sup>5</sup> respectively. We can save around 3.8 and 19.6 h for the corresponding datasets. The reason is that MDPN shares the similar nature with PageRank-Nibble, which is that the running time is independent of the size of the graph. Similar result is found when comparing the space usage as shown in Table 7. By using MDPN, we can save up to 1.4G storage (15.2%) compared to storing the exact commuting matrices. In our real setting, we can save 45.5G and 235.5G storage for 20NG and GCAT datasets, respectively. Because MDPN only saves the nodes that have relatively high degree, which is important in sparse matrix. We thus can conclude that our MDPN based commuting matrix is effective for ensuring the KnowSim performance by considering the trade-off between effectiveness and efficiency.

## 7 Related work

In this section, we review our related work in following categories.

<sup>5</sup> We keep 20 and 43 entity types for 20NG and GCAT respectively, because they have relatively larger number of instances.

**Table 7** Analysis of execution time (min) and space usage (G) of using MDPN-based local graph, and the original graph for commuting matrix generation and storage in the sampled meta-path set

	MDPN	Original
Execution time (min)	41 (− 14.6%)	48
Space usage (G)	7.8 (− 15.2%)	9.2

## 7.1 Heterogeneous information networks

HIN is defined as a graph of multi-typed entities and relations (Han et al. 2010). Different from traditional graphs, HIN incorporates the type information which can be useful to identify the semantic meaning of the paths in the graph (Sun et al. 2011). Original HINs are developed for the applications of scientific publication network analysis (Sun et al. 2011, 2012). Then social network analysis also leverages this representation for user similarity and link prediction (Kong et al. 2013; Zhang et al. 2013, 2014). Seamlessly, we can see that the knowledge in world knowledge bases, e.g., Freebase, can be represented as an HIN, since the entities and relations in the knowledge base are all typed. Wang et al. (2015a, b, 2016a, b, 2017) introduce this representation to knowledge based analysis, and show that it can be very useful for our document categorization task. More recently, embedding based methods have been used to find a unified vector representation of nodes in HINs (Tang et al. 2015; Shang et al. 2016; Dong et al. 2017; Fu et al. 2017; Huang and Mamoulis 2017; Tu et al. 2018; Cui et al. 2017; Goyal and Ferrara 2018; Cai et al. 2018). Compared to embedding based methods, we use an explicit semantic approach to compute the similarities, which makes the similarities more explainable to humans.

## 7.2 Knowledge based text similarity

There have been existing studies using linguistic knowledge bases such as WordNet (Hotho et al. 2003) or general purpose knowledge bases such as Open Directory Project (ODP) (Gabrilovich and Markovitch 2005), Wikipedia (Gabrilovich and Markovitch 2007; Hu et al. 2008, 2009a, b; Song et al. 2015), or knowledge extracted from open domain data such as Probase (Song et al. 2011, 2015), to extend the features of documents to improve similarity measures. The similarity between documents is measured using one of several similarity measures in the vector space, such as cosine similarity, Jaccard correlation, dice similarity, Pearson's coefficient, and KL divergence. A lot of work (Huang 2008; Strehl et al. 2000) have experimented with them in different document datasets and found their performances are similar. More recently, there has been an increased emphasis on modeling objects using more complex data structures for the domains such as graphs or trees (Ganesan et al. 2003; Wan et al. 2005; Lakkaraju et al. 2008). However, they treat knowledge in such knowledge bases as "flat features" and do not consider the structural information contained in the links in knowledge bases.

There have been studies on evaluating word similarity based on WordNet considering the structural information (Budanitsky and Hirst 2006), and using word similarity to compute short text similarity (Do et al. 2009; Wan et al. 2005). For example, the distance from words to the root is used to capture the semantic relatedness between two words. However, WordNet is designed for single words. For named entities, a separate similarity should be designed (Do et al. 2009; Cohen et al. 2003). These studies do not consider the relationships between entities (e.g., “Obama” being related to “United States”). Thus, they may still lose structural information even if the knowledge base provides rich linked information. For example, nowadays there exist numerous general-purpose knowledge bases, e.g., Freebase (Bollacker et al. 2008), KnowItAll (Etzioni et al. 2004), TextRunner (Banko et al. 2007), WikiTaxonomy (Ponzetto et al. 2007), DBpedia (Auer et al. 2007), YAGO (Suchanek et al. 2007), NELL (Mitchell et al. 2015) and Knowledge Vault (Dong et al. 2014). They contain a lot of world knowledge about entity types and their relationships and provide us rich opportunities to develop a better measure to evaluate document similarities. Recently, there have been studies extending the above idea using knowledge bases (or knowledge graphs) (Paul et al. 2016; Traverso et al. 2016). However, they are still based on Lowest Common Ancestor (LCA), which could be computational costly when both the document number and the knowledge graph size are large.

### 7.3 Text categorization

Text classification is the task of predicting label of a given document. An in-depth review of the early studies in the area can be found in Sebastiani (2002). Besides, we refer to Aggarwal et al. (2012) for a bunch of recent works on utilizing additional information for the classification task. Some in-depth reviews of the early studies in text classification can be found in Sebastiani (2002) and Aggarwal et al. (2012). Several milestone studies include using support vector machine (SVM) (Joachims 1998) and Naive Bayes (McCallum et al. 1998) with BOW features for text classification. One direction of recent work is on leveraging structural information for better classification. Link based classification (Lu and Getoor 2003; Kong et al. 2012) use relationship between text (e.g., number of links) as additional features to original BOWs feature. Graph-of-words (Wang et al. 2005; Hassan et al. 2007; Rousseau et al. 2015) representation is recently proposed and show better results compared to BOW. However, these approaches focus on data statistics without considering the semantics of the link. For example, in graph-of-words, if two words occur near in one document, the words will be linked. Our method aims to leverage the semantics of links for classification, i.e., the entities and links are with types. Recently, more studies emerge that focus on using graph-of-words representation rather the BOW, thus improve the classification result (Hassan et al. 2007). Similar to our work, Rousseau et al. (2015) propose to transfer text categorization to graph classification. Both of the work represent the textual document with a graph of words instead of BOW. However, we want to do more, and represent the document with the specified world knowledge in the form of heterogeneous information networks. Besides graph-of-words, the structured network also contains knowledge base entities and their relations, as well as their type information.

Another direction is on enriching the text representation with semantics from world knowledge. Linguistic knowledge bases such as WordNet (Hotho et al. 2003) or general purpose knowledge bases such as Open Directory (Gabrilovich and Markovitch 2005), Wikipedia (Gabrilovich and Markovitch 2007; Hu et al. 2008, 2009b), or knowledge extracted from open domain data such as Web pages (Wang et al. 2013, 2015c) and Probase (Song et al. 2011, 2015), have been used to extend the features of documents to improve text categorization. Yet we do not use such knowledge as flat features, and instead encode link (meta-path) based similarities among documents in kernels, in the networks generated from knowledge base, Freebase.

Building semantic kernel using world knowledge for text categorization has been proposed in Siolas and Buc (2000), Wang et al. (2007) and Wang et al. (2008). The semantic kernel is constructed in a supervised way and only considers the direct (one-hop) links. However, we do not need an extra proximity matrix to construct the kernel. Besides, KnowSim kernel takes multi-hop links (i.e., meta-paths) via a totally unsupervised way. Besides KnowSim, knowledge-based graph semantic similarity (GSim) is proposed in Schuhmacher and Ponzetto (2014) to measure the document similarity based on DBpedia. However, the time complexity of computing GSim is high. So it is not feasible on our large-scale datasets (in original paper they experiment on a document set with 50 documents). KnowSim however can be computed in nearly linear time. Recently, Kim et al. (2015) introduce sentence kernel generated by word distances from a given word vector space based on word embedding. Yet our proposed KnowSim based kernel is built on the HIN constructed by explicit world knowledge from the knowledge base. It is also interesting to integrate the word embedding results and explicit world knowledge information (Song et al. 2015). In this way, the KnowSim can be more robust when facing the scarcity of knowledge for some specific domains.

## 8 Conclusion

Computing text similarity is a fundamental task with lots of applications such as text clustering, classification, and ranking. The existing text similarity measures focus on figuring out the useful “flat” features that could be critical for similarity computation and have found many such features that lead to the improvement of similarity performance. We consider to use the wealth of world knowledge to enable measuring the similarity with automatic domain-dependent feature generation and rich link information. In this paper, we use semantic parsing and semantic filtering modules to specify the world knowledge to domains, and then model the specified world knowledge in the form of heterogeneous information network, which enables the representation of the link information for the documents. By defining a novel document similarity measure, KnowSim (document similarity with world knowledge), the similarity between documents can be measured based on the meta-paths in the HIN constructed from the documents. We select to use Freebase as our source of world knowledge, which is collaboratively collected knowledge about entities and their organizations. Experiments on two benchmark datasets (20 newsgroups and RCV1) have demonstrated the power of our KnowSim against the state-of-the-art similarity measures. In the experiments, we further show both state-of-the-art performances are achieved by using KnowSim

in document clustering and classification tasks. Besides documents, we plan to generalize KnowSim to measure the similarity between entities with any same entity type, to improve the performance of more similarity based applications, such as entity similarity based classification and ranking.

**Acknowledgements** Chenguang Wang, Haoran Li, and Ming Zhang gratefully acknowledge the support by the National Natural Science Foundation of China (NSFC Grant Nos. 61772039, 91646202 and 61472006). Yangqiu Song was supported by China 973 Fundamental R&D Program (No. 2014CB340304) and the Early Career Scheme (ECS, No. 26206717) from Research Grants Council in Hong Kong. Jiawei Han was sponsored in part by U.S. Army Research Lab. under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), DARPA under Agreement No. W911NF-17-C-0099, National Science Foundation IIS 16-18481, IIS 17-04532, and IIS-17-41317, DTRA HDTRA11810026, and Grant 1U54GM114838 awarded by NIGMS through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative ([www.bd2k.nih.gov](http://www.bd2k.nih.gov)). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied by these agencies. We also thank the conference versions' and journal version's anonymous reviewers for their valuable comments and suggestions that help improve the quality of this manuscript.

## References

- Aggarwal CC, Zhai C (2012) A survey of text classification algorithms. In: Mining text data. Springer, pp 163–222
- Andersen R, Chung F, Lang K (2006) Local graph partitioning using pagerank vectors. In: Proceedings of the IEEE annual symposium on foundations of computer science (FOCS), pp 475–486
- Auer S, Bizer C, Kobilarov G, Lehmann J, Cyganiak R, Zachary I (2007) A nucleus for a web of open data. Springer, Dbpedia
- Banko M, Cafarella MJ, Soderland S, Broadhead M, Etzioni O (2007) Open information extraction from the web. In: Proceedings of the international joint conference on artificial intelligence (IJCAI), pp 2670–2676
- Basu S, Bilenko M, Mooney RJ (2004) A probabilistic framework for semi-supervised clustering. In: Proceedings of the ACM SIGKDD conference on knowledge discovery and data mining (KDD), pp 59–68
- Berant J, Chou A, Frostig R, Liang P (2013) Semantic parsing on Freebase from question-answer pairs. In: Proceedings of the conference on empirical methods in natural language processing (EMNLP), pp 1533–1544
- Berg C, Christensen JP, Ressel P (1984) Harmonic analysis on semigroups: theory of positive definite and related functions, volume 100 of graduate texts in mathematics, 1st edn. Springer, Berlin
- Blei DM, Ng AY, Jordan MI (2003) Latent Dirichlet allocation. *J Mach Learn Res (JMLR)* 3:993–1022
- Bollacker KD, Evans C, Paritosh P, Sturge T, Taylor J (2008) Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the ACM special interest group on management of data (SIGMOD), pp 1247–1250
- Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge University Press, Cambridge
- Budanitsky A, Hirst G (2006) Evaluating wordnet-based measures of lexical semantic relatedness. *Comput Linguist* 32(1):13–47
- Cai H, Zheng VW, Chang KC (2018) A comprehensive survey of graph embedding: problems, techniques and applications. *IEEE Trans Knowl Data Eng (TKDE)*
- Cohen WW, Ravikumar P, Fienberg SE (2003) A comparison of string distance metrics for name-matching tasks. In: Proceedings of the international joint conference on artificial intelligence (IJCAI) workshop on information integration, pp 73–78
- Cui P, Wang X, Pei J, Wenwu Z (2017) A survey on network embedding. *CoRR*. [arXiv:1711.08752](https://arxiv.org/abs/1711.08752)
- Do Q, Roth D, Sammons M, Tu Y, Vydiswaran VGV (2009) Robust, light-weight approaches to compute lexical similarity. In: Computer science research and technical reports. University of Illinois, pp 94–94

- Dong X, Gabrilovich E, Heitz G, Horn W, Lao N, Murphy K, Strohmman T, Sun S, Zhang W (2014) Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In: Proceedings of the ACM SIGKDD conference on knowledge discovery and data mining (KDD), pp 601–610
- Dong Y, Chawla NV, Swami A (2017) metapath2vec: scalable representation learning for heterogeneous networks. In: Proceedings of the ACM SIGKDD conference on knowledge discovery and data mining (KDD), pp 135–144
- Etzioni O, Cafarella M, Downey D (2004) Webscale information extraction in knowitall (preliminary results). In: Proceedings of the international world wide web conference (WWW), pp 100–110
- Fu T-Y, Lee W-C, Lei Z (2017) Hin2vec: explore meta-paths in heterogeneous information networks for representation learning. In: Proceedings of the ACM international conference on information and knowledge management (CIKM), pp 1797–1806
- Gabrilovich E, Markovitch S (2005) Feature generation for text categorization using world knowledge. In: Proceedings of the international joint conference on artificial intelligence (IJCAI), pp 1048–1053
- Gabrilovich E, Markovitch S (2007) Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In: Proceedings of the international joint conference on artificial intelligence (IJCAI), pp 1606–1611
- Ganesan P, Garcia-Molina H, Widom J (2003) Exploiting hierarchical domain structure to compute similarity. *ACM Trans Inf Syst (TOIS)* 21(1):64–93
- Goyal P, Ferrara E (2018) Graph embedding techniques, applications, and performance: a survey. *Knowl Based Syst* 151:78–94
- Han J, Sun Y, Yan X, Yu PS (2010) Mining knowledge from databases: an information network analysis approach. In: Proceedings of the ACM special interest group on management of data (SIGMOD), pp 1251–1252
- Hassan S, Mihalcea R, Banea C (2007) Random walk term weighting for improved text classification. In: Proceedings of the international conference on semantic computing (ICSC), pp 242–249
- He X, Cai D, Niyogi P (2006) Laplacian score for feature selection. In: Proceedings of the neural information processing systems (NIPS), pp 507–514
- Hotho A, Staab S, Stumme G (2003) Ontologies improve text document clustering. In: Proceedings of the IEEE international conference on data mining (ICDM), pp 541–544
- Hu J, Fang L, Cao Y, Zeng H-J, Li H, Yang Q, Chen Z (2008) Enhancing text clustering by leveraging Wikipedia semantics. In: Proceedings of the international ACM SIGIR conference on research and development in information retrieval (SIGIR), pp 179–186
- Hu X, Sun N, Zhang C, Chua T-S (2009a) Exploiting internal and external semantics for the clustering of short texts using world knowledge. In: Proceedings of the ACM international conference on information and knowledge management (CIKM), pp 919–928
- Hu X, Zhang X, Lu C, Park EK, Zhou X (2009b) Exploiting Wikipedia as external knowledge for document clustering. In: Proceedings of the ACM SIGKDD conference on knowledge discovery and data mining (KDD), pp 389–396
- Huang A (2008) Similarity measures for text document clustering. In: Proceedings of the New Zealand computer science research student conference (NZCSRSC), pp 49–56
- Huang Z, Mamouli N (2017) Heterogeneous information network embedding for meta path based proximity. CoRR. [arXiv:1701.05291](https://arxiv.org/abs/1701.05291)
- Joachims T (1998) Text categorization with support vector machines: learning with many relevant features. Springer, Berlin
- Kim J, Rousseau F, Vazirgiannis M (2015) Convolutional sentence kernel from word embeddings for short text categorization. In: Proceedings of the conference on empirical methods in natural language processing (EMNLP), pp 775–780
- Kong X, Yu PS, Ding Y, Wild DJ (2012) Meta path-based collective classification in heterogeneous information networks. In: Proceedings of the ACM international conference on information and knowledge management (CIKM), pp 1567–1571
- Kong X, Zhang J, Yu PS (2013) Inferring anchor links across multiple heterogeneous social networks. In: Proceedings of the ACM international conference on information and knowledge management (CIKM), pp 179–188
- Lakkaraju P, Gauch S, Speretta M (2008) Document similarity based on concept tree distance. In: Proceedings of the nineteenth ACM conference on hypertext and hypermedia, pp 127–132
- Lang K (1995) Newsweeder: learning to filter netnews. In: Proceedings of the international conference on machine learning (ICML), pp 331–339



- Lao N, Cohen WW (2010) Relational retrieval using a combination of path-constrained random walks. *Mach Learn* 81(1):53–67
- Lao N, Mitchell T, Cohen WW (2011) Random walk inference and learning in a large scale knowledge base. In: *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, pp 529–539
- Lewis DD, Yang Y, Rose TG, Li F (2004) RCV1: a new benchmark collection for text categorization research. *J Mach Learn Res (JMLR)* 5:361–397
- Li Y, Wang C, Han F, Han J, Roth D, Yan X (2013) Mining evidences for named entity disambiguation. In: *Proceedings of the ACM SIGKDD conference on knowledge discovery and data mining (KDD)*, pp 1070–1078
- Lu Q, Getoor L (2003) Link-based classification. In: *Proceedings of the international conference on machine learning (ICML)*, pp 496–503
- Luss R, d'Aspremont A (2008) Support vector machine classification with indefinite kernels. In: *Proceedings of the neural information processing systems (NIPS)*, pp 953–960
- McCallum A, Nigam K, et al (1998) A comparison of event models for naive bayes text classification. In: *Proceedings of the association for the advancement of artificial intelligence (AAAI) workshop*, vol 752, pp 41–48
- Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: *Proceedings of the neural information processing systems (NIPS)*, pp 3111–3119
- Mitchell TM, Cohen WW, Hruschka ER Jr, Talukdar PP, Betteridge J, Carlson A, Mishra BD, Gardner M, Kisiel B, Krishnamurthy J, Lao N, Mazaitis K, Mohamed T, Nakashole N, Platanios EA, Ritter A, Samadi M, Settles B, Wang RC, Wijaya DT, Gupta A, Chen X, Saparov A, Greaves M, Welling J (2015) Never-ending learning. In: *Proceedings of the association for the advancement of artificial intelligence (AAAI)*, pp 2302–2310
- Mooney RJ (2007) Learning for semantic parsing. In: *Proceedings of the international conference on computational linguistics and intelligent text processing (CICLing)*, pp 311–324
- Nesterov Y (2005) Smooth minimization of non-smooth functions. *Math Program* 103(1):127–152
- Paul C, Rettinger A, Mogadala A, Knoblock CA, Pedro S (2016) Efficient graph-based document similarity. In: *International conference on the semantic web. latest advances and new domains*, pp 334–349
- Ponzetto SP, Strube M (2007) Deriving a large-scale taxonomy from Wikipedia. In: *Proceedings of the association for the advancement of artificial intelligence (AAAI)*, pp 1440–1445
- Ratinov L, Roth D (2009) Design challenges and misconceptions in named entity recognition. In: *Proceedings of the SIGNLL conference on computational natural language learning (CoNLL)*, pp 147–155
- Rousseau F, Kiagias E, Vazirgiannis M (2015) Text categorization as a graph classification problem. In: *Annual meeting of the association for computational linguistics (ACL)*, pp 1702–1712
- Sahami M (1998) Using machine learning to improve information access. PhD thesis, Stanford University
- Schuhmacher M, Ponzetto SP (2014) Knowledge-based graph document modeling. In: *Proceedings of the ACM international conference on web search and data mining (WSDM)*, pp 543–552
- Sebastiani F (2002) Machine learning in automated text categorization. *ACM Comput Surv (CSUR)* 34(1):1–47
- Shang J, Qu M, Liu J, Kaplan LM, Han J, Peng J (2016) Meta-path guided embedding for similarity search in large-scale heterogeneous information networks. *CoRR*. [arXiv:1610.09769](https://arxiv.org/abs/1610.09769)
- Siolas G, d'Alché-Buc F (2000) Support vector machines based on a semantic kernel for text categorization. In: *Proceedings of international joint conference on neural networks (IJCNN)*, pp 205–209
- Song Y, Roth D (2015) Unsupervised sparse vector densification for short text similarity. In: *Proceedings of the annual conference of the North American chapter of the association for computational linguistics: human language technologies (NAACL-HLT)*, pp 1275–1280
- Song Y, Pan S, Liu S, Zhou MX, Qian W (2009) Topic and keyword re-ranking for LDA-based topic modeling. In: *Proceedings of the ACM conference on information and knowledge management (CIKM)*, pp 1757–1760
- Song Y, Wang H, Wang Z, Li H, Chen W (2011) Short text conceptualization using a probabilistic knowledgebase. In: *Proceedings of the international joint conference on artificial intelligence (IJCAI)*, pp 2330–2336
- Song Y, Wang S, Wang H (2015) Open domain short text conceptualization: a generative + descriptive modeling approach. In: *Proceedings of the international joint conference on artificial intelligence (IJCAI)*, pp 3820–3826


- Strehl A, Ghosh J (2003) Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *J Mach Learn Res (JMLR)* 3:583–617
- Strehl A, Ghosh J, Mooney R (2000) Impact of similarity measures on web-page clustering. In: Proceedings of the association for the advancement of artificial intelligence (AAAI), pp 58–64
- Suchanek FM, Kasneci G, Weikum G (2007) Yago: a core of semantic knowledge. In: Proceedings of the international world wide web conference (WWW), pp 697–706
- Sun Y, Han J, Yan X, Yu PS, Tianyi W (2011) Pathsim: meta path-based top-k similarity search in heterogeneous information networks. *Proc VLDB Endow (PVLDB)* 4(11):992–1003
- Sun Y, Norrick B, Han J, Yan X, Yu PS, Yu X (2012) Integrating meta-path selection with user-guided object clustering in heterogeneous information networks. In: Proceedings of the ACM SIGKDD conference on knowledge discovery and data mining (KDD), pp 1348–1356
- Sun Y, Norrick B, Han J, Yan X, Yu PS, Yu X (2013) Pathselclus: integrating meta-path selection with user-guided object clustering in heterogeneous information networks. *ACM Trans Knowl Discov Data (TKDD)* 7(3):11:1–11:23
- Tang J, Qu M, Mei Q (2015) PTE: predictive text embedding through large-scale heterogeneous text networks. In: Proceedings of the ACM SIGKDD conference on knowledge discovery and data mining (KDD), pp 1165–1174
- Traverso I, Vidal M-E, Kämpgen B, Sure-Vetter Y (2016) GADES: a graph-based semantic similarity measure. In: International conference on semantic systems (SEMANTICS), pp 101–104
- Tu K, Cui P, Wang X, Wang F, Zhu W (2018) Structural deep embedding for hyper-networks. In: Proceedings of the association for the advancement of artificial intelligence (AAAI)
- Wan X, Peng Y (2005) The earth mover’s distance as a semantic measure for document similarity. In: Proceedings of the ACM international conference on information and knowledge management (CIKM), pp 301–302
- Wang C, Duan N, Zhou M, Zhang M (2013) Paraphrasing adaptation for web search ranking. In: Annual meeting of the association for computational linguistics (ACL), pp 41–46
- Wang C, Song Y, El-Kishky A, Roth D, Zhang M, Han J (2015a) Incorporating world knowledge to document clustering via heterogeneous information networks. In: Proceedings of the ACM SIGKDD conference on knowledge discovery and data mining (KDD), pp 1215–1224
- Wang C, Song Y, Li H, Zhang M, Han J (2015b) Knowsim: a document similarity measure on structured heterogeneous information networks. In: Proceedings of the IEEE international conference on data mining (ICDM), pp 506–513
- Wang C, Song Y, Roth D, Wang C, Han J, Ji H, Zhang M (2015c) Constrained information-theoretic tripartite graph clustering to identify semantically similar relations. In: Proceedings of the international joint conference on artificial intelligence (IJCAI), pp 3882–3889
- Wang C, Song Y, Li H, Zhang M, Han J (2016a) Text classification with heterogeneous information network kernels. In: Proceedings of the association for the advancement of artificial intelligence (AAAI), pp 2130–2136
- Wang C, Song Y, Roth D, Zhang M, Han J (2016) World knowledge as indirect supervision for document clustering. *ACM Trans Knowl Discov Data (TKDD)* 11(2):13:1–13:36
- Wang C, Song Y, Li H, Sun Y, Zhang M, Han J (2017) Distant meta-path similarities for text-based heterogeneous information networks. In: Proceedings of the ACM international conference on information and knowledge management (CIKM), pp 1629–1638
- Wang P, Domeniconi C (2008) Building semantic kernels for text classification using Wikipedia. In: Proceedings of the ACM SIGKDD conference on knowledge discovery and data mining (KDD), pp 713–721
- Wang P, Hu J, Zeng H-J, Chen L, Chen Z (2007) Improving text classification by using encyclopedia knowledge. In: Proceedings of the IEEE international conference on data mining (ICDM), pp 332–341
- Wang W, Do DB, Lin X (2005) Term graph model for text classification. In: Proceedings of the international conference on advanced data mining and applications (ADMA), pp 19–30
- Ying Y, Campbell C, Girolami M (2009) Analysis of SVM with indefinite kernels. In: Proceedings of the neural information processing systems (NIPS), pp 2205–2213
- Zelnik-Manor L, Perona P (2005) Self-tuning spectral clustering. In: Saul LK, Weiss Y, Bottou L (eds) Proceedings of the neural information processing systems (NIPS), pp 1601–1608
- Zhang J, Kong X, Yu PS (2013) Predicting social links for new users across aligned heterogeneous social networks. In: Proceedings of the IEEE international conference on data mining (ICDM), pp 1289–1294



Zhang J, Kong X, Yu PS (2014) Transferring heterogeneous links across location-based social networks. In: Proceedings of the ACM international conference on web search and data mining (WSDM), pp 303–312

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Affiliations

**Chenguang Wang<sup>1</sup> · Yangqiu Song<sup>2</sup>  · Haoran Li<sup>3</sup> · Ming Zhang<sup>3</sup> · Jiawei Han<sup>4</sup>**

✉ Yangqiu Song  
yqsong@cse.ust.hk

Chenguang Wang  
chgwang@amazon.com

Haoran Li  
lihaoran\_2012@pku.edu.cn

Ming Zhang  
mzhang\_cs@pku.edu.cn

Jiawei Han  
hanj@illinois.edu

<sup>1</sup> Amazon AI, 2100 University Ave, East Palo Alto, CA, USA

<sup>2</sup> Department of CSE, HKUST, Clear Water Bay, Hong Kong

<sup>3</sup> School of EECS, Peking University, Beijing, China

<sup>4</sup> Department of CS, UIUC, Urbana, IL 61801, USA