KnowSim: A Document Similarity Measure on Structured Heterogeneous Information Networks

Chenguang Wang† , Yangqiu Song‡ , Haoran Li† , Ming Zhang† , Jiawei Han‡
†School of EECS, Peking University
‡Department of Computer Science, University of Illinois at Urbana-Champaign
{wangchenguang, lihaoran_2012, mzhang_cs}@pku.edu.cn, {yqsong, hanj}@illinois.edu

***Abstract*—As a fundamental task, document similarity measure has broad impact to document-based classification, clustering and ranking. Traditional approaches represent documents as bag-of-words and compute document similarities using measures like cosine, Jaccard, and dice. However, entity phrases rather than single words in documents can be critical for evaluating document relatedness. Moreover, types of entities and links between entities/words are also informative. We propose a method to represent a document as a typed heterogeneous information network (HIN), where the entities and relations are annotated with types. Multiple documents can be linked by the words and entities in the HIN. Consequently, we convert the document similarity problem to a graph distance problem. Intuitively, there could be multiple paths between a pair of documents. We propose to use the meta-path defined in HIN to compute distance between documents. Instead of burdening user to define meaningful meta-paths, an automatic method is proposed to rank the meta-paths. Given the meta-paths associated with ranking scores, an HIN-based similarity measure, KnowSim, is proposed to compute document similarities. Using 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 generates impressive high-quality document clustering.**

I. Introduction

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, as Fig. 1 shown in [1], the similarity between the two documents is 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 "Bush" 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.

Some existing studies use linguistic knowledge bases such as WordNet [2] or general purpose knowledge bases such as Open Directory Project (ODP) [3], Wikipedia [4], [5], [6], [7], [8], [9], or knowledge extracted from open domain data such as Probase [10], [11], to extend the features of documents to improve similarity measures. 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 or string similarity based on WordNet or other knowledge [12] considering the structural information [13], and using word similarity to compute short text similarity [14], [15]. 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 [14], [16]. 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 [17], KnowItAll [18], TextRunner [19], WikiTaxonomy [20], DBpedia [21], YAGO [22], NELL [23] and Knowledge Vault [24]. 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.

In this paper, we propose KnowSim, a heterogeneous information network (HIN) [25] based similarity measure that explores the structural information from knowledge bases to compute document similarities. We use Freebase as the source of world knowledge. Freebase is a collaboratively collected knowledge base about entities and their organizations [17]. We follow [1] 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. Since the knowledge bases provide entity types, the resulting data naturally form an HIN. The named entities and their types, as well as the documents and the words form the HIN.

Given a constructed HIN, we use meta-path based similarity [26] 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 compute the best ones. Based on the PageRank-Nibble algorithm [27] that can conduct efficient graph pruning locally 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 [28] and Laplacian score [29] 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.

II. Construction of Document HIN

In this section, we introduce how to generate heterogeneous information network (HIN) for the documents based on world knowledge bases. Please find the basic concepts related to HIN, such as network schema, meta-path, and commuting matrix in [30]. We use the unsupervised semantic parser and conceptualization based semantic filter proposed in [1] to generate the semantic meaning of each document. The output 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 [1], the network contains multiple entity types: *document* $\mathcal{D}$, *word* $\mathcal{W}$, *named entities* $\{\mathcal{E}^{\mathrm{I}}\}_{I=1}^{T}$, and *relation types* connecting the *entity types*. Different from [1] 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. However, in Freebase, there are about $1,500+$ entity types and $3,500+$ relation types, which will generate an exponential number of meta-paths. In previous work [26], [31], 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. 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 (Section III), and (ii) use unsupervised feature selection approaches to select semantically meaningful meta-paths for final document similarity computation (Section IV).

III. 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 [26]. 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 ([32], [33]), 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 [27] 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 [27]. 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* (as outlined in Algorithm 1).

Input : A graph $\mathcal{G}$, a meta-path $\mathcal{P}$, a node set $\hat{\mathcal{V}}$, and two parameters: α and ϵ.

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 [27]. 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 $j \in [1, |Supp(\mathbf{p})|]$, 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*$(\mathcal{G}, \mathcal{P}, \hat{\mathcal{V}}, \alpha, \epsilon)$.

Based on the proof in [27], for any graph, a good approximation can be guaranteed, thus satisfy the efficiency bound, which holds independent of the size of the network. 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 [26] $\mathbf{M}_{\mathcal{P}}$ 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 $\{\mathcal{E}^{\mathrm{I}}\}_{I=1}^{T}$ 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.

IV. HIN-Based Document Similarity

In this section, we introduce HIN-based document 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 the document HIN extracted from the world knowledge base, meta-paths can be used to compute the similarity between documents. PathSim [26] is proposed to define the similarity along a meta-path. However, previous approaches require human to define the mata-path(s). Here we should have multiple meta-paths useful for finding similar documents. Therefore, it is necessary to provide an automated mechanism to select the most meaningful meta-paths to define similarity between documents.

A. Meta-Path Selection

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

1) Document-Meta-Path Representation: For each meta-path $\mathcal{P}_j$, we have a commuting matrix $\mathbf{M}_{\mathcal{P}_j}$. Suppose we have N documents 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 documents or between meta-paths. Here we propose to use a simplest way based on document-meta-path co-occurrence representation. We generate a document 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 document i. If the meta-path j is dense for document i in the HIN, then most pairs related to document 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 documents to evaluate the meta-path similarity. Specifically, we can define $sim(\mathbf{D}_{\cdot,j_1}, \mathbf{D}_{\cdot,j_2})$ 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 document similarity based on all the meta-path densities for the documents. Specifically, we can define $sim(\mathbf{D}_{i_1,\cdot}, \mathbf{D}_{i_2,\cdot})$ where $\mathbf{D}_{i_1,\cdot}$ is the i_1-th row of $\mathbf{D}$. Note that we do not use this document similarity as our final similarity between two documents because it is only based on meta-path density. What we need is more elaborate document similarity based on each document meta-path pair. We will introduce the meta-path specific semantically meaningful similarity in the next subsection.

Given the similarities defined above, we introduce two feature selection methods based on them to select the most meaningful meta-paths.

2) Maximal Spanning Tree based Selection: Inspired by the mutual information-based feature selection [28], [34], 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 [28]. Intuitively, if two meta-paths have similar density distributions over all the documents, 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.

3) Laplacian Score based Selection: We also use the Laplacian score to select meta-paths [29], [34]. 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 documents from different clusters.

B. KnowSim: Knowledge-Driven Document Similarity

Given the selected meta-paths, we now define our knowledge-driven document similarity measure, *KnowSim*. Intuitively, if two documents are more strongly connected by the important (i.e., highly weighted) meta-paths, they tend to be more similar. Formally, we have

Definition 1: ***KnowSim: a knowledge-driven document similarity measure.*** *Given a collection of symmetric meta-paths, denoted as $\mathbf{P} = \{\mathcal{P}_m\}_{m=1}^{M'}$, KnowSim between two documents d_i and d_j is defined as:*

$$KS(d_i, d_j) = \frac{2 \times \sum_m^{M'} \omega_m |\{p_{i \rightsquigarrow j} \in \mathcal{P}_m\}|}{\sum_m^{M'} \omega_m |\{p_{i \rightsquigarrow i} \in \mathcal{P}_m\}| + \sum_m^{M'} \omega_m |\{p_{j \rightsquigarrow j} \in \mathcal{P}_m\}|}. \quad (1)$$

where $p_{i \rightsquigarrow j} \in \mathcal{P}_m$ 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 $\boldsymbol{\omega} = [\omega_1, \cdots, \omega_m, \cdots, \omega_{M'}]$ to denote the meta-path weights, where ω_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 documents 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 documents are, which is further normalized by the semantic broadness of d_i and d_j.

Note that KnowSim can be generalized to measure the similarity between any two entities rather than documents, which will be very helpful to determine the entity similarity, because we take more link information into consideration rather than a single meta-path. If KnowSim only contains a single meta-path, it degenerates to PathSim.

V. Experiments

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

A. Datasets

We use the following two benchmark datasets to evaluate the document similarity task.

20Newsgroups (20NG): The 20newsgroups dataset [35] contains about 20,000 newsgroups documents evenly distributed across 20 newsgroups.[1]

RCV1: The RCV1 dataset is a dataset containing manually labeled newswire stories from Reuter Ltd [36]. 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.

B. World Knowledge Base

Then we introduce the knowledge base we use. In [1], 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[2] 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 1,500+ entity types and 3,500+ 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 I. 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 $1,682$ symmetric meta-paths are generated based on the MDPN algorithm (Section III), for 20NG and GCAT respectively. We can save around 3.8 hours and 19.6 hours 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.

TABLE I: Statistics of entities in different datasets with semantic parsing and filtering using Freebase: #(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).

	#(Document)	#(Word)	#(FBEntity)	#(Total)	#(Types)
20NG	19,997	60,691	28,034	108,722	2,615
GCAT	60,608	95,001	110,344	265,953	2,665

C. 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 Section IV-A. Following [1], we use the specified world knowledge as features to enhance cosine, Jaccard and dice. The feature settings are defined as follows.

- BOW: Traditional bag-of-words model with the tf weighting mechanism.
- BOW+TOPIC: BOW integrated with additional features from topics generated by LDA [37]. 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 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 II, 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

[1] http://qwone.com/~jason/20Newsgroups/

[2] https://developers.google.com/freebase/

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 I 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 suggests that if there exists world knowledge bases with better precision and coverage, we could get better performance.

D. Application: Spectral Clustering Using KnowSim Matrix

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 [38] 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) [39] 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.

As shown in Table III, 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 II). 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.

VI. Conclusion

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 to represent 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 automatically generated meta-paths in the HIN constructed from the documents.

Acknowledgments

Chenguang Wang gratefully acknowledges the support by the National Natural Science Foundation of China (NSFC Grant Number 61472006) and the National Basic Research Program (973 Program No. 2014CB340405). The research is also partially supported by the Army Research Laboratory (ARL) under agreement W911NF-09-2-0053, and by DARPA under agreement number FA8750-13-2-0008. Research is also partially sponsored by National Science Foundation IIS-1017362, IIS-1320617, and IIS-1354329, HDTRA1-10-1-0120, and grant 1U54GM114838 awarded by NIGMS through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative (www.bd2k.nih.gov), and MIAS, a DHS-IDS Center for Multimodal Information Access and Synthesis at UIUC. 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 or the U.S. Government.

References

[1] C. Wang, Y. Song, A. El-Kishky, D. Roth, M. Zhang, and J. Han, "Incorporating world knowledge to document clustering via heterogeneous information networks," in *KDD*, 2015, pp. 1215–1224.

[2] A. Hotho, S. Staab, and G. Stumme, "Ontologies improve text document clustering," in *ICDM*, 2003, pp. 541–544.

[3] E. Gabrilovich and S. Markovitch, "Feature generation for text categorization using world knowledge," in *IJCAI*, 2005, pp. 1048–1053.

[4] ——, "Computing semantic relatedness using wikipedia-based explicit semantic analysis," in *IJCAI*, 2007, pp. 1606–1611.

[5] J. Hu, L. Fang, Y. Cao, H.-J. Zeng, H. Li, Q. Yang, and Z. Chen, "Enhancing text clustering by leveraging Wikipedia semantics," in *SIGIR*, 2008, pp. 179–186.

[6] X. Hu, X. Zhang, C. Lu, E. K. Park, and X. Zhou, "Exploiting wikipedia as external knowledge for document clustering," in *KDD*, 2009, pp. 389–396.

[7] X. Hu, N. Sun, C. Zhang, and T.-S. Chua, "Exploiting internal and external semantics for the clustering of short texts using world knowledge," in *CIKM*, 2009, pp. 919–928.

[8] Y. Song and D. Roth, "On dataless hierarchical text classification," in *AAAI*, 2014, pp. 1579–1585.

[9] ——, "Unsupervised sparse vector densification for short text similarity," in *NAACL*, 2015.

[10] Y. Song, H. Wang, Z. Wang, H. Li, and W. Chen, "Short text conceptualization using a probabilistic knowledgebase," in *IJCAI*, 2011, pp. 2330–2336.

[11] Y. Song, S. Wang, and H. Wang, "Open domain short text conceptualization: A generative + descriptive modeling approach," in *IJCAI*, 2015.

[12] C. Wang, N. Duan, M. Zhou, and M. Zhang, "Paraphrasing adaptation for web search ranking." in *ACL*, 2013, pp. 41–46.

[13] A. Budanitsky and G. Hirst, "Evaluating wordnet-based measures of lexical semantic relatedness," *Computational Linguistics*, vol. 32, no. 1, pp. 13–47, 2006.

[14] Q. Do, D. Roth, M. Sammons, Y. Tu, and V. Vydiswaran, "Robust, light-weight approaches to compute lexical similarity," 2009.

TABLE II: Correlation coefficient of different similarity measures on 20NG and GCAT. "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.

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

TABLE III: NMI of clustering on 20NG and GCAT using the similarity matrix generated by different similarity measures. "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.

Dataset	Similarity Matrix Source	BOW	BOW+TOPIC	BOW+ENTITY	BOW+TOPIC+ENTITY
20NG	Cosine	0.3440	0.3461	0.3896	0.4247
	Jaccard	0.3547	0.3517	0.3850	0.4292
	Dice	0.3440	0.3457	0.3894	0.4248
KnowSim+UNI	0.4304	KnowSim+MST	0.4412	KnowSim+LAP	0.4461 (+**3.9**%)
GCAT	Cosine	0.3932	0.4352	0.2394	0.4106
	Jaccard	0.3887	0.4292	0.2497	0.4159
	Dice	0.3932	0.4355	0.2392	0.4112
KnowSim+UNI	0.4463	KnowSim+MST	0.4653	KnowSim+LAP	0.4736 (+**8.8**%)

[15] X. Wan and Y. Peng, "The earth mover's distance as a semantic measure for document similarity," in *CIKM*, 2005, pp. 301–302.

[16] W. W. Cohen, P. Ravikumar, and S. E. Fienberg, "A comparison of string distance metrics for name-matching tasks." in *IJCAI Workshop on Information Integration*, 2003, pp. 73–78.

[17] K. D. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *SIGMOD*, 2008, pp. 1247–1250.

[18] O. Etzioni, M. Cafarella, and D. Downey, "Webscale information extraction in knowitall (preliminary results)," in *WWW*, 2004, pp. 100–110.

[19] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, "Open information extraction from the web," in *IJCAI*, 2007, pp. 2670–2676.

[20] S. P. Ponzetto and M. Strube, "Deriving a large-scale taxonomy from wikipedia," in *AAAI*, 2007, pp. 1440–1445.

[21] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, *Dbpedia: A nucleus for a web of open data*. Springer, 2007.

[22] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: a core of semantic knowledge," in *WWW*, 2007, pp. 697–706.

[23] T. M. Mitchell, W. W. Cohen, E. R. H. Jr., P. P. Talukdar, J. Betteridge, A. Carlson, B. D. Mishra, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. A. Platanios, A. Ritter, M. Samadi, B. Settles, R. C. Wang, D. T. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling, "Never-ending learning," in *AAAI*, 2015, pp. 2302–2310.

[24] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang, "Knowledge vault: A web-scale approach to probabilistic knowledge fusion," in *KDD*, 2014, pp. 601–610.

[25] J. Han, Y. Sun, X. Yan, and P. S. Yu, "Mining knowledge from databases: An information network analysis approach," in *SIGMOD*, 2010, pp. 1251–1252.

[26] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks," *PVLDB*, pp. 992–1003, 2011.

[27] R. Andersen, F. Chung, and K. Lang, "Local graph partitioning using pagerank vectors," in *FOCS*, 2006, pp. 475–486.

[28] M. Sahami, "Using machine learning to improve information access," Ph.D. dissertation, stanford university, 1998.

[29] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *NIPS*, 2006, pp. 507–514.

[30] Y. Sun and J. Han, "Mining heterogeneous information networks: principles and methodologies," *Synthesis Lectures on Data Mining and Knowledge Discovery*, vol. 3, no. 2, pp. 1–159, 2012.

[31] Y. Sun, B. Norick, J. Han, X. Yan, P. S. Yu, and X. Yu, "Integrating meta-path selection with user-guided object clustering in heterogeneous information networks," in *KDD*, 2012, pp. 1348–1356.

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

[33] N. Lao, T. Mitchell, and W. W. Cohen, "Random walk inference and learning in a large scale knowledge base," in *EMNLP*, 2011, pp. 529–539.

[34] Y. Song, S. Pan, S. Liu, M. X. Zhou, and W. Qian, "Topic and keyword re-ranking for lda-based topic modeling," in *CIKM*, 2009, pp. 1757–1760.

[35] K. Lang, "Newsweeder: Learning to filter netnews," in *ICML*, 1995, pp. 331–339.

[36] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, "RCV1: A new benchmark collection for text categorization research," *JMLR*, vol. 5, pp. 361–397, 2004.

[37] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *JMLR*, vol. 3, pp. 993–1022, 2003.

[38] L. Zelnik-manor and P. Perona, "Self-tuning spectral clustering," in *NIPS*, L. Saul, Y. Weiss, and L. Bottou, Eds., 2005, pp. 1601–1608.

[39] A. Strehl and J. Ghosh, "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," *JMLR*, vol. 3, pp. 583–617, 2003.