IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 3, 2019, accepted March 25, 2019, date of current version May 6, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2909104

A Lexical Resource-Constrained Topic Model for Word Relatedness

YONGJING YIN, JIALI ZENG, HONGJI WANG, KEQING WU, BIN LUO, AND JINSONG SU
Software School, Xiamen University, Xiamen 361005, China

Corresponding author: Jinsong Su (jssu@xmu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61672440, and in part by the Beijing Advanced Innovation Center for Language Resources, Natural Science Foundation of Fujian Province of China, under Grant 2016J05161.

ABSTRACT Word relatedness computation is an important supporting technology for many tasks in natural language processing. Traditionally, there have been two distinct strategies for word relatedness measurement: one utilizes corpus-based models, whereas the other leverages external lexical resources. However, either solution has its strengths and weaknesses. In this paper, we propose a lexical resource-constrained topic model to integrate the two complementary strategies effectively. Our model is an extension of probabilistic latent semantic analysis, which automatically learns word-level distributed representations forward relatedness measurement. Furthermore, we introduce generalized expectation maximization (GEM) algorithm for statistical estimation. The proposed model not merely inherit the advantage of conventional topic models in dimension reduction, but it also refines parameter estimation by using word pairs that are known to be related. The experimental results in different languages demonstrate the effectiveness of our model in topic extraction and word relatedness measurement.

INDEX TERMS Natural language processing, unsupervised learning.

I. INTRODUCTION

Semantic relatedness computation between two words is of great importance in the field of Natural Language Processing (NLP) [1]. It has been widely used in many NLP tasks, such as word sense disambiguation, discourse analysis and so on. Therefore, how to accurately compute word relatedness is always a hot topic in the community of NLP.

Recently, many methods have been proposed to identify semantic similarity of word pairs and can be mainly classified into the following two categories: (1) Corpus based approaches. In this respect, the related approaches based on large-scale corpus can be further divided into neural and non-neural processes. For instance, statistical approaches like Vector Space Model(VSM) [2], Probabilistic Latent Semantic Analysis (PLSA) and Latent Dirichlet allocation (LDA) are typical non-neural processes, while word embeddings such as Word2vec [3] are neural solutions. However, results of these approaches rely too much on the quantity and quality of training corpus while lack of the external influence of lexical resources which generally possess high quality. (2) Lexical resource-based approaches. Unlike the former category of approaches, these ones leverage external lexical resources such as WordNet and Wikipedia to calculate word relatedness [4]–[16]. Yet one obvious drawback of these approaches is that the construction of lexical resources requires a large quantity of manual annotation, which is a heavy burden. And consequently, such resources cover only a small fraction of language lexicons. As mentioned earlier, both strategies have advantages and disadvantages, and since they are complementary to each other, the combination of them will further improves the measurement of word relatedness. This assumption is demonstrated in many experiments [17]–[24] that have yielded a great number of positive results by combining them together.

Following the aforementioned ideas, in this paper we propose a topic model with lexical resource constraints for word relatedness computation with the aim of absorbing advantages of both two categories. As is known, topic models can be used to learn word-level distributed representations in a dimension reduction way. Specifically, the neighboring context words are collected to form a pseudo document representing the meaning of the center word, and then, topic models are employed to automatically mine the topic-document

The associate editor coordinating the review of this manuscript and approving it for publication was Jihad Aljaam.

posterior distribution in latent topic space. Compared with conventional approaches such as vector space model, topic models generate distributed representations with the ability to alleviate data sparsity to some extent, and thus have been widely used in word relatedness measurement and related NLP tasks. For instance, Latent Semantic Analysis [25]–[27] is the first model applied to word relatedness computation. Also based on topic models, [28] compute word semantic relatedness for word sense disambiguation. And in a related application of text segmentation, [29] adopt a topic model to measure semantic similarity with a Fisher kernel. However, in conventional topic models, such as PLSA and LDA, only document-level word co-occurrence has been taken into account, neglecting interactions between documents that deserve consideration. Therefore, the potential of topic models for word relatedness is far from being exploited. To compensate the above-mentioned defect of topic models, in our work, we extend the conventional PLSA model with external lexical resource constraints. Besides the capability of dimensionality reduction in word meaning representation, our model also utilizes external lexical resource to refine model training. This is achieved by incorporating word pairs that are known to be related as constraints in the learning process. To explain in detail, for two related words, posterior distributions of their pseudo documents are no longer assumed to be isolated, but rather encouraged to be close to each other in semantic space.

We mainly make three contributions. Firstly, we propose a novel topic model for word relatedness measurement, which incorporates external lexical resource constraints into PLSA. Our model has benefits of both conventional statistics-based approaches and lexical resource-based approaches. Secondly, we introduce Generalized Expectation Maximization(GEM) algorithm for statistical estimation of the model. Thirdly, experimental results on topic extraction and word relatedness tasks demonstrate the effectiveness of our model.

The rest of this paper is structured as follows: We begin in Section 2 with the summary of related work followed by the approach overview in Section 3. Then we elaborate our model in Section 4. Section 5 presents details about how to estimate parameters of our model. Experimental results are reported in Section 6 and in Section 7 we conclude our work.

II. RELATED WORK

The measurement of word relatedness has always been a subject in need of intensive investigation for years. The related work mainly includes the following two categories:

(1) Corpus based approaches. Since [30] first proposed an information-based approach to define and quantify lexical similarity, quantities of works largely based on corpus have emerged, which can be further divided into neural and non-neural processes. Statistical approaches are typical non-neural processes, in which they either use statical metrics such as *Point Mutual Information* to measure word relatedness, or formulate the notion of distributional similarity as [31] did, deriving the distributional vectors of words from a large corpus to measure the relatedness of words based on the similarity of their context. Besides the work of [32], Reisinger and Mooney's [33] also belongs to statistical approaches. Afterwards, many researchers apply topic models like LSA [34] and LDA [35] to compute word semantic relatedness by embedding words into latent semantic space. The other kind of approaches adopts neural models to access word relatedness by building distributed word representations(a.k.a. word embeddings) [36]. Different from conventional one-hot representations and distributional word representations based on co-occurrence matrix, distributed word representations are low-dimensional dense vectors trained with neural networks by maximizing the likelihood of a text corpus. Based on the theory, a series of works employ deep learning techniques to learn high-quality word representations [37], [38], among which Word2vec [3] with skip-gram generates considerable excitement. Nevertheless, all these approaches greatly depend on the training corpus, whose quantity and quality is hard to guarantee, while neglecting the access to external lexical resources, which are generally of high quality.

(2) Lexical resource-based approaches. With the appearance of lexical databases such as WordNet and Rogets' Thesaurus, many researchers tried to implement these external lexical resources into word relatedness computation [4]–[16]. Most of these approaches leverage lexical resources that encode relations between words including *synonymy*, *hypernymy*, and *meronymy*. However, the construction of lexical resources not only requires expertise in lexicography, but also costs a lot of time and effort. Consequently, such resources often provide limited coverage, which could aggravate the ineffectiveness of relatedness computation. To solve this problem, recently, some approaches based on large amounts of human knowledge like Wikipedia have been proposed, such as Explicit Semantic Analysis [39] and Temporal Semantic Analysis [40]. Besides, [41] devise a method that uses lexical resources, as opposed to collaboratively constructed resources such as Wikipedia, to measure relatedness.

With the purpose of absorbing the advantages of the two above-mentioned directions, many researchers have tried to introduce external lexical resources into corpus based models and have obtained good results. In this respect, the related work often applies supervised learning techniques to assemble relatedness scores obtained by different methods [17]–[22]. In accord with our idea, [23] and [24] effectively combine the above two directions using neural networks. Despite the same motivation, our model, distinctive from neural ones, is a non-neural process. To be specific, our model is an extension of topic model which also suits word relatedness computation well.

III. THE APPROACH OVERVIEW

When using our method, we implement word relatedness computation in the following three steps.

In **Step 1**, we first use context information to represent word meaning, as implemented in the previous works [42],

[43]. Specifically, for each target word w_i, $i = 1 \ldots N$ in vocabulary V of training corpus C, we collect its N_c neighboring context words to form a pseudo document, which reflects the meaning of w_i. The collection of all these extracted pseudo documents forms a new training data $D = \{d_1, d_2, \ldots d_N\}$ that can be used to induce posterior topic distributions of these pseudo documents via topic models. Note that the context words are also in vocabulary V and the number of pseudo documents we extracted is exactly as many as that of individual words found in C. Besides, we preprocess the training corpus by removing numbers, punctuation, stop words[1] and infrequent words to speed up and refine model training.

In **Step 2**, we induce the distributed representations of word meaning via topic models. We assume that all the target words found in the training corpus share a global set of latent meanings or senses[2] $Z = \{z_k|k{:}1 \ldots K\}$, where K is the prior number of senses. Using topic models, we regard each pseudo document as a multinomial distribution over these latent senses, and each latent sense is also a multinomial distribution over words. Based on these conditions, the meaning of a target word can be modeled as the posterior distribution of its pseudo document over this set of latent variables. Formally, the meaning representation of target word w_i is defined as follows:

$$v(w_i) = (P(z_1|d_i), P(z_2|d_i), \ldots, P(z_K|d_i))$$

In our work, we accomplish this step with standard PLSA and our model RCPLSA. And we will expand on the latter one in the following sections.

In **Step 3**, with the above-mentioned word meaning representations based on topic model, we compute the relatedness between two words using Cosine distance function. Given two representation vectors v_1 and v_2, we apply Cosine distance function to compute their distance:

$$cos(v_1, v_2) = \frac{\langle v_1, v_2 \rangle}{||v_1|| \cdot ||v_2||} \tag{1}$$

IV. THE PROPOSED TOPIC MODEL

By introducing lexical resource as constraints into the conventional PLSA model, we extend PLSA to a lexical resource-constrained one for word relatedness measurement.

A. OBJECTIVE FUNCTION

Our model is extended from PLSA, so let us first review the objective function of PLSA. Formally, the log likelihood, which represents the probability of PLSA generating training data D, can be expressed as follows:

$$L(D) = \sum_{j=1}^{M} \sum_{i=1}^{N} n(w_i, d_j) \log \sum_{k=1}^{K} p(z_k|d_j) p(w_i|z_k) \tag{2}$$

where M is the number of pseudo documents,[3] $n(w_i, d_j)$ denotes the co-occurrence frequency of word w_i in document d_j, and $p(w_i|z_k)$ is the probability of word w_i given topic z_k. Note that PLSA is based on conditional independence assumption, namely that documents and words are assumed independent conditioned on the associated latent variables, which means the interaction between documents is neglected. As a result, target word meaning representations, denoted by pseudo document-topic distributions, are calculated with complete independency. Hence, the potential of PLSA for word relatedness is far from being fulfilled.

To deal with the above-mentioned deficiency, we extend the objective function of PLSA with a pairwise regularization term using lexical resource constraints. The lexical resource consisting of word pairs that are known to be related can be obtained easily. We represent all the word pairs with a set $S = \{(w_u, w_v)|w_u, w_v \in V \;\&\&\; w_u \; and \; w_v \; is \; semantic \; related\}$. Here we mainly consider the word pairs with *synonym*, *hypernym*, *hyponym*, and *meronym* relations.

Our model is inspired by network regularized statistical topic model [44]. We call our model lexical resource-constrained Probabilistic Latent Semantic Analysis (RCPLSA). The main idea of our model is simple and natural that pseudo documents of known related words should be assigned similar topic distributions in latent space. Formally, the objective function of the proposed model is defined as:

$$O(D, S) = \lambda \frac{L(D)}{\sum_{j=1}^{M} \sum_{i=1}^{N} n(w_i, d_i)} - (1 - \lambda) \frac{R(D, S)}{|S|} \tag{3}$$

where $R(D, S)$ is the pairwise regularization term restrained by the related word pair set S, and $\lambda \in (0, 1)$ is a parameter used to balance the log likelihood and the pairwise regularization term. Note that RCPLSA can be simplified to PLSA when $\lambda = 1$. We consider to define $R(D, S)$ similar to the regularization term proposed by [44], and the definition is shown as follows:

$$R(D, S) = \frac{1}{2} \sum_{(w_u, w_v) \in S} \kappa_{u,v} \sum_{k=1}^{K} (p(z_k|d_u) - p(z_k|d_v))^2 \tag{4}$$

where d_u and d_v are corresponding pseudo documents of words w_u and w_v respectively and $\kappa_{u,v}$ is the weight of the related word pair (w_u, w_v), which is set to 1 in our experiments. Intuitively, $R(D, S)$ measures the topic distribution difference between w_u and w_v. Essentially, it is a penalty function for topic distributions of related word pairs. The more they differ, the larger $R(D, S)$ will be. In expectation, a topic model with small $R(D, S)$ should be obtained.

V. PARAMETER ESTIMATION

The standard method to optimize PLSA is Exception Maximization (EM) algorithm [45], which searches for the

[1] We use stopword list from NLTK(https://www.nltk.org/).

[2] To avoid misunderstanding, in the following sections, we still name latent semantic variables in topic model as *topic* rather than *sense*.

[3] As described before, the pseudo document number M is equal to the target word number N in training corpus. However, to keep consistent with conventional denotations in PLSA, we still use different symbols to denote these two numbers.

local maximum of objective function iteratively. However, RCPLSA is more complicated than PLSA, since its objective function has an additional pairwise regularization term. Hence, there exists no closed form solution for $O(D, S)$, and EM algorithm can not be directly applied to estimate parameters of RCPLSA. Inspired by [44], we apply GEM algorithm [46] to solve this problem. Similar to EM algorithm, GEM also has an expectation step (E-step) and a maximization step (M-step) in each iteration.

In the **E-step**, GEM algorithm computes the expectation of the complete likelihood $Q(\Psi; \Psi_n)$, where Ψ denotes all parameters $\{p(w_i|z_k), p(z_k|d_j)\}_{i,j,k}$ and Ψ_n is the values of all parameters estimated in the n-th iteration. In specific implementation, the distribution of the hidden variable, which indicates the probability of the word w_i in the document d_j assigned to the topic z_k, is computed as:

$$p(z_k|w_i, d_j) = \frac{p(w_i|z_k)p(z_k|d_j)}{\sum_{k'=1}^{K} p(w_i|z_{k'})p(z_{k'}|d_j)} \tag{5}$$

In the **M-step**, GEM algorithm finds a better estimation of parameters Ψ_{n+1} to maximize $Q(\Psi; \Psi_n)$, which is defined as follows:

$$\begin{aligned} Q(\Psi; \Psi_n) &= \lambda \frac{L'(D)}{\sum_{j=1}^{M}\sum_{i=1}^{N} n(w_i, d_i)} - (1-\lambda)\frac{R(D,S)}{|S|}, \\ L'(D) &= \sum_{j=1}^{M}\sum_{i=1}^{N} n(w_i, d_j) \sum_{k=1}^{K} p(z_k|w_i, d_j) \\ &\quad \times \log p(z_k|d_j)p(w_i|z_k) \end{aligned} \tag{6}$$

with the constraints that $\sum_{k=1}^{K} p(z_k|d_j) = 1$ and $\sum_{i=1}^{N} p(w_i|z_k) = 1$.

As previously mentioned, we are unable to directly update parameters using EM algorithm in the M-step for $Q(\Psi; \Psi_n)$. Fortunately, according to GEM algorithm, we do not have to find the local maximum of $Q(\Psi; \Psi_n)$ at every M-step. Instead, we only need to find a better set of values Ψ_{n+1} for Ψ that ensures $Q(\Psi; \Psi_{n+1}) \geq Q(\Psi; \Psi_n)$. GEM does that through optimizing the likelihood part and the regularization part of the objective function separately. To be specific, we divide each M-step into the following two sub-steps:

- First, we search for a set of values $\Psi_{n+1}^{(0)}$ which maximizes $L'(C)$ instead of the whole $Q(\Psi; \Psi_n)$. The computation of this sub-step is similar to the M-step of EM algorithm. Formally, parameters of RCPLSA can also be re-estimated as follows:

$$p(z_k|d_j) = \frac{\sum_{i=1}^{N} n(w_i, d_j)p(z_k|w_i, d_j)}{\sum_{i=1}^{N} n(w_i, d_j)} \tag{7}$$

$$p(w_i|z_k) = \frac{\sum_{j=1}^{M} n(w_i, d_j)p(z_k|w_i, d_j)}{\sum_{i'=1}^{N}\sum_{j=1}^{M} n(w_{i'}, d_j)p(z_k|w_{i'}, d_j)} \tag{8}$$

TABLE 1. Statistics of External Lexical Resource. #Word Pair = word pair number.

Language	Lexical Resource	#Word Pair
English	WordNet	66, 884
Chinese	HIT Thesaurus	15, 436

- Second, we gradually decrease $R(D, S)$ by repeatedly applying Newton-Raphson method to obtain a better set of values $\Psi_{n+1}^{(t+1)}$ according to $\Psi_{n+1}^{(t)}$, where $\Psi_{n+1}^{(t)}$ denotes values of Ψ_{n+1} in the t-th inner iteration. Note that the estimation of $p(w_i|z_k)$ does not rely on the regularization term, so that we only need to make further re-estimation of $p(z_k|d_j)$. For this, we start from $\Psi_{n+1}^{(0)}$ obtained in the previous sub-step and try the following equation to re-estimate the parameters until $Q(\Psi_{t+1}; \Psi_t) < Q(\Psi_t; \Psi_t)$:

$$\begin{aligned} p_{n+1}^{(t+1)}(z_k|d_u) &= (1-\alpha)p_{n+1}^{(t)}(z_k|d_u) \\ &\quad + \alpha \frac{\sum_{(w_u,w_v)\in S} \kappa_{u,v} p_{n+1}^{(t)}(z_k|d_u)}{\sum_{(w_u,w_v)\in S} \kappa_{u,v}} \end{aligned} \tag{9}$$

To guarantee that $\sum_k P(z_k|d_j) = 1$, we normalize the parameters $\{p(z_k|d_j)\}_k$ for document d_j after each smoothing step. If there is no $\Psi_{n+1}^{(t+1)}$ ensuring iterative condition, we then consider $\Psi_{n+1}^{(t)}$ as the local maximum point of the objective function Equation (6), and continue to next E-step.

VI. EXPERIMENTS

In order to investigate the effectiveness and versatility of our approach, we conducted semantic relatedness experiments on Chinese and English data sets with PLSA and RCPLSA respectively. After that we analyzed modeling results through topic extraction.

A. EXPERIMENT SETUP

First of all, we set $N_c = 5$ for all experiments in our work. That means, for each word, we collect 10 neighboring context words in total, 5 from the left and 5 from the right, to form its pseudo document.

Table 1 shows statistics of external lexical resource in our experiments.

In English experiments, our training data is mainly from Wikipedia. Specifically, using the above-mentioned approach, we finally obtained 91,189,675 pseudo documents for model training. We used three test sets in word relatedness experiments: MTURK-287,[4] MTURK-771[5] and WS-353[6] comprised of 287, 771 and 353 word pairs respectively. The external lexical resource is extracted from WordNet.[7]

In Chinese experiments, the training data is sampled from LDC which mainly includes LDC2005T10 and the

[4]http://tx.technion.ac.il/~kirar/Datasets.html
[5]http://www2.mta.ac.il/~gideon/mturk771.html
[6]http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353
[7]http://wordnet.princeton.edu

TABLE 2. Experimental results for PLSA using different topic numbers. #Topic denotes the number of topic.

Corpus \ #Topic	100	150	200	250	300
MTURK-287	0.637	0.676	0.628	0.643	0.636
MTURK-771	0.544	0.559	0.551	0.554	0.542
WS-353	0.627	0.669	0.638	0.634	0.619
WORD-240	0.398	0.492	0.479	0.490	0.468

AFP_CMN part of LDC2007T38. Further, we collected 62,398,617 pseudo documents to train various models. We used WORD-240[8] [47] comprised of 240 word pairs as test data. As for the external lexical resource, we used word pairs from HIT Thesaurus[9] for experiment.

B. WORD RELATEDNESS

In the first group of experiments, we conducted experiments on Chinese and English data sets to verify the effectiveness of RCPLSA compared with PLSA through word relatedness computation.

1) RESULTS OF PLSA MODELING

To compare our model with PLSA, we first conducted experiments using PLSA to find out the best topic numbers for comparison. Here we trained PLSA with different topic numbers: 100, 150, 200, 250 and 300. We applied cosine distance function to compute relatedness within each word pair from test sets according to the distributed word representations generated by PLSA. After that, we compute Spearman's rank order correlation coefficient[10] between predictions of different topic models and human-annotated orders.

Experimental results reported in Table 2 shows that PLSA achieves its best performance when K are taken the value of 150, and thus we set K as 150 when modeling with PLSA and RCPLSA for further investigation.

2) RESULTS OF RCPLSA MODELING

When comparing with PLSA, we set $K = 150$ for both PLSA and RCPLSA, with which PLSA reaches its best results. Besides, we set the following parameters for RCPLSA: constraint weight $\lambda = 5e\text{-}5$, learning rate $\alpha = 0.3$ (following [44]).

Results are listed in Table 3. We find that our model is superior to the conventional PLSA, no matter what test data set we use. Specifically, our model achieves 4.7, 3.9, 0.9 and 2.9 percentage points higher than PLSA on the four data sets, respectively.

[8]http://download.csdn.net/detail/chjshan55/3462335

[9]http://www.ltp-cloud.com/download

[10]Since Spearman's correlation coefficient is considered to be much more robust than Pearson's linear correlation, we use this metric in the experiments.

TABLE 3. Experimental results of word relatedness. $K = 150$, $\lambda = 5e\text{-}5$, and $\alpha = 0.3$.

Corpus \ Model	PLSA	RCPLSA
MTURK-287	0.643	0.690
MTURK-771	0.559	0.598
WS-353	0.634	0.643
WORD-240	0.492	0.521

C. EFFECTS OF PARAMETER SETTINGS

In the second group of experiments, we studied effects of parameter settings on our model. There are mainly three parameters that can influence the results: (a) Topic number K. (b) Constraint weight λ. (c) Learning rate α.

1) THE EFFECT OF TOPIC NUMBER K

Topic number K represents the dimensionality of latent factor space. Using the whole lexical resource, we built RCPLSA models with constraint weight $\lambda = 5e\text{-}5$, learning rate $\alpha = 0.3$, and different topic numbers: 100, 150, 200, 250 and 300.

Experimental results for RCPLSA compared with PLSA using different topic numbers are reported in Table 4. Generally, our model shows fair robustness over different topic numbers.

2) THE EFFECT OF CONSTRAINT WEIGHT λ

Constraint weight λ controls the impact extent of external lexical resource constraint. Using the whole lexical resource, we still used $\alpha = 0.3$, $K = 150$ to train RCPLSA models with different constraint weights: $5e$-6, $1e$-5, $5e$-5, $1e$-4, $5e$-4 and $1e$-3.

Table 5 reports the effects of different constraint weights. Optimal performances are obtained when $\lambda = 5e\text{-}5$. In contrast, these constraints with $\lambda = 1e\text{-}5$, $\lambda = 5e\text{-}6$ and $\lambda = 1e\text{-}6$ becomes counter-productive. But the performance of RCPLSA is still better than that of PLSA. This result indicates the robustness of our model over different constraint weights.

3) THE EFFECT OF LEARNING RATE α

Learning Rate α determines how far the inner iteration step is and how fast the algorithm will converge. In our experiments, we fix the constraint weight $\lambda = 5e\text{-}5$ and topic number $K = 150$ to train different RCPLSA models with different learning rates: 0.5, 0.3, 0.1 and 0.05.

Table 6 presents effects of different learning rates. Obviously, for all datasets, the best results are achieved when the learning rate α takes the value of 0.3, which coincides with the study of [44]. When α is set as 0.1 or 0.05, the performance of RCPLSA degrades because the effect of lexical resource constraint is reduced.

D. RESULT ANALYSIS

In this part, we analyze the modeling results through topic extraction. In order to qualitatively compare the topic

TABLE 4. Experimental results for PLSA and RCPLSA using different topic numbers with $\lambda = 5e\text{-}5$ and $\alpha = 0.3$. We highlight the highest relatedness score in bold for each set.

Corpus \ #Topic	100	150	200	250	300
MTURK-287	0.637/0.642	0.676/**0.690**	0.628/0.633	0.643/0.675	0.636/0.610
MTURK-771	0.544/0.562	0.559/**0.598**	0.551/0.584	0.554/0.555	0.542/0.577
WS-353	0.627/0.631	0.669/**0.643**	0.638/0.630	0.634/0.667	0.619/0.627
WORD-240	0.398/0.495	0.492/**0.521**	0.479/0.501	0.490/0.000	0.468/0.482

TABLE 5. Experimental results for RCPLSA using different constraint weights with $K = 150$ and $\alpha = 0.3$. Note that the second column shows the results of PLSA.

Corpus \ λ	-	5e-6	1e-5	5e-5	1e-4	5e-4	1e-3
MTURK-287	0.643	0.662	0.670	0.690	0.676	0.659	0.651
MTURK-771	0.559	0.569	0.576	0.598	0.572	0.562	0.552
WS-353	0.634	0.636	0.641	0.643	0.641	0.635	0.634
WORD-240	0.492	0.513	0.514	0.521	0.513	0.511	0.509

TABLE 6. Experimental results for RCPLSA using different learning rates with $K = 150$ and $\lambda = 5e\text{-}5$. We still provide the results of PLSA in the second column.

Corpus \ α	-	0.5	0.3	0.1	0.05
MTURK-287	0.643	0.664	0.690	0.639	0.603
MTURK-771	0.559	0.557	0.598	0.558	0.566
WS-353	0.634	0.630	0.643	0.627	0.637
WORD-240	0.492	0.492	0.521	0.516	0.508

TABLE 7. Examples of English topic modeling. $K = 150$, $\lambda = 5e\text{-}5$, and $\alpha = 0.3$.

PLSA1	PLSA2	RCPLSA1	RCPLSA2	RCPLSA3	RCPLSA4	RCPLSA5
province	south	province	city	south	japan	month
china	korea	china	beijing	korea	saturday	july
city	asian	guangdong	shanghai	korean	japanese	october
provincial	asia	provincial	provincial	dprk	campaign	april
guangdong	japan	local	guangdong	kim	tokyo	september
local	korean	jiangsu	local	republic	national	november
capital	dprk	zhejiang	capital	seoul	rally	thursday
country	kim	shandong	county	north	yen	august
guangzhou	republic	sichuan	guangzhou	rok	hashimoto	january
jiangsu	southeast	southwest	jiangsu	pyongyang	war	december

modeling results of the two models, we compared the word distributions of the topics, which are trained with PLSA and RCPLSA from the first group of experiments respectively.

Table 7 shows some English topical words learned using PLSA and RCPLSA, respectively. Although the semantic topics extracted by PLSA and RCPLSA are similar, there are still something differences between them. Compared with PLSA, RCPLSA can achieve finer granularity in topics extraction. For example, PLSA extracts one topic related to "*Chinese Province and City*" (the 1st column), while RCPLSA divides it into two distinguishable topics: "*Chinese Province*" (the 4th column) and "*Chinese City*" (the 5th column). Also, the topic mentioned about "*Northeast Asia*" in PLSA (the 2nd column) is further divided into two topics, i.e. "*Korea*" (the 6th column) and "*Japan*" (the 7th column), by RCPLSA. Furthermore, RCPLSA extract a topic about "*Time and Date*" (the 3th column), which is not isolated as an independent topic in PLSA.

Table 8 lists some Chinese topical words extracted by PLSA and RCPLSA, respectively. In order to improve the readability of these topics, we append English translation to each Chinese word in the table. Similar to English experiments, topics obtained in RCPLSA shows finer granularity than the ones in PLSA. For instance, PLSA extracts one topic which mentions both "中国两岸(*Cross-straits of China*)" and "东盟各国(*ASENAN countries*)" (the 1st

TABLE 8. Examples of Chinese topic modeling. $K = 150$, $\lambda = 5e\text{-}5$, and $\alpha = 0.3$.

PLSA1	RCPLSA1	RCPLSA2	RCPLSA3
台湾(Taiwan)	中国(China)	新加坡(Singapore)	六月(June)
亚洲(Asia)	台湾(Taiwan)	泰国(Thailand)	十一月(November)
中国(China)	北京(Beijing)	美国(American)	七月(July)
马来西亚(Malaysia)	香港(Hong Kong)	越南(Vietnam)	八月(August)
地区(region)	大陆(Main land)	政治(politics)	三月(March)
柬埔寨(Cambodia)	胡锦涛(Jintao Hu)	马来西亚(Malaysia)	九月(September)
泰国(Thailand)	中共(CPC)	亚洲(Asia)	五月(May)
政府(government)	组织(organization)	东协(ASEAN)	一日(1st)
东南亚(Southeast asia)	陈水扁(Shui-Bian Chen)	区域(region)	四月(April)
陈水扁(Shui-Bian Chen)	上海(Shanghai)	经济(economy)	三十日(30th)

column), while this topic is detected as two seperate topics about "中国两岸(*Cross-straits of China*)" (the 2nd column) and "东盟各国(*ASENAN countries*)" (the 3th column) in RCPLSA. Besides, the topic about "*Time and Date*" in RCPLSA (the 4th column) is not found in PLSA.

The above-mentioned results demonstrate the effectiveness of RCPLSA in fine-grained topic modeling.

VII. CONCLUSION

In this paper, we proposed a lexical resource-constrained topic model for word relatedness. Our model is an extension of PLSA. It refines the training of model parameters by using word pairs known to be related as constraints. Compared with the previous approaches, the proposed model has the advantages of the conventional statistics-based approaches and lexical resource-based approaches. Experimental results on English and Chinese data sets demonstrate the effectiveness and generality of our model.

In the future, we plan to enlarge the scale of lexical resource to further improve our model. Besides, our model is incapable of distinguishing the various meaning of polysemous word. Thus, how to distinguish the different meanings of polysemous words will become another study emphasis in our further research.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their insightful comments.

REFERENCES

[1] A. Budanitsky and G. Hirst, "Evaluating wordnet-based measures of lexical semantic relatedness," *Comput. Linguistics*, vol. 32, no. 1, pp. 13–47, Mar. 2006.
[2] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, pp. 613–620, Nov. 1975.
[3] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. NIPS*, 2013, pp. 3111–3119.
[4] J. Morris and G. Hirst, "Lexical cohesion computed by thesaural relations as an indicator of the structure of text," *Comput. Linguistics*, vol. 17, no. 1, pp. 21–48, Mar. 1991.
[5] H. Kozima and T. Furugori, "Similarity between words computed by spreading activation on an english dictionary," in *Proc. 6th Conf. Eur. Chapter Assoc. Comput. Linguistics*, Apr. 1993, pp. 232–239.
[6] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," in *Proc. IJCAI*, Aug. 1995, pp. 448–453.
[7] J. J. Jiang and D. W. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," in *Proc. ROCLING*, 1997, pp. 19–33.
[8] H. Kozima and A. Ito, "Context-sensitive word distance by adaptive scaling of a semantic space," in *Proc. RANLP*, 1997, pp. 111–124.
[9] M. Jarmasz and S. Szpakowic, "Roget 's thesaurus and semantic similarity," in *Proc. RANLP*, 2003, pp. 212–219.
[10] S. Banerjee and T. Pedersen, "Extended gloss overlaps as a measure of semantic relatedness," in *Proc. IJCAI*, Aug. 2003, pp. 805–810.
[11] T. Hughes and D. Ramage, "Lexical semantic relatedness with random graph walks," in *Proc. EMNLP*, 2007, pp. 581–589.
[12] D. Milne and I. H. Witten, "An effective, low-cost measure of semantic relatedness obtained from wikipedia links," in *Proc. AAAI*, 2008, pp. 25–30.
[13] E. Yeh, D. Ramage, C. D. Manning, E. Agirre, and A. Soroa, "Wikiwalk: Random walks on wikipedia for semantic relatedness," in *Proc. Workshop Graph-Based Methods Natural Lang. Process.*, 2009, pp. 41–49.
[14] G. Tsatsaronis, I. Varlamis, M. Vazirgiannis, and K. Norvag, "Omiotis: A thesaurus-based measure of text relatedness," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2009, pp. 742–745.
[15] G. Tsatsaronis, I. Varlamis, and M. Vazirgiannis, "Text relatedness based on a word thesaurus," *J. Artif. Intell. Res.*, vol. 37, no. 1, pp. 1–40, Jan. 2010.
[16] P. Resnik, "Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language," *J. Artif. Intell. Res.*, vol. 11, pp. 95–130, Jul. 2011.
[17] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Pa ca, and A. Soroa, "A study on similarity and relatedness using distributional and wordnet-based approaches," in *Proc. HLT-NAACL*, Jun. 2009, pp. 19–27.
[18] D. Bollegala, Y. Matsuo, and M. Ishizuka, "Measuring semantic similarity between words using web search engines," in *Proc. 16th Int. Conf. World Wide Web*, May 2007, pp. 757–766.
[19] M. Strube and S. P. Ponzetto, "Wikirelate! computing semantic relatedness using wikipedia," in *Proc. AAAI*, 2006, pp. 1419–1424.
[20] S. P. Ponzetto and M. Strube, "Knowledge derived from wikipedia for computing semantic relatedness," *J. Artif. Intell. Res.*, vol. 30, pp. 181–212, Oct. 2007.
[21] Y. Haralambous and V. Klyuev, "A semantic relatedness measure based on combined encyclopedic, ontological and collocational knowledge," in *Proc. AFNLP*, 2011, pp. 1397–1402.
[22] R. El-Yaniv and D. Yanay, "Supervised learning of semantic relatedness," in *Proc. ECML-PKDD*, 2012, pp. 744–759.
[23] G. Halawi, G. Dror, E. Gabrilovich, and Y. Koren, "Large-scale learning of word relatedness with constraints," in *Proc. SIGKDD*, Aug. 2012, pp. 1406–1414.
[24] A. Celikyilmaz, D. Hakkani-Tur, P. Pasupat, and R. Sarikaya, "Enriching word embeddings using knowledge graph for semantic tagging in conversational dialog systems," in *Proc. AAAI*, 2015, pp. 19–27.
[25] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *J. Amer. Soc. Inf. Sci.*, vol. 41, no. 6, pp. 391–407, 1990.
[26] C. H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala, "Latent semantic indexing: A probabilistic analysis," *J. Comput. Syst. Sci.*, vol. 61, no. 2, pp. 217–235, Oct. 2000.
[27] L. Finkelstein *et al.*, "Placing search in context: The concept revisited," in *Proc. 10th Int. Conf. World Wide Web*, vol. 20, May 2001, pp. 406–414.

[28] J. Boyd-Graber, D. Blei, and X. Zhu, "A topic model for word sense disambiguation," in *Proc. Joint Conf. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learn.*, Jun. 2007, pp. 1024–1033.
[29] Q. Sun, R. Li, D. Luo, and X. Wu, "Text segmentation with LDA-based Fisher kernel," in *Proc. 46th Annu. Meeting Assoc. Comput. Linguistics Hum. Lang. Technol., Short Papers*, Jun. 2008, pp. 269–272.
[30] D. Lin, "An information-theoretic definition of similarity," in *Proc. ICML*, vol. 98, 1998, pp. 296–304.
[31] I. Dagan, L. Lee, and F. C. N. Pereira, "Similarity-based models of word co-occurrence probabilities," *Mach. Learn.*, vol. 34, nos. 1–3, pp. 43–69, 1999.
[32] E. Terra and C. L. A. Clarke, "Frequency estimates for statistical word similarity measures," in *Proc. NAACL*, Jun. 2003, pp. 165–172.
[33] J. Reisinger and R. J. Mooney, "Multi-prototype vector-space models of word meaning," in *Proc. NAACL*, Jun. 2010, pp. 109–117.
[34] S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, and R. Harshman, "Using latent semantic analysis to improve access to textual information," in *Proc. SIGCHI*, May 1988, pp. 281–285.
[35] D. M. Blei and J. D. Lafferty, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, pp. 993–1022, Jan. 2003.
[36] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," in *Proc. J. Mach. Learn. Res.*, Feb. 2003, pp. 1137–1155.
[37] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. ICML*, Jul. 2008, pp. 160–167.
[38] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proc. EMNLP*, 2014, pp. 1532–1543.
[39] E. Gabrilovich and S. Markovitch, "Wikipedia-based semantic interpretation for natural language processing," *J. Artif. Intell. Res.*, vol. 34, pp. 443–498, Mar. 2009.
[40] K. Radinsky, E. Agichtein, E. Gabrilovich, and S. Markovitch, "A word at a time: Computing word relatedness using temporal semantic analysis," in *Proc. 20th Int. Conf. World Wide Web*, Apr. 2011, pp. 337–346.
[41] T. Zesch and I. Gurevych, "Wisdom of crowds versus wisdom of linguists—measuring the semantic relatedness of words," *Natural Lang. Eng.*, vol. 16, no. 1, pp. 25–59, 2010
[42] G. Dinu and M. Lapata, "Measuring distributional similarity in context," in *Proc. EMNLP*, Oct. 2010, pp. 1162–1172.
[43] D. Xiong and M. Zhang, "A sense-based translation model for statistical machine translation," in *Proc. ACL*, 2014, pp. 1459–1469.
[44] Q. Mei, D. Cai, D. Zhang, and C. Zha, "Topic modeling with network regularization," in *Proc. 17th Int. Conf. World Wide Web*, 2008, pp. 101–110.
[45] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Roy. Stat. Soc.*, vol. 39, no. 1, pp. 1–22, 1977.
[46] R. M. Neal and G. E. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants," in *Learning in Graphical Models*. Dordrecht, The Netherlands: Kluwer, 1998, pp. 355–368.
[47] X. Wang, Y. Jia, B. Zhou, Z. Y. Ding, and Z. Liang, "Computing semantic relatedness using Chinese Wikipedia links and taxonomy," *Chin. Comput. Syst.*, vol. 32, no. 11, pp. 2237–2242, 2011.

YONGJING YIN received the B.S. degree in software engineering from Xiamen University, Xiamen, China, in 2017, where he is currently pursuing the M.S. degree with the Software School, under the supervision of Prof. J. Su. His main research interest includes natural language processing.

JIALI ZENG was born in 1994. She received the bachelor's degree in software engineering from the Software School, Xiamen University, Xiamen, China, where she is currently pursuing the master's degree with the Software School, under the supervision of Prof. J. Su. Her major research interests include natural language processing and neural machine translation.

HONGJI WANG received the Ph.D. degree in computer software and theory from the Institute of Software, Chinese Academy of Sciences, in 2005.

He is currently an Associate Professor with the Software School, Xiamen University. His research interests include information security and data analysis.

KEQING WU received the M.S. degree in software engineering from the Software School, Xiamen University, in 2013, where she is currently a Software Engineer.

Her research interests include information management and software engineering.

BIN LUO received the master's degree in software engineering from the Software School, Xiamen University, in 2013, where he is currently an Engineer.

His research interest includes software engineering.

JINSONG SU was born in 1982. He received the Ph.D. degree from the Chinese Academy of Sciences, Beijing, China.

He is currently an Associate Professor with Xiamen University, Xiamen, China. His research interests include natural language processing and machine translation.