LTSG: Latent Topical Skip-Gram for Mutually Improving Topic Model and Vector Representations

Jarvan Law, Hankz Hankui Zhuo(✉), JunHua He, and Erhu Rong

Department of Computer Science, Sun Yat-Sen University, GuangZhou 510006, China
JarvanLaw@gmail.com, zhuohank@mail.sysu.edu.cn,
{hejunh,rongerhu}@mail2.sysu.edu.cn

Abstract. Topic models have been widely used in discovering latent topics which are shared across documents in text mining. Vector representations, word embeddings and topic embeddings, map words and topics into a low-dimensional and dense real-value vector space, which have obtained high performance in NLP tasks. However, most of the existing models assume the results trained by one of them are perfect correct and used as prior knowledge for improving the other model. Some other models use the information trained from external large corpus to help improving smaller corpus. In this paper, we aim to build such an algorithm framework that makes topic models and vector representations mutually improve each other within the same corpus. An EM-style algorithm framework is employed to iteratively optimize both topic model and vector representations. Experimental results show that our model outperforms state-of-the-art methods on various NLP tasks.

Keywords: Topic modeling · Polysemous-word · Word embeddings
Text mining

1 Introduction

Word embeddings, e.g., distributed word representations [16], represent words with low dimensional and dense real-value vectors, which capture useful semantic and syntactic features of words. Distributed word embeddings can be used to measure word similarities by computing distances between vectors, which have been widely used in various IR and NLP tasks, such as entity recognition [23], disambiguation [5] and parsing [21]. Despite the success of previous approaches on word embeddings, they all assume each word has a specific meaning and represent each word with a single vector, which restricts their applications in fields with polysemous words, e.g., "bank" can be either "a financial institution" or "a raised area of ground along a river".

To overcome this limitation, [14] propose a topic embedding approach, namely Topical Word Embeddings (TWE), to learn topic embeddings to characterize various meanings of polysemous words by concatenating topic embeddings

J.-H. Lai et al. (Eds.): PRCV 2018, LNCS 11258, pp. 375–387, 2018.
https://doi.org/10.1007/978-3-030-03338-5_32

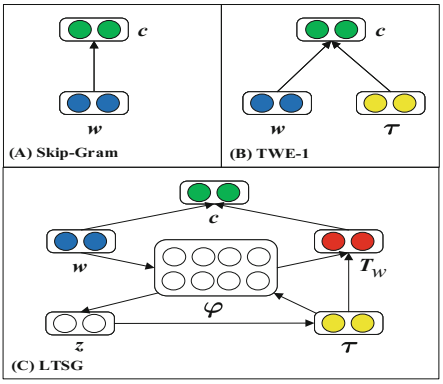

Fig. 1. Skip-Gram, TWE and LTSG models. Blue, yellow, green circles denote the embeddings of word, topic and context, while red circles in LTSG denote the global topical word. White circles denote the topic model part, topic-word distribution $\boldsymbol{\varphi}$ and topic assignment $\boldsymbol{z}$. (Color figure online)

with word embeddings. Despite the success of TWE, compared to previous multi-prototype models [11,20], it assumes that word distributions over topics are provided by off-the-shelf topic models such as LDA, which would limit the applications of TWE once topic models do not perform well in some domains [19]. As a matter of fact, pervasive polysemous words in documents would harm the performance of topic models that are based on co-occurrence of words in documents. Thus, a more realistic solution is to build both topic models with regard to polysemous words and polysemous word embeddings simultaneously, instead of using off-the-shelf topic models. In this work, we propose a novel learning framework, called Latent Topical Skip-Gram (LTSG) model, to mutually learn polysemous-word models and topic models. To the best of our knowledge, this is the first work that considers learning polysemous-word models and topic models simultaneously. Although there have been approaches that aim to improve topic models based on word embeddings MRF-LDA [24], they fail to improve word embeddings provided words are polysemous; although there have been approaches that aim to improve polysemous-word models TWE [14] based on topic models, they fail to improve topic models considering words are polysemous. Different from previous approaches, we introduce a new node $\boldsymbol{T}_w$, called *global topic*, to capture all of the topics regarding polysemous word w based on topic-word distribution $\boldsymbol{\varphi}$, and use the global topic to estimate the context of polysemous word w. Then we characterize polysemous word embeddings by concatenating word embeddings with topic embeddings. We illustrate our new model in Fig. 1, where Fig. 1(A) is the skip-gram model [16], which aims to maximize the probability of context c given word w. Figure 1(B) is the TWE model, which extends the skip-gram model to maximize the probability of context c given both word w and topic t, and Fig. 1(C) is our LTSG model which aims to maximize the probability of context c given word w and global topic $\boldsymbol{T}_w$. $\boldsymbol{T}_w$ is generated based on topic-word distri-

bution $\boldsymbol{\varphi}$ (i.e., the joint distribution of topic embedding $\boldsymbol{\tau}$ and word embedding $\boldsymbol{w}$) and topic embedding $\boldsymbol{\tau}$ (which is based on topic assignment $\boldsymbol{z}$). Through our LTSG model, we can simultaneously learn word embeddings $\boldsymbol{w}$ and global topic embeddings $\boldsymbol{T}_w$ for representing polysemous word embeddings, and topic word distribution $\boldsymbol{\varphi}$ for mining topics with regard to polysemous words. We will exhibit the effectiveness of our LTSG model in text classification and topic mining tasks with regard to polysemous words in documents.

In the remainder of the paper, we first introduce preliminaries of our LTSG model, and then present our LTSG algorithm in detail. After that, we evaluate our LTSG model by comparing our LTSG algorithm to state-of-the-art models in various datasets. Finally we review previous work related to our LTSG approach and conclude the paper with future work.

2 Preliminaries

In this section, we briefly review preliminaries of Latent Dirichlet Allocation (LDA), Skip-Gram, and Topical Word Embeddings (TWE), respectively. We show some notations and their corresponding meanings in Table 1, which will be used in describing the details of LDA, Skip-Gram, and TWE.

Table 1. Notations of the text collection.

Term	Notation	Definition or description
Vocabulary	$\boldsymbol{\mathcal{V}}$	Set of words in the text collection, $\lvert\boldsymbol{\mathcal{V}}\rvert = W$
Word	w	A basic item from vocabulary indexed as $w \in \{1, 2, \ldots, W\}$
Document	$\mathbf{w}$	A sequence of N words, $\mathbf{w} = (w_1, w_2, \ldots, w_N)$
Corpus	$\boldsymbol{\mathcal{D}}$	A collection of M documents, $\boldsymbol{\mathcal{D}} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_M\}$
Topic-word	$\boldsymbol{\varphi}$	K distributions over vocabulary ($K \times W$ matrix), $\lvert\boldsymbol{\varphi}\rvert = K, \lvert\boldsymbol{\varphi}_k\rvert = W$
Word embedding	$\boldsymbol{v}$	Distributed representation of *word*, denoted by $\boldsymbol{v}_w$, $\boldsymbol{v} \in \mathbb{R}^d$
Topic embedding	$\boldsymbol{\tau}$	Distributed representation of *topic*, denoted by $\boldsymbol{\tau}_k$, $\boldsymbol{\tau} \in \mathbb{R}^d$

2.1 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) [2], a three-level hierarchical Bayesian model, is a well-developed and widely used probabilistic topic model. Extending Probabilistic Latent Semantic Indexing (PLSI) [10], LDA adds Dirichlet priors to document-specific topic mixtures to overcome the overfitting problem in PLSI. LDA aims at modeling each document as a mixture over sets of topics, each associated with a multinomial word distribution. Given a document corpus $\boldsymbol{\mathcal{D}}$, each document $\mathbf{w}_m \in \boldsymbol{\mathcal{D}}$ is assumed to have a distribution over K topics. The generative process of LDA is shown as follows,

1. For each topic $k = 1 \rightarrow K$, draw a distribution over words $\boldsymbol{\varphi}_k \sim Dir(\boldsymbol{\beta})$
2. For each document $\mathbf{w}_m \in \boldsymbol{\mathcal{D}}, m \in \{1, 2, \ldots, M\}$
   (a) Draw a topic distribution $\boldsymbol{\theta}_m \sim Dir(\boldsymbol{\alpha})$
   (b) For each word $w_{m,n} \in \mathbf{w}_m, n = 1, \ldots, N_m$
      i. Draw a topic assignment $z_{m,n} \sim Mult(\boldsymbol{\theta}_m)$, $z_{m,n} \in \{1, \ldots, K\}$.
      ii. Draw a word $w_{m,n} \sim Mult(\boldsymbol{\varphi}_{z_{m,n}})$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are Dirichlet hyperparameters, specifying the nature of priors on $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$. Variational inference and Gibbs sampling are the common ways to learn the parameters of `LDA`.

2.2 The Skip-Gram Model

The Skip-Gram model is a well-known framework for learning word vectors [16]. Skip-Gram aims to predict context words given a target word in a sliding window, as shown in Fig. 1(A).

Given a document corpus $\boldsymbol{\mathcal{D}}$ defined in Table 1, the objective of Skip-Gram is to maximize the average log-probability

$$\mathcal{L}(\boldsymbol{\mathcal{D}}) = \frac{1}{\sum_{m=1}^{M} N_m} \sum_{m=1}^{M} \sum_{n=1}^{N_m} \sum_{-c \le j \le c, j \ne 0} \log \Pr(w_{m,n+j} | w_{m,n}), \tag{1}$$

where c is the context window size of the target word. The basic Skip-Gram formulation defines $\Pr(w_{m,n+j}|w_{m,n})$ using the softmax function:

$$\Pr(w_{m,n+j} | w_{m,n}) = \frac{\exp(\boldsymbol{v}_{w_{m,n+j}} \cdot \boldsymbol{v}_{w_{m,n}})}{\sum_{w=1}^{W} \exp(\boldsymbol{v}_w \cdot \boldsymbol{v}_{w_{m,n}})}, \tag{2}$$

where $\boldsymbol{v}_{w_{m,n}}$ and $\boldsymbol{v}_{w_{m,n+j}}$ are the vector representations of target word $w_{m,n}$ and its context word $w_{m,n+j}$, and W is the number of words in the vocabulary $\boldsymbol{\mathcal{V}}$. Hierarchical softmax and negative sampling are two efficient approximation methods used to learn Skip-Gram.

2.3 Topical Word Embeddings

Topical word embeddings (`TWE`) is a more flexible and powerful framework for multi-prototype word embeddings, where topical word refers to a word taking a specific topic as context [14], as shown in Fig. 1(B). `TWE` model employs `LDA` to obtain the topic distributions of document corpora and topic assignment for each word token. `TWE` model uses topic $z_{m,n}$ of target word to predict context word compared with only using the target word $w_{m,n}$ to predict context word in Skip-Gram. `TWE` is defined to maximize the following average log probability

$$\mathcal{L}(\boldsymbol{\mathcal{D}}) = \frac{1}{\sum_{m=1}^{M} N_m} \sum_{m=1}^{M} \sum_{n=1}^{N_m} \sum_{-c \le j \le c, j \ne 0} \log \Pr(w_{m,n+j} | w_{m,n}) + \log \Pr(w_{m,n+j} | z_{m,n}). \tag{3}$$

TWE regards each topic as a pseudo word that appears in all positions of words assigned with this topic. When training TWE, Skip-Gram is being used for learning word embeddings. Afterwards, each topic embedding is initialized with the average over all words assigned to this topic and learned by keeping word embeddings unchanged.

Despite the improvement over Skip-Gram, the parameters of LDA, word embeddings and topic embeddings are learned separately. In other word, TWE just uses LDA and Skip-Gram to obtain external knowledge for learning better topic embeddings.

3 Our LTSG Algorithm

Extending from the TWE model, the proposed Latent Topical Skip-Gram model (LTSG) directly integrates LDA and Skip-Gram by using topic-word distribution $\boldsymbol{\varphi}$ mentioned in topic models like LDA, as shown in Fig. 1(C). We take three steps to learn topic modeling, word embeddings and topic embeddings simultaneously, as shown below.

Step 1. Sample topic assignment for each word token. Given a specific word token $w_{m,n}$, we sample its latent topic $z_{m,n}$ by performing Gibbs updating rule similar to LDA.

Step 2. Compute topic embeddings. We average all words assigned to each topic to get the embedding of each topic.

Step 3. Train word embeddings. We train word embeddings similar to Skip-Gram and TWE. Meanwhile, topic-word distribution $\boldsymbol{\varphi}$ is updated based on Eq. (10). The objective of this step is to maximize the following function

$$\mathcal{L}(\boldsymbol{\mathcal{D}}) = \frac{1}{\sum_{m=1}^{M} N_m} \sum_{m=1}^{M} \sum_{n=1}^{N_m} \sum_{-c \le j \le c, j \ne 0} \log \Pr(w_{m,n+j} | w_{m,n}) + \log \Pr(w_{m,n+j} | T_{w_{m,n}}), \tag{4}$$

where $T_{w_{m,n}} = \sum_{k=1}^{K} \boldsymbol{\tau}_k \cdot \varphi_{k,w_{m,n}}$. $\boldsymbol{\tau}_k$ indicates the k-th topic embedding. $T_{w_{m,n}}$ can be seen as a distributed representation of global topical word of $w_{m,n}$.

We will address the above three steps in detail below.

3.1 Topic Assignment via Gibbs Sampling

To perform Gibbs sampling, the main target is to sample topic assignments $z_{m,n}$ for each word token $w_{m,n}$. Given all topic assignments to all of the other words, the full conditional distribution $\Pr(z_{m,n} = k | \mathbf{z}^{-(m,n)}, \mathbf{w})$ is given below when applying collapsed Gibbs sampling [9],

$$\Pr(z_{m,n} = k | \mathbf{z}^{-(m,n)}, \mathbf{w}) \propto \frac{n_{k,w_{m,n}}^{-(m,n)} + \beta}{\sum_{w=1}^{w} n_{k,w}^{-(m,n)} + W\beta} \cdot \frac{n_{m,k}^{-(m,n)} + \alpha}{\sum_{k'=1}^{K} n_{m,k'}^{-(m,n)} + K\alpha}, \tag{5}$$

where $-(m,n)$ indicates that the current assignment of $z_{m,n}$ is excluded. $n_{k,w}$ and $n_{m,k}$ denote the number of word tokens w assigned to topic k and the count of word tokens in document m assigned to topic k, respectively. After sampling all the topic assignments for words in corpus $\boldsymbol{\mathcal{D}}$, we can estimate each component of $\boldsymbol{\varphi}$ and $\boldsymbol{\theta}$ by Eqs. (6) and (7).

$$\hat{\varphi}_{k,w} = \frac{n_{k,w} + \beta}{\sum_{w'=1}^{W} n_{k,w'} + W\beta} \tag{6}$$

$$\hat{\theta}_{d,k} = \frac{n_{m,k} + \alpha}{\sum_{k'=1}^{K} n_{m,k'} + K\alpha} \tag{7}$$

Unlike standard `LDA`, the topic-word distribution $\boldsymbol{\varphi}$ is used directly for constructing the modified Gibbs updating rule in `LTSG`. Following the idea of DRS [7], with the conjugacy property of Dirichlet and multinomial distributions, the Gibbs updating rule of our model `LTSG` can be approximately represented by

$$\Pr(z_{m,n} = k|\mathbf{w}, \mathbf{z}^{-(m,n)}, \boldsymbol{\varphi}, \boldsymbol{\alpha}) \propto \varphi_{k,w_{m,n}} \cdot \frac{n_{m,k}^{-(m,n)} + \alpha}{\sum_{k'=1}^{K} n_{m,k'}^{-(m,n)} + K\alpha}. \tag{8}$$

In different corpus or applications, Eq. (8) can be replaced with other Gibbs updating rules or topic models, eg. LFLDA [18].

3.2 Topic Embeddings Computing

Topic embeddings aim to approximate the latent semantic centroids in vector space rather than a multinomial distribution. `TWE` trains topic embeddings after word embeddings have been learned by Skip-Gram. In `LTSG`, we use a straightforward way to compute topic embedding for each topic. For the kth topic, its topic embedding is computed by averaging all words with their topic assignment z equivalent to k, i.e.,

$$\boldsymbol{\tau}_k = \frac{\sum_{m=1}^{M} \sum_{n=1}^{N_m} \mathbb{I}(z_{m,n} = k) \cdot \boldsymbol{v}_{w_{m,n}}}{\sum_{w=1}^{W} n_{k,w}} \tag{9}$$

where $\mathbb{I}(x)$ is indicator function defined as 1 if x is true and 0 otherwise.

Similarly, you can design your own more complex training rule to train topic embedding like TopicVec [13] and Latent Topic Embedding (`LTE`) [12].

3.3 Word Embeddings Training

`LTSG` aims to update $\boldsymbol{\varphi}$ during word embeddings training. Following the similar optimization as Skip-Gram, hierarchical softmax and negative sampling are used for training the word embeddings approximately due to the computationally expensive cost of the full softmax function which is proportional to vocabulary

size W. LTSG uses stochastic gradient descent to optimize the objective function given in Eq. (4).

The hierarchical softmax uses a binary tree (eg. a Huffman tree) representation of the output layer with the W words as its leaves and, for each node, explicitly represents the relative probabilities of its child nodes. There is a unique path from root to each word w and $node(w, i)$ is the i-th node of the path. Let $L(w)$ be the length of this path, then $node(w, 1) = root$ and $node(w, L(w)) = w$. Let $child(u)$ be an arbitrary child of node u, e.g. left child. By applying hierarchical softmax on $\Pr(w_{m,n+j}|T_{w_{m,n}})$ similar to $\Pr(w_{m,n+j}|w_{m,n})$ described in Skip-gram [16], we can compute the log gradient of $\boldsymbol{\varphi}$ as follows,

$$\frac{\partial \log \Pr(w_{m,n+j}|T_{w_{m,n}})}{\partial \varphi_{k=z_{m,n},w=w_{m,n}}} = \frac{1}{L(w_{m,n})-1} \sum_{i=1}^{L(w_{m,n})-1} \left[1 - h_{i+1}^{w_{m,n+j}} - \sigma(T_{w_{m,n}} \cdot \boldsymbol{v}_i^{w_{m,n+j}})\right] \boldsymbol{\tau}_k \cdot \boldsymbol{v}_i^{w_{m,n+j}}, \tag{10}$$

where $\sigma(x) = 1/(1 + \exp(-x))$. Given a path from root to word $w_{m,n+j}$ constructed by Huffman tree, $\boldsymbol{v}_i^{w_{m,n+j}}$ is the vector representation of i-th node. And $h_{i+1}^{w_{m,n+j}}$ is the Huffman coding on the path defined as $h_{i+1}^{w_{m,n+j}} = \mathbb{I}\big(node(w_{m,n+j}, i+1) = child(node(w_{m,n+j}, i)\big)$.

Follow this idea, we can compute the gradients for updating the word w and non-leaf node. From Eq. (10), we can see that $\boldsymbol{\varphi}$ is updated by using topic embeddings $\boldsymbol{\tau}_k$ directly and word embeddings indirectly via the non-leaf nodes in Huffman tree, which is used for training the word embeddings.

3.4 An Overview of Our LTSG algorithm

In this section we provide an overview of our LTSG algorithm, as shown in Algorithm 1. In line 1 in Algorithm 1, we run the standard LDA with certain iterations and initialize $\boldsymbol{\varphi}$ based on Eq. (6). From lines 4 to 6, there are the three steps mentioned in Sect. 3. From lines 7 to 13, $\boldsymbol{\varphi}$ will be updated after training the whole corpus $\boldsymbol{\mathcal{D}}$ rather than per word, which is more suitable for multi-thread training. Function $f(\xi, n_{k,w})$ is a dynamic learning rate, defined by $f(\xi, n_{k,w}) = \xi \cdot \log(n_{k,w})/n_{k,w}$. In line 16, document-topic distribution $\theta_{m,k}$ is computed to model documents.

4 Experiments

In this section, we evaluate our LTSG model in three aspects, i.e., contextual word similarity, text classification, and topic coherence.

We use the dataset 20NewsGroup, which consists of about 20,000 documents from 20 different newsgroups. For the baseline, we use the default settings of parameters unless otherwise specified. Similar to TWE, we set the number of topics $K = 80$ and the dimensionality of both word embeddings and topic embeddings $d = 400$ for all the relative models. In LTSG, we initialize $\boldsymbol{\varphi}$ with $init\_nGS = 2500$. We perform $nItrs = 5$ runs on our framework. We perform $nGS = 200$ Gibbs sampling iterations to update topic assignment with $\alpha = 0.01, \beta = 0.1$.

Algorithm 1. Latent Topical Skip-Gram

Input: corpus $\mathcal{D}$, # topics K, size of vocabulary W, Dirichlet hyperparameters α, β, # iterations of LDA for initialization $init\_nGS$, # iterations of framework $nItrs$, # Gibbs sampling iterations nGS.

Output: $\theta_{m,k}$, $\varphi_{k,w}$, $\boldsymbol{v}_w$, $\boldsymbol{\tau}_k$, $m = 1, 2, \ldots, M; k = 1, 2, \ldots, K; w = 1, 2, \ldots, W$

Initialization. Initialize $\varphi_{k,w}$ as in Equation (6) with $init\_nGS$ iterations in standard LDA as in Equation (5)
$i \leftarrow 0$
while $(i < nItrs)$ **do**
  Step 1. Sample $z_{m,n}$ as in Equation (8) with nGS iterations
  Step 2. Compute each topic embedding $\boldsymbol{\tau}_k$ as in Equation (9)
  Step 3. Train word embeddings with objective function as in Equation (4)
  Compute the first-order partial derivatives $\mathcal{L}'(\mathcal{D})$
  Set the learning rate ξ
  for $(k = 1 \rightarrow K)$ **do**
    for $(w = 1 \rightarrow W)$ **do**
      $\varphi_{k,w}^{(i+1)} \leftarrow \varphi_{k,w}^{(i)} + f(\xi, n_{k,w}) \frac{\partial \mathcal{L}'(\mathcal{D})}{\partial \varphi_{k,w}}$
    end for
  end for
  $i \leftarrow i + 1$
end while
Compute each $\theta_{m,k}$ as in Equation (7)

4.1 Contextual Word Similarity

To evaluate contextual word similarity, we use Stanford's Word Contextual Word Similarities (SCWS) dataset introduced by [11], which has been also used for evaluating state-of-art model [14]. There are totally 2,003 word pairs and their contexts, including 1328 noun-noun pairs, 399 verb-verb pairs, 140 verb-noun, 97 adjective-adjective, 30 noun-adjective, 9 verb-adjective pairs. Among all of the pairs, there are 241 same-word pairs which may show different meaning in the giving context. The dataset provide human labeled similarity scores based on the meaning in the context. For comparison, we compute the Spearman correlation similarity scores of different models and human judgments.

Following the TWE model, we use two scores AvgSimC and MaxSimC to evaluate the multi-prototype model for contextual word similarity. The topic distribution $\Pr(z|w, c)$ will be inferred by using $\Pr(z|w, c) \propto \Pr(w|z) \Pr(z|c)$ with regarding c as a document. Given a pair of words with their contexts, namely (w_i, c_i) and (w_j, c_j), AvgSimC aims to measure the averaged similarity between the two words all over the topics:

$$AvgSimC = \sum_{z,z' \in K} \Pr(z|w_i, c_i) \Pr(z'|w_j, c_j) S(\boldsymbol{v}_{w_i}^z, \boldsymbol{v}_{w_j}^{z'}) \tag{11}$$

where $\boldsymbol{v}_w^z$ is the embedding of word w under its topic z by concatenating word and topic embeddings $\boldsymbol{v}_w^z = \boldsymbol{v}_w \oplus \boldsymbol{\tau}_z$. $S(\boldsymbol{v}_{w_i}^z, \boldsymbol{v}_{w_j}^{z'})$ is the cosine similarity between $\boldsymbol{v}_{w_i}^z$ and $\boldsymbol{v}_{w_j}^{z'}$.

`MaxSimC` selects the corresponding topical word embedding $\boldsymbol{v}_w^z$ of the most probable topic z inffered using w in context c as the contextual word embedding, defined as

$$MaxSimc = S(\boldsymbol{v}_{w_i}^z, \boldsymbol{v}_{w_j}^{z'}) \tag{12}$$

where

$z = \arg\max_z \Pr(z|w_i, c_i)$, $z' = \arg\max_z \Pr(z|w_j, c_j)$.

We consider the two baselines Skip-Gram and `TWE`. Skip-Gram is a well-known single prototype model and `TWE` is the state-of-the-art multi-prototype model. We use all the default settings in these two model to train the 20NewsGroup corpus.

Table 2. Spearman correlation $\rho \times 100$ of contextual word similarity on the SCWS dataset.

Model	$\rho \times 100$	
Skip-Gram	51.1	
`LTSG`-word	53.4	
	`AvgSimC`	`MaxSimC`
`TWE`	52.0	49.2
`LTSG`	**54.2**	**54.1**

From Table 2, we can see that `LTSG` achieves better performance compared to the two competitive baseline. It shows that topic model can actually help improving polysemous-word model, including word embeddings and topic embeddings. The meaning of a word is certain by giving its specify context so that `MaxSimC` is more relative to real application. Then LTSG model achieves more improvement in `MaxSimC` than `AvgSimC` compared to `TWE`, which tells that LTSG could perform better in telling a word meaning in specify context.

4.2 Text Classification

In this sub-section, we investigate the effectiveness of `LTSG` for document modeling using multi-class text classification. The 20NewsGroup corpus has been divided into training set and test set with ratio 60% to 40% for each category. We calculate macro-averaging precision, recall and F1-score to measure the performance of `LTSG`.

We learn word and topic embeddings on the training set and then model document embeddings for both training set and testing set. Afterwards, we consider document embeddings as document features and train a linear classifier using Liblinear [8]. We use $\boldsymbol{v}_m$, $\boldsymbol{\tau}_k$, $\boldsymbol{v}_w$ to represent document embeddings, topic embeddings, word embeddings, respectively, and model documents on both topic-based and embedding-based methods as shown below.

Table 3. Evaluation results of multi-class text classification.

Model	Accuracy	Precision	Recall	F1-score
BOW	79.7	79.5	79.0	79.2
`LDA`	72.2	70.8	70.7	70.7
Skip-Gram	75.4	75.1	74.7	74.9
`TWE`	81.5	81.2	80.6	80.9
LTSG-theta	74.1	73.1	72.7	72.9
LTSG-topic	74.8	74.0	73.3	73.7
LTSG-word	81.4	81.0	80.4	80.7
LTSG	**82.7**	**82.5**	**81.7**	**82.1**

Table 4. Top words of some topics from `LTSG` and `LDA` on 20NewsGroup for $K = 80$.

`LTSG`	`LDA`	`LTSG`	`LDA`	`LTSG`	`LDA`	`LTSG`	`LDA`
image	image	jet	printer	stimulation	doctor	anonymous	list
jpeg	files	ink	good	diseases	disease	faq	mail
gif	color	laser	print	disease	coupons	send	information
format	gif	printers	font	toxin	treatment	ftp	internet
files	jpeg	deskjet	graeme	icts	pain	mailing	send
file	file	ssa	laser	newsletter	medical	server	posting
convert	format	printer	type	staffed	day	mail	email
color	bit	noticeable	quality	volume	microorganisms	alt	group
formats	images	canon	printers	health	medicine	archive	news
images	quality	output	deskjet	aids	body	email	nonymous
−75.66	−88.76	−91.53	−119.28	−66.91	−100.39	−78.23	−95.47

- **LTSG-theta.** Document-topic distribution $\boldsymbol{\theta}_m$ estimated by Eq. (7).
- **LTSG-topic.** $\boldsymbol{v}_m = \sum_{k=1}^{K} \theta_{m,k} \cdot \boldsymbol{\tau}_k$.
- **LTSG-word.** $\boldsymbol{v}_m = (1/N_m) \sum_{n=1}^{N_m} \boldsymbol{v}_{w_{m,n}}$.
- **LTSG.** $\boldsymbol{v}_m = (1/N_m) \sum_{n=1}^{N_m} \boldsymbol{v}_{w_{m,n}}^{z_{m,n}}$, where contextual word is simply constructed by $\boldsymbol{v}_{w_{m,n}}^{z_{m,n}} = \boldsymbol{v}_{w_{m,n}} \oplus \boldsymbol{\tau}_{z_{m,n}}$.

Result Analysis. We consider the following baselines, bag-of-word (BOW) model, `LDA`, Skip-Gram and `TWE`. The BOW model represents each document as a bag of words and use TFIDF as the weighting measure. For the TFIDF model, we select top 50,000 words as features according to TFIDF score. `LDA` represents each document as its inferred topic distribution. In Skip-Gram, we build the embedding vector of a document by simply averaging over all word embeddings in the document. The experimental results are shown in Table 3.

From Table 3, we can see that, for topic modeling, LTSG-theta and LTSG-topic perform better than `LDA` slightly. For word embeddings, LTSG-word

significantly outperforms Skip-Gram. For topic embeddings using for multi-prototype word embeddings, LTSG also outperforms state-of-the-art baseline TWE. This verifies that topic modeling, word embeddings and topic embeddings can indeed impact each other in LTSG, which lead to the best result over all the other baselines.

4.3 Topic Coherence

In this section, we evaluate the topics generated by LTSG on both quantitative and qualitative analysis. Here we follow the same corpus and parameters setting in Sect. 4.2 for LSTG model.

Quantitative Analysis. Although perplexity (held-out likehood) has been widely used to evaluate topic models, [3] found that perplexity can be hardly to reflect the semantic coherence of individual topics. Topic Coherence metric [17] was found to produce higher correlation with human judgments in assessing topic quality, which has become popular to evaluate topic models [1,4]. A higher topic coherence score indicates a more coherent topic.

We compute the score of the top 10 words for each topic. We present the score for some of topics in the last line of Table 4. By averaging the score of the total 80 topics, LTSG gets -92.23 compared with -108.72 of LDA. We can conclude that LTSG performs better than LDA in finding higher quality topics.

Qualitative Analysis. Table 4 shows top 10 words of topics from LTSG and LDA model on 20NewsGroup. The words in this two models are ranked based on the probability distribution φ for each topic. As shown, LTSG is able to capture more concrete topics compared with general topics in LDA. For the topic about "image", LTSG shows about image conversion on different format, while LDA shows the image quality of different format. In topic "printer", LTSG emphasizes the different technique of printer in detail and LDA generally focus on "good quality" of printing.

5 Releated Work

Recently, researches on cooperating topic models and vector representations have made great advances in NLP community. [24] proposed a Markov Random Field regularized LDA model (MRF-LDA) which encourages similar words to share the same topic for learning more coherent topics. [6] proposed Gaussian LDA to use pre-trained word embeddings in Gibbs sampler based on multivariate Gaussian distributions. LFLDA [18] is modeled as a mixture of the conventional categorical distribution and an embedding link function. These works have given the faith that vector representations are capable of helping improving topic models. On the contrary, vector representations, especially topic embeddings, have been promoted for modeling documents or polysemy with great help of topic models.

For examples, [14] used topic model to globally cluster the words into different topics according to their context for learning better multi-prototype word embeddings. [13] proposed generative topic embedding (TopicVec) model that replaces categorical distribution in LDA with embedding link function. However, these models do not show close interactions among topic models, word embeddings and topic embeddings. Besides, these researches lack of investigation on the influence of topic model on word embeddings.

6 Conclusion and Future Work

In this paper, we propose a basic model Latent Topical Skip-Gram (`LTSG`) which shows that `LDA` and Skip-Gram can mutually help improve performance on different task. The experimental results show that `LTSG` achieves the competitive results compaired with the state-of-art models.

We consider the following future research directions: (I) We will investigate non-parametric topic models [22] and parallel topic models [15] to set parameters automatically and accelerate training using multi threading for large-scale data. (II) We will construct a package which can be convenient to extend with other topic models and word embeddings models to our framework by using the interfaces. (III) We will deal with unseen words in new documents like Gaussian `LDA` [6].

Acknowledgments. We thank all reviewers for their valuable comments and feedback that greatly improved our paper. Zhuo thanks the National Key Research and Development Program of China (2016YFB0201900), National Natural Science Foundation of China (U1611262), Guangdong Natural Science Funds for Distinguished Young Scholar (2017A030306028), Pearl River Science and Technology New Star of Guangzhou, and Guangdong Province Key Laboratory of Big Data Analysis and Processing for the support of this research.

References

1. Arora, S., et al.: A practical algorithm for topic modeling with provable guarantees. In: ICML, pp. 280–288 (2013)
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. JMLR **3**, 993–1022 (2003)
3. Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J.L., Blei, D.M.: Reading tea leaves: how humans interpret topic models. In: NIPS, pp. 288–296 (2009)
4. Chen, Z., Liu, B.: Topic modeling using topics from many domains, lifelong learning and big data. In: ICML, pp. 703–711 (2014)
5. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.P.: Natural language processing (almost) from scratch. JMLR **12**, 2493–2537 (2011)
6. Das, R., Zaheer, M., Dyer, C.: Gaussian LDA for topic models with word embeddings. In: ACL, pp. 795–804 (2015)
7. Du, J., Jiang, J., Song, D., Liao, L.: Topic modeling with document relative similarities. In: IJCAI 2015, pp. 3469–3475 (2015)

8. Fan, R., Chang, K., Hsieh, C., Wang, X., Lin, C.: LIBLINEAR: a library for large linear classification. JMLR **9**, 1871–1874 (2008)
9. Griffiths, T.L., Steyvers, M.: Finding scientific topics. Proc. Nat. Acad. Sci. **101**(Suppl. 1), 5228–5235 (2004)
10. Hofmann, T.: Probabilistic latent semantic indexing. In: SIGIR 1999, pp. 50–57 (1999)
11. Huang, E.H., Socher, R., Manning, C.D., Ng, A.Y.: Improving word representations via global context and multiple word prototypes. In: ACL, pp. 873–882 (2012)
12. Jiang, D., Shi, L., Lian, R., Wu, H.: Latent topic embedding. In: COLING, pp. 2689–2698 (2016)
13. Li, S., Chua, T., Zhu, J., Miao, C.: Generative topic embedding: a continuous representation of documents. In: ACL (2016)
14. Liu, Y., Liu, Z., Chua, T., Sun, M.: Topical word embeddings. In: AAAI, pp. 2418–2424 (2015)
15. Liu, Z., Zhang, Y., Chang, E.Y., Sun, M.: PLDA+: parallel latent dirichlet allocation with data placement and pipeline processing. ACM TIST **2**(3), 26 (2011)
16. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS, pp. 3111–3119 (2013)
17. Mimno, D.M., Wallach, H.M., Talley, E.M., Leenders, M., McCallum, A.: Optimizing semantic coherence in topic models. In: EMNLP, pp. 262–272 (2011)
18. Nguyen, D.Q., Billingsley, R., Du, L., Johnson, M.: Improving topic models with latent feature word representations. TACL **3**, 299–313 (2015)
19. Phan, X.H., Nguyen, C., Le, D., Nguyen, M.L., Horiguchi, S., Ha, Q.: A hidden topic-based framework toward building applications with short web documents. IEEE Trans. Knowl. Data Eng. **23**(7), 961–976 (2011)
20. Reisinger, J., Mooney, R.J.: Multi-prototype vector-space models of word meaning. In: NAACL, pp. 109–117 (2010)
21. Socher, R., Bauer, J., Manning, C.D., Ng, A.Y.: Parsing with compositional vector grammars. In: ACL, pp. 455–465 (2013)
22. Teh, Y.W., Jordan, M.I., Beal, M.J., Blei, D.M.: Hierarchical dirichlet processes. J. Am. Stat. Assoc. **101**(476), 1566–1581 (2006)
23. Turian, J.P., Ratinov, L., Bengio, Y.: Word representations: a simple and general method for semi-supervised learning. In: ACL 2010, pp. 384–394 (2010)
24. Xie, P., Yang, D., Xing, E.P.: Incorporating word correlation knowledge into topic modeling. In: NAACL, pp. 725–734 (2015)