A Bidirectional Hierarchical Skip-Gram Model for Text Topic Embedding

Suncong Zheng, Hongyun Bao, Jiaming Xu, Yuexing Hao, Zhenyu Qi, Hongwei Hao
Institute of Automation, Chinese Academy of Sciences, 100190, Beijing, P.R. China
{suncong.zheng, hongyun.bao, jiaming.xu, haoyuexing2014, zhenyu.qi, hongwei.hao}@ia.ac.cn

***Abstract*—Taking advantage of the large scale corpus on the web to effectively and efficiently mine the topics within texts is an essential problem in the era of big data. We focus on the problem of learning text topic embedding in an unsupervised manner, which enjoys the properties of efficiency and scalability. Text topic embedding represents words and documents in a semantic topic space, in which the words and documents with similar topic will be embedded close to each other. When compared with conventional topic models, which implicitly capture the document-level word co-occurrence patterns, text topic embedding alleviates the data sparsity problem and captures the semantic relevance between different words and documents. To model text topic embedding, we propose a Bidirectional Hierarchical Skip-Gram model (BHSG) based on skip-gram model. BHSG includes two components: semantic generation module to learn semantic relevance between texts and topic enhance module to produce the text topic embedding based on text embedding learned in the former module. We evaluated our method on two kinds of topic-related tasks: text classification and information retrieval. The experimental results on four public datasets and one dataset we provide all demonstrate that our proposed method can achieve a better performance.**

I. Introduction

Recent years have witnessed a dramatic increase of text data on the web. How to take full advantage of the large scale corpus to effectively mine the topics within texts is an essential problem in the era of big data. Besides, topic mining is an essential task for a wide range of content analysis problem, such as text classification [26], answer extraction [1] and emerging topic detecting [25].

Conventional topic modeling techniques [3], [13], [14] consider each document as a separate bag-of-words composition, which only capture the document-level word co-occurrence patterns and suffer from the data sparsity. They also cannot capture the semantic relevance between different words and documents. Fortunately, text embedding [20], [24], [29] can effectively address this problem through representing the words or documents in latent space, in which the texts with the similar semantic will be embedded close to each other. Based on the idea of text embedding methods and the property of topic mining problem, in this paper, we propose a text topic embedding method, which can be more conductive to topic-related tasks, such as text classification and information retrieval. Besides, our text topic embedding model is implemented in an unsupervised manner, which enjoys the properties of simplicity and scalability. Therefore, it can be fit for the tasks with large scale textual data.

TABLE I
Examples of Model Motivation.

Example 1
1). *"The CEO of apple will attend the conference"*
2). *"Tim Cook will go to the forum"*
Example 2
1). *"Steve Jobs co-founded Apple in 1976 to sell personal computer"*
2). *"Apple is obtained from medium-sized tree belonging to the Rosaceae family"*
Example 3
Description: *The capital of the United States in the District of Columbia and a tourist mecca; George Washington commissioned Charles L'Enfant to lay out the city in 1791.* **Topic:** *Washington D.C*

Method Motivation: In order to meet the above requirements, we start with the following aspects: 1) The semantic relationship between words is important to the sentence/document representation. As *Example 1* in Table I shows that few words in the given two sentences overlap, but we still can judge the high similarity of the sentences based on the semantic relationship between words. 2) The context information can be beneficial to enhance the topic information of words. Both of the sentences, in *Example 2*, contain the word of "apple", but the meanings of "apple" in these two sentences are quite different. If we take the context information into consideration, the "apple" in sentence *1* tends to talk about an IT company while the "apple" in sentence *2* tends to talk about a kind of fruit. 3) The words within a sentence/document contribute to the main semantic of the given sentence/document in some degree. *Example 3* is the description of "Washington D.C" in WordNet[1]. Although the description text does not contain the phrase of "Washington D.C." explicitly, the words, such as "George Washington", "United States" and "capital", can be linked together by the topic term "Washington D.C.". Therefore, mining the words' similar semantic features can enhance the topic information of the given text.

Model Description: Based on the above motivations, we

[1]http://wordnet.princeton.edu/

978-1-5090-0620-5/16/$31.00 ©2016 IEEE

propose Bidirectional Hierarchical Skip-Gram (BHSG) model to settle the text topic embedding.

The skip-gram model tries to predict the surrounding words within a certain distance based on the current one. Words occurring in similar contexts tend to have similar meanings. Therefore it can capture the semantic relationship between the words. If we consider the whole sentence/document as a special word, called global context word, and apply the skip-gram to capture the semantic relationship between the words and the global context word, the words can contain the global context information and the global context word capture the basic topic information of the given text. Based on the above analysis, we adopt a hierarchical skip-gram model to learn the semantic relationship between different words and enhance the topic relationship between words and document. The first level skip-gram model is to link the sentence/document and the words within it. The second level is to capture the semantics and syntax of words as Mikolov et al. [29] done. Since the semantic relationship between different texts is not enough for mining the deep topic information, we design a topic enhanced module, which is to mine words' common semantic features, to produce the text topic embedding based on the text embedding learned in hierarchical skip-gram model. The topic enhanced module is based on the idea of Word Mover's Distance (WMD) [21] that the distance between text topic embedding and the topic-related words should be minimum. When given a word, the measurement of topic correlation is based on the similarity between current global context word and the given word.

In order to demonstrate the effectiveness of the text topic embedding learned by our BHSG model, we set up two important topic-related tasks: text classification and information retrieval. For text classification task, we apply four public datasets to validate the model. Besides, we also create a information retrieval dataset to validate the model. Summary of our main contributions:

- We proposed a text topic embedding method based on the idea of text embedding methods and the property of topic mining problem. Hence it can be more conductive to topic-related tasks, such as text classification and information retrieval.
- Bidirectional Hierarchical Skip-Gram (BHSG) model we proposed is a high-efficiency and unsupervised method, which can be suitable for the problems with large scale textual data.
- The experimental results on five datasets and two kinds of topic-related tasks all demonstrate the effectiveness of our proposed methods.

The rest of this paper is organized as follows: We first introduce the related work in Section II. In Section III we present the details of BHSG model, then share the implementation techniques. Section IV introduces the experimental setting and Section V presents the results of empirical experiments. We analyze the different methods, which related to the tasks, in Section VI. At last, we conclude in Section VII.

II. Related works

In this paper, the problem is related to text representation for topic mining and the model we propose is related to skip-gram model.

Topic mining has been a key problem for text analysis, such as text classification [26] and answer extraction [1] . In order to mine the topics within text, the first thing we should do is to represent texts. One of the most common and classical method is that each document is represented as a "bag-of-words" [10]. However, the bag-of-words (BoW) has many disadvantages. It cannot discriminate semantic relevance between different words and suffers from data sparsity and high dimensionality. Topic models [3], [13], [14] are based upon the idea that documents are mixtures of topics, where a topic is a probability distribution over words. It can make up the BoW's shortages of high dimension and can also handle the problem about words dependency effectively. But topic model implicitly captures the word co-occurrence patterns in document-level and lost the information of word order in sentence.

Text embedding provides a new manner to represent text's semantic information, which is to represent the text in a semantic topic space and the text have the similar semantic will be embedded close to each other. These methods for text embedding can be divided into two main categories: the unsupervised text embeddings [20], [24], [33] and the supervised text embeddings [19], [6], [12], [17], [32], [23].

The most popular unsupervised text embedding method is the word2vec proposed by Mikolov et al. [29], which is to learn the semantic representation of words. Le and Mikolov [24] further extended this idea and proposed the Paragraph Vectors, which can embed arbitrary pieces of text, e.g., sentences and documents. The basic idea of Paragraph Vectors is to represent each document by a dense vector which is trained to predict words in the document. Different from the simple manners of Paragraph Vectors, Kiros et al. [20] proposed an encoder-decoder model, called "Skip-Thought", to learn sentence embedding. The basic idea of "Skip-Thought" is to use the embedding of current sentence to reconstruct the surrounding sentences. The training datasets they used must be the continuity of text from books not a single sentence. The existing unsupervised text embedding methods for sentences (documents), such as Paragraph Vector [24] and Skip-thought [20], aim to embed sentences (documents) into a semantic space so that the embedded text can be suitable for any Natural Language Processing (NLP) tasks. However, due to the complexity of sentence syntax and the diversity of sentence semantic, the universal text embedding can reflect the basic semantic but cannot do the best in one specific task.

Different from the generality of unsupervised text embedding, supervised text embedding only suits for respective task. It can learn high quality sentence representations. In recent years, deep learning models have been successfully applied to composite sentence/paragraph semantic representations based on word embeddings. The most common neural-network-

based models are Recurrent Neural Networks (RNN) [17], [28], Recursive Neural Networks (RecNN) [32], [31], [16] and Convolutional Neural Network (CNN) [19], [18], [7].

Skip-gram model The model of Bidirectional Hierarchical Skip-Gram (BHSG) we proposed, in this paper, is based on skip-gram model [27], which has shown the effectiveness to capture the semantics and syntax of words. Skip-gram model uses the target word to predict each individual context word in a local window, so the words occurring in similar contexts tend to have similar meanings. Neural probabilistic language models [2] and recurrent neural network based language models have the same idea of skip-gram. Unfortunately, the training of these models is quite time consuming. Skip-gram model use a simple single-layer architecture based on inner product between two word vectors and a negative sampling strategy to approximate the softmax, so it can be quite efficient and can improve the accuracy.

III. The Model of Bidirectional Hierarchical Skip-Gram

The task of text topic modeling, in this paper, can be described as: given a sentence or document d, we need to produce a distributed representation $\vec{t}$ of the given text, so that the texts have the similar semantics or topics will be embedded close to each other. We propose Bidirectional Hierarchical Skip-Gram to achieve the goal. In the following sections, we firstly present the architecture of our model shown in Figure 1 and then detail each component of the model. After that, we introduce objective function and parameter inference. At last, we share the implementation techniques.

A. The Architecture of BHSG

The Architecture of BHSG is shown in Figure 1 and it can be cast into two modules: semantic generation module and topic enhanced module. Semantic generation module is learning semantic relevance between texts and can be treated as the path $\{ d \rightarrow \vec{d} \rightarrow \vec{w} \rightarrow \vec{c} \}$ in Figure 1. Topic enhanced module is to enhance the text topic embedding based on the text embedding learned in semantic generation module and can be represented as: $\{ (\vec{d}, \vec{w}) \rightarrow \vec{t} \}$. Where d denotes a sentence and its context embedding is represented as $\vec{d} \in \mathbb{R}^k$. w_i is the i-th word in text d and it has two kinds of word embeddings: input vector $\vec{w}_i \in \mathbb{R}^k$ and output vector $\vec{c}_i \in \mathbb{R}^k$. Besides, the output of the model is $\vec{t} \in \mathbb{R}^k$, which is the topic embedding of text d. The entries in the vectors are treated as parameters to be learned and k is embedding dimensionality. In what follows, we describe these modules in detail.

B. Semantic Generation Module

The semantic generation module aims to catch the semantic relationship between words and enhance the topic information of words. Besides, it also captures the basic topic information of the given text, which can be used to measure the importance of the words within the text. We adopt a hierarchical skip-gram structure to meet the above requirements.

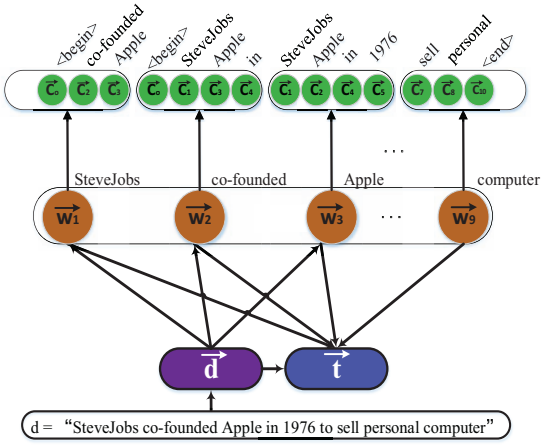

Fig. 1. The architecture of Bidirectional Hierarchical Skip-Gram Model.

The first layer of skip-gram is to capture the document-level words co-occurrence patterns as topic models do, so that the words within text can contain the topic distribution information and we can also get the basic topic information $\vec{d}$ of the given text. The objective of the first layer skip-gram model is to maximize the average log probability:

$$L_1 = \frac{1}{|W|} \cdot \sum_{d_n \in D} \sum_{w_i^n \in d_n} logp(w_i^n | d_n), \tag{1}$$

where $|W|$ is the words number of the given corpus D. The probability of $p(w_i^n | d_n)$ can be accurately computed by using the softmax function as:

$$p(w_i^n | d_n) = \frac{exp(\vec{w_i^n} \cdot \vec{d_n})}{\sum_{w_j^n \in V} exp(\vec{w_j^n} \cdot \vec{d_n})}, \tag{2}$$

where V is the vocabulary. However, to compute the denominator of Formula 2 is time-cost and impractical. Fortunately, Mikolov et al. [29] propose an negative sampling manner which can approximate log probability of the softmax well. Therefore, the objective function L_1 of first layer skip-gram model can be approximated as :

$$\begin{aligned} L_1 \approx \frac{1}{|W|} \cdot \sum_{d_n \in D} \sum_{w_i^n \in d_n} \{ & log\ \sigma(\vec{w_i^n} \cdot \vec{d_n}) \\ & - \sum_{m=1}^{M} \mathbb{E}_{w_m^{n'} \backsim P_n(w)} log\ \sigma(\vec{w_m^{n'}} \cdot \vec{d_n}) \} \end{aligned}, \tag{3}$$

where $\sigma(x) = 1/(1 + exp(-x))$ and M is the number of negative sampless. Different from word2vec [29], the negative word $w_m^{n'}$ here is the word that not appears in the given text d_n. $\mathbb{E}_{w_m^{n'} \backsim P_n(w)}$ means that the negative word $w_m^{n'}$ is sampled based on the word frequency distribution.

The second layer of skip-gram is to capture the semantics and syntax of words as Mikolov et al. [29] done and the

objective function L_2 of second layer skip-gram model is to maximize the log likelihood function:

$$L_2 \approx \frac{1}{|W|} \cdot \sum_{w_i^n \in d_n} \sum_{\substack{j=i-l \\ j \neq i}}^{i+l} \{log\ \sigma(\vec{w_i^n} \cdot \vec{c_j^n}) - \sum_{m=1}^{M} \mathbb{E}_{w_m^{i'} \backsim P_n(w)} log\ \sigma(\vec{c_j^n} \cdot \vec{w}_m^{\ i'})\}, \quad (4)$$

where l is the window size and $w_m^{i'}$ is the negative sample of word w_i^n.

C. Topic Enhanced Module

The semantic generation module have caught the semantic and topic-related relationship between words. Hence, mining the words' common semantic features based on semantic generation module can further enhance the topic information of given text. Kusner et al. [21] present a novel distance function named Word Mover's Distance (WMD) to measure the texts' similarity. WMD [21] defines that the similarity between document A and document B is the minimum amount of distance that the embedded words of A need to reach the embedded words of B.

We assume that the document A is the given text which need to be processed and document B can be treated as the topic embedding of document A. Therefore, the topic enhanced module is based on the objective function that the distance between text topic embedding and the topic-related words in given text should be minimum. Considering the issue that different words in text have different contribution to the text topic, we add a weight of topic correlation to each word w based on the current global context embedding $\vec{d}$ and the word embedding $\vec{w}$ produced in the semantic generation module. Therefore, the objective function of topic enhanced module is to minimize the weighted distance between document d and the topic embedding $\vec{t}$ as Formula 5 shows:

$$E = \sum_{d_n \in D} \sum_{w_i^n \in d_n} \|\vec{t_n} - f(\vec{d_n}, \vec{w_i^n}) \cdot \vec{w_i^n}\|^2, \quad (5)$$

where $\vec{t_n}$ is the topic embedding of text n and the $f(\vec{d_n}, \vec{w_i^n})$ is the topic correlation weight of word w_i^n in text d_n. Because $\vec{d_n}$ contains the basic topic information of the given text d_n and the words embedding w^n also capture the semantic relationship and topic-related information, the topic correlation weight of word w_i^n can be compute as follows:

$$f(\vec{d_n}, \vec{w_i^n}) = \frac{\|\vec{d_n} - \vec{w_i^n}\|^2}{\sum_{w_j^n \in d_n} \|\vec{d_n} - \vec{w_j^n}\|^2}. \quad (6)$$

D. Training and Inference

After describing the semantic generation module and topic enhanced module, it is natural to combine them into a global framework which is to obtain the topic embeddings. Hence, the global objective function we want to optimize is to minimize the combined function showed in Formula 7:

$$J = E - L_1 - L_2. \quad (7)$$

We call this global framework as Bidirectional Hierarchical Skip-Gram and denote it as BHSG.

The parameters of the model to be trained are concluded as: $\Theta = \{\vec{d}, \vec{w}, \vec{c}, \vec{t}\}$. The model is optimized using stochastic gradient descent and the gradient is obtained via backpropagation. When processing a new text, one needs to perform the inference by gradient descent to get the topic embedding.

E. Implementation

Since natural language has a very strong Zipfian distribution [5], the frequent words contain less information and may also hurt the final results. A simple solution is to subsample (discard) words from the given text based on its unigram frequency of occurrence [29].

Although, semantic generation module and topic enhanced module are combined into a global framework, topic enhanced module strongly relies on semantic generation module. In practice we found that if we firstly pre-train the semantic generation module then train the global framework simultaneously, the results could be better. The reason is that the pre-train procedure can make BHSG have suitable initial parameters, which is benefit for getting stabilized topic embedding.

IV. Experimental Setup

In this paper, we propose a Bidirectional Hierarchy Skip-Gram (BHSG) to represent the text topic embedding, so that it can be more conducive to topic-related tasks. In order to demonstrate the effectiveness of the text topic embedding learned by our BHSG model, we set up two important topic-related tasks: text classification and information retrieval. One is the classification task and the other one is the rank task. They all require the text to be represented as a meaningful and effective vector.

A. Text Classification

Text classification is the task of automatically classifying a set of text documents into different categories from a predefined label set. We use accuracy as our metric to evaluate the performances of different methods on the task. We first describe each dataset used on the task and then present a set of classical baselines for comparison.

1) Datasets: We validate our model on four widely used text classification datasets[2]: 20newsgroups, Reuters-21578, Cade, and Webkb.

- 20newsgroups (20NEWS): It is a news article set and is classified into 20 different categories. In this paper, the "Bydate" version of 20NEWS is used, which already had a standard train/test split.
- Reuters-21578 (R8): The documents in this collection appeared on the Reuters newswire in 1987 and were

[2]http://web.ist.utl.pt/acardoso/datasets/

manually classified. The version we used is processed by Ana [4] and it has 8 different categories in total.
- Cade : The documents in the Cade collection correspond to a subset of web pages extracted from the CADWeb Directory and are classified by human experts.
- Webkb: The Webkb collection (also called 4 Universities Data Set) contains webpages collected from computer science departments of various universities in 1997.

These datasets we used have been preprocessed by Ana [4] and Table II shows relevant statistics for each of these datasets.

2) Baselines: The baselines used here include the widely used representation learning algorithms for text data and the deep learning based model.

- Bag-of-words (BoW) [30]: Each text is represented with a $|V|$-dimensional vector, in which the weight of each dimension is calculated by TF-IDF (term frequency-inverse document frequency).
- Latent Semantic Indexing (LSI) [8]: Latent semantic indexing is an information retrieval technique based on the spectral analysis of the term-document matrix. Ana [4] also uses it as an effective baseline on these four datasets.
- Latent Dirichlet Allocation (LDA) [3]: LDA is a celebrated generative model for text documents and it can learns representations for documents as distributions over word topics.We use the default parameters provided by Matlab Topic Modeling Toolbox[3].
- Average Embedding (VecAvg) [32]: Averaging the word embedding is also an effective and widely used method to represent the given text. We employ the latest version of word2vec[4] and also use the default parameters to produce the word embeddings.
- Paragraph Vector (PVDBOW and PVDM) [24], an unsupervised learning algorithm that learns vector representations for variable length pieces of texts. It has two kinds of framework: distributed memory model of paragraph vectors (PVDM) and the distributed bag of words version (PVDBOW).
- We also select the convolutional neural networks [19] for comparison. CNN [19] improves upon the state of the art on 4 out of 7 classification tasks, which include the topic classification task. Different from the above methods, this is a supervised method. We also set two kinds of CNN for comparison: CNN-random whose input words embeddings are randomly initialized and CNN-w2v whose input words embeddings are provided by word2vec.

For each unsupervised text representation method we use two established classification algorithms: Logistics Regression (LR) [15] and Support Vector Machine (SVM) [11] to classify the text based on the representation.

[3]http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm
[4]https://code.google.com/p/word2vec/

TABLE II
STATISTICS OF THE TEXT CLASSIFICATION DATASET. $|L|$ IS THE AVERAGE LENGTH OF DOCUMENT IN THE DATASET AND $|y|$ IS THE NUMBER OF CATEGORIES.

Name	Train	Test	Total	$\|L\|$	$\|y\|$
20NEWS	11293	7528	18821	266	20
R8	5485	2189	7674	106	8
Cade	27322	13661	40983	116	12
Webkb	2803	1396	4199	134	4

B. Information Retrieval

Information Retrieval (IR) is to obtain information resources relevant to the query from a collection of information resources.

1) Datasets: We create a dataset for the information retrieval task. Firstly, we obtain 4,900 questions from the college entrance examination on history subjects and then collect the historical materials that can be used for answering the questions. The historical materials are divided into 447 parts by historian based on their topics. Each part of the historical materials can be treated as a historical document. Three volunteers, who are good at the historical knowledge, are invited to answer the questions and they mark the relevant historical documents of the given questions. The relevant historical documents contains the answer of the given question. Especially, the relevant historical documents of a given question are not single, sometimes a question corresponds to multiple relevant historical documents.

2) Baselines: The question is short text and the historical documents is the long text. To assess the performance of different text representation methods on the task, we firstly represent the question and documents as fixed length vectors, then use the cosine similarity to measure the relevancy. The text representation baselines used in this task are the same as the baselines in text classification: Bag-of-words (BoW)[30], Latent Dirichlet Allocation (LDA) [3] , Paragraph Vector (PVDBOW and PVDM) [24], and Average Embedding (VecAvg) [32]. Considering that the question is short text, we also add Biterm Topic Model (BTM) [34] which learns topics over short texts.

3) Metric: In this task we need to return a ranked sequence of documents, it is also desirable to consider the order in which the returned documents are presented. Hence, we use the mean Average Precision (mAP), which has shown the discrimination and stability. The definition of mAP is shown as Formula 8:

$$mAP@n = \frac{1}{|Q|}\sum_{j=1}^{|Q|}\frac{1}{n}\sum_{k=1}^{n} Precision(R_{jk}), \qquad (8)$$

where R_{jk} is the set of ranked retrieval results from the top results until you get to document d_k. Instead of setting the similar threshold, we return the top n results as the relevant historical documents when given the question.

TABLE III
THE ACCURACY ON THE TEXT CLASSIFICATION. THE BASELINES WE USED CAN BE DIVIDED INTO 4 CATEGORIES: BAG OF WORDS METHODS, TOPIC MODELS, TEXT EMBEDDING METHODS AND DEEP LEARNING METHODS.

Methods	20NEWS	R8	Webkb	Cade
BoW-LR	81.82	95.07	85.82	53.86
BoW-SVM	82.84	96.98	86.12	52.84
LSI	74.91	94.11	73.57	43.28
LDA-LR	67.94	92.14	72.56	42.12
LDA-SVM	68.41	92.20	72.92	41.24
VecAvg-LR	79.60	96.57	86.15	50.03
VecAvg-SVM	79.00	97.00	87.10	50.51
PVDBOW-LR	80.46	95.16	87.33	56.01
PVDBOW-SVM	75.15	94.00	86.37	56.14
PVDM-LR	79.01	93.60	79.48	53.86
PVDM-SVM	75.10	91.46	75.04	54.14
CNN-random	79.23	95.97	91.82	55.43
CNN-w2v	80.06	97.57	**92.33**	56.48
BHSG-LR	85.36	96.80	87.83	**59.05**
BHSG-SVM	**85.86**	**97.62**	88.06	**59.04**

V. RESULTS

A. Performance on Text Classification

Table III compares the performance of different methods on text classification task. When compared with the unsupervised representation methods, BHSG performs best on the text classification task. Perhaps surprisingly, the topic models, LSI and LDA, perform worst in all methods. Results of [9] also show that LDA models do not generally outperform BoW and LSI. There is no much different among the performances of BoW, VecAvg and PVDBOW on 20NEWS, R8 and Webkb. When tested on dataset Cade, PVDBOW shows the advantage. Besides, when compared with these baselines, the improvement of BHSG is higher on the dataset 20NEWS and Cade. Because BHSG and PVDBOW both are based on the skip-gram structure which have shown the effectiveness to capture the semantics and syntax of words. If the training corpus is larger, the skip-gram model can be better to capture the semantic relationship between words, which can benefit for final representation. The size of Cade and 20NEWS are larger than R8 and Webkb.

Apart from the unsupervised text representation methods, we also compare BHSG with the deep learning model: CNN [19]. On datasets 20NEWS and Cade, BHSG achieves the 4% improvements. But on the dataset R8, the improvement of BHSG is little. Notably, the performance of BHSG is worse than the CNNs when tested on the dataset of Webkb. The explanation for the above results is that BHSG is a kind of unsupervised text representation and the scale of training corpus can affect the semantic representation. On the contrary, CNN is a supervised model and CNN-w2v even uses the word embedding learned from other large scale corpus. Therefore, the scale of dataset has less effect on CNN than BHSG. The sizes of 20NEWS and Cade are large enough for BHSG to capture the semantic information, hence it can outperform the CNNs. But the sizes of R8 and Webkb are small, which cannot show the advantage of BHSG when compared with CNN.

TABLE VI
THE ACCURACY OF INTERMEDIATE VARIABLES ON TEXT CLASSIFICATION.

Methods	20NEWS	R8	Webkb	Cade
BHSG-w-LR	84.20	96.61	87.24	57.00
BHSG-d-LR	81.85	94.97	85.86	55.58
BHSG-t-LR	**85.36**	**96.80**	**87.83**	**59.05**
BHSG-w-SVM	84.42	97.3	87.30	56.05
BHSG-d-SVM	75.53	93.06	82.12	55.84
BHSG-t-SVM	**85.86**	**97.62**	**88.06**	**59.04**

B. Performance on Information Retrieval

The performance of topic embeddings and other text representation methods on the information retrieval task are shown in Table IV. As we can see that BHSG achieves the highest mean average precision. BoW outperform the more recent approaches PVDBOW and PVDM. Because the information retrieval task here is to obtain the relevant historical documents that contain the answer when given a question. The key-word information in question and historical documents determine the relevance of question and historical documents. PVDBOW and PVDM try to capture the general information of text, yet BoW use the TF-IDF to measure the importance of words within text. Hence BoW can has a better result. The results of PVDM is poor. The reason is that PVDM [22] cannot work well when the dataset is small.

The topic models, LDA and BTM, still perform badly. Because the question is short text and the document is long text. If we use LDA to represent the question, it will suffer from the severe data sparsity problem. If we use btm to represent the document, it can work well on the long text. Therefore, the LDA and BTM cannot work well on this task.

VI. ANALYSIS

A. The Intermediate Variables of BHSG

To model the topic embedding of the given text, we propose a BHSG model which can be cast into two modules: semantic generation module and topic enhanced module. In semantic generation module, we obtain a context embedding $\vec{d} \in \mathbb{R}^k$ and topic-enhanced word embeddings $\vec{w}_i \in \mathbb{R}^k$. In order to analyze the performance of these intermediate variables, we compare topic embedding $\vec{t} \in \mathbb{R}^k$ with context embedding $\vec{d}$ and topic-enhanced word embeddings $\vec{w}_i$ on the tasks of text classification and information retrieval. We directly use context embedding $\vec{d}$ produced in semantic generation module to represent the text embedding and call it BHSG-d. Besides, we also average the word embeddings $\vec{w}_i$ produced by BHSG to represent the text embedding and represent it as BHSG-w. The performance of intermediate variables on text classification task and on information retrieval task are shown in Table VI and Table V.

The topic embedding $\vec{t}$ outperforms all of the intermediate variables, when tested on these two topic-related tasks. Besides, the performance of BHSG-w can be better than BHSG-d on all datasets. The topic embedding $\vec{t}$ is produced based on context embedding $\vec{d}$ and the topic-enhanced word embeddings

TABLE IV
THE MEAN AVERAGE PRECISION OF DIFFERENT METHODS ON INFORMATION RETRIEVAL.

Methods	mAP@1	mAP@2	mAP@3	mAP@5	mAP@7	mAP@10
BoW	36.82	44.07	47.00	49.42	50.30	50.89
LDA	13.60	18.98	21.86	24.72	26.12	27.22
BTM	12.92	17.93	20.60	23.49	24.90	25.95
PVDM	16.78	20.86	22.59	24.22	25.06	25.72
PVDBOW	32.10	38.96	41.73	44.08	45.03	45.67
VecAvg	37.63	45.62	48.77	51.40	52.39	53.08
BHSG	**42.56**	**50.16**	**52.85**	**55.14**	**56.02**	**56.58**

TABLE V
THE MEAN AVERAGE PRECISION OF INTERMEDIATE VARIABLES ON INFORMATION RETRIEVAL.

Methods	mAP@1	mAP@2	mAP@3	mAP@5	mAP@7	mAP@10
BHSG-w	41.65	49.36	52.28	54.64	55.47	55.99
BHSG-d	32.10	38.96	41.73	44.08	45.03	45.67
BHSG-t	**42.56**	**50.16**	**52.85**	**55.14**	**56.02**	**56.58**

$\vec{w}_i$. We adopt the idea of WMD [21] to enhance the topic information. Hence, it can achieve a better result than BHSG-w and BHSG-d on the topic-related tasks. Different from the classical word embedding, the word vector learned in BHSG not only contains the information of syntax and semantic but also contains the document's topic information. While the context embedding $\vec{d}$ only contains the rough information of the given text. Therefore, BHSG-w can outperform BHSG-d on topic representation.

B. Parameter Sensitivity

BHSG has two important parameters: embedding size and the number of negative samples. If the embedding size is too small, the text topic embedding cannot capture the complex semantic information of the given text. If the embedding size is too large, it will affect the training speed. In order to analyze the effect of different embedding sizes, we change the embedding size from 50 to 1000 and test the corresponding embedding on the above four widely used text classification datasets. We fix the number of negative samples to 5 when analyze the embedding sizes. The influence of embedding size is shown in Figure 2. We can find that the accuracy curve on four datasets all rise with the increasing of embedding size and achieve stability at last. The bigger of the embedding size, the more semantic information it contains. When the embedding size is big enough, the influence of the size tend to stability. Therefore, considering both the running speed and accuracy, we suggest that the range of embedding size should be 200 to 600.

The computation of the softmax function over the vocabulary is time-cost and impractical. Hence, we consider the negative sampling manner to approximate softmax as Mikolov et al. [29] do. The number of negative samples can also affect the efficiency and effectiveness of the given task. We change the number of negative samples from 5 to 30 and fix the size of embedding to 200 when analyze the number of negative samples. The results, test on the text classification task, are shown in Figure 3. We find that the number of negative samples is not the more the better. When the number of negative samples is 5, it can also obtain a good result.

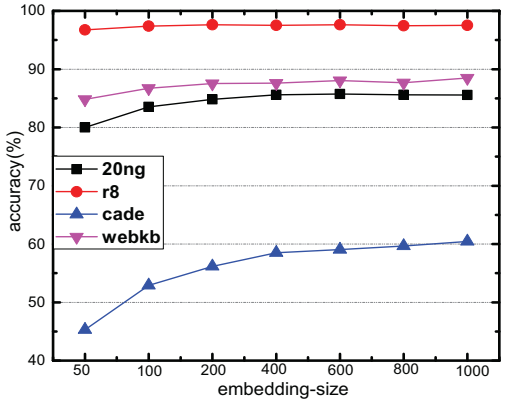

Fig. 2. Effect of embedding size on text classification

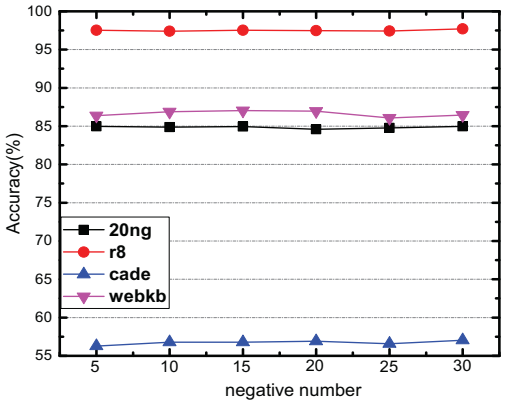

Fig. 3. Effect of negative sample number on text classification

VII. CONCLUSION

In this paper, we propose a Bidirectional Hierarchy Skip-Gram (BHSG) model to mine the topic information within

the given texts. Different from the universal text embedding methods, the text topic embedding learned by BHSG not only contains the semantic information of text but also enhances the topic information of text, which can be more conducive to topic-related tasks, such as text classification and information retrieval.

BHSG consists of two modules: semantic generation module, which is to learn semantic relevance between texts, and topic enhanced module, which is to enhance the text topic embedding based on the text embedding learned by semantic generation module. We validated our method for text topic embedding on two kinds of topic-related tasks: text classification and information retrieval. The experimental results all demonstrate the effectiveness of our proposed method. Our model is based on the negative sampling manner, which has high efficiency and it is also an unsupervised method. Hence BHSG is very suitable for the large scale data. In the future, we will extend BHSG to more topic-related tasks such as: keywords extraction and text summarization.

VIII. Acknowledgments

We appreciate the assistance from Jianwen Zhan (Microsoft Research Asia). This work is also supported by the National High Technology Research and Development Program of China (863 Program) (Grant No. 2015AA015402), the Hundred Talents Program of Chinese Academy of Sciences (No. Y3S4011D31) and National Natural Science Foundation (Grant No. 71402178).

References

[1] S. Abney, M. Collins, and A. Singhal. Answer extraction. In *Proceedings of the sixth conference on Applied natural language processing*, pages 296–301. Association for Computational Linguistics, 2000.

[2] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155, 2003.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[4] A. M. d. J. C. Cachopo. *Improving Methods for Single-label Text Categorization*. PhD thesis, Universidade Técnica de Lisboa, 2007.

[5] W. B. Cavnar, J. M. Trenkle, et al. N-gram-based text categorization. *Ann Arbor MI*, 48113(2):161–175, 1994.

[6] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder–decoder approaches. *Syntax, Semantics and Structure in Statistical Translation*, page 103, 2014.

[7] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011.

[8] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *JAsIs*, 41(6):391–407, 1990.

[9] P. V. Gehler, A. D. Holub, and M. Welling. The rate adapting poisson model for information retrieval and object recognition. In *Proceedings of the 23rd international conference on Machine learning*, pages 337–344. ACM, 2006.

[10] Z. S. Harris. Distributional structure. *Word*, 1954.

[11] M. A. Hearst, S. T. Dumais, E. Osman, J. Platt, and B. Scholkopf. Support vector machines. *Intelligent Systems and their Applications, IEEE*, 13(4):18–28, 1998.

[12] K. M. Hermann and P. Blunsom. Multilingual models for compositional distributed semantics. pages 58–68, 2014.

[13] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

[14] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

[15] D. W. Hosmer Jr and S. Lemeshow. *Applied logistic regression*. John Wiley & Sons, 2004.

[16] M. Iyyer, J. Boyd-Graber, L. Claudino, R. Socher, and H. Daumé III. A neural network for factoid question answering over paragraphs. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 633–644, 2014.

[17] N. Kalchbrenner and P. Blunsom. Recurrent continuous translation models. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1700–1709, 2013.

[18] N. Kalchbrenner, E. Grefenstette, and P. Blunsom. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 212–217. Association for Computational Linguistics.

[19] Y. Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[20] R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler. Skip-thought vectors. *arXiv preprint arXiv:1506.06726*, 2015.

[21] M. J. Kusner, E. Y. Sun, E. N. I. Kolkin, and W. EDU. From word embeddings to document distances. In *Proceedings of The 32st International Conference on Machine Learning*, 2015.

[22] S. Lai, K. Liu, L. Xu, and J. Zhao. How to generate a good word embedding? *arXiv preprint arXiv:1507.05523*, 2015.

[23] S. Lai, L. Xu, K. Liu, and J. Zhao. Recurrent convolutional neural networks for text classification. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[24] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *Proceedings of The 31st International Conference on Machine Learning*, pages 1188–1196, 2014.

[25] C. X. Lin, B. Zhao, Q. Mei, and J. Han. Pet: a statistical model for popular events tracking in social communities. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 929–938. ACM, 2010.

[26] L. M. Manevitz and M. Yousef. One-class svms for document classification. *the Journal of machine Learning research*, 2:139–154, 2002.

[27] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[28] T. Mikolov, S. Kombrink, L. Burget, J. H. Černockỳ, and S. Khudanpur. Extensions of recurrent neural network language model. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5528–5531. IEEE, 2011.

[29] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[30] Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases. In *Computer Vision, 1998. Sixth International Conference on*, pages 59–66. IEEE, 1998.

[31] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2:207–218, 2014.

[32] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, page 1642. Citeseer, 2013.

[33] J. Xu, P. Wang, G. Tian, B. Xu, J. Zhao, F. Wang, and H. Hao. Short text clustering via convolutional neural networks. In *Proceedings of NAACL-HLT*, pages 62–69, 2015.

[34] X. Yan, J. Guo, Y. Lan, and X. Cheng. A biterm topic model for short texts. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1445–1456. International World Wide Web Conferences Steering Committee, 2013.