World Wide Web (2018) 21:1705–1719
https://doi.org/10.1007/s11280-018-0542-9

An enhanced short text categorization model with deep abundant representation

Yanhui Gu[1] · Min Gu[1] · Yi Long[2] · Guandong Xu[3] · Zhenglu Yang[4] · Junsheng Zhou[1] · Weiguang Qu[1]

Received: 14 September 2017 / Revised: 22 January 2018 / Accepted: 1 March 2018 /
Published online: 14 April 2018

Abstract Short text categorization is a crucial issue to many applications, e.g., Information Retrieval, Question-Answering System, MRI Database Construction and so forth. Many researches focus on data sparsity and ambiguity issues in short text categorization. To tackle these issues, we propose a novel short text categorization strategy based on abundant representation, which utilizes Bi-directional Recurrent Neural Network(Bi-RNN) with

This article belongs to the Topical Collection: *Special Issue on Deep Mining Big Social Data*
Guest Editors: Xiaofeng Zhu, Gerard Sanroma, Jilian Zhang, and Brent C. Munsell

✉ Min Gu
152202008@stu.njnu.edu.cn

✉ Zhenglu Yang
yangzl@nankai.edu.cn

✉ Weiguang Qu
wgqu@njnu.edu.cn

Yanhui Gu
gu@njnu.edu.cn

Yi Long
longyi@njnu.edu.cn

Guandong Xu
guandong.xu@uts.edu.au

Junsheng Zhou
zhoujs@njnu.edu.cn

[1] School of Computer Science and Technology, Nanjing Normal University, Nanjing, China

[2] School of Geography Science, Nanjing Normal University, Nanjing, China

[3] Advanced Analytics Institute, University of Technology Sydney, Sydney, Australia

[4] Institute of Big Data, College of Computer and Control Engineering, Institute of Statistics, Nankai University, Tianjin, China

Long Short-Term Memory(LSTM) and topic model to catch more contextual and semantic information. Bi-RNN enriches contextual information, and topic model discovers more latent semantic information for abundant text representation of short text. Experimental results demonstrate that the proposed model is comparable to state-of-the-art neural network models and method proposed is effective.

Keywords Short text categorization · Topic model · Bi-directional LSTM

1 Introduction

Along with the development of Internet, the amount of short texts on the Internet grows faster and faster which demands text categorization methods to classify disordered texts into specified categories for further analysis. So far, the need of effective and efficient text categorization approaches have been more and more urgent.

Previous researches mainly focus on text representation learning and classification algorithms. Traditionally, normal text categorization methods utilized Vector Space Mode(VSM) [31] to represent text and used classifiers such as Naïve Bayes [36], Support Vector Machine(SVM) [13], Decision Tree [15], and so forth. With the rapid development of deep learning, many neural network models have been applied to categorization tasks, like image categorization [43] and text categorization [35]. Words in text can be represented as embeddings which act as features for classification and models like Feedforward Neural Network are applied to classifying texts.

However, short text is totally different from normal text, which encounters problems of data sparsity and ambiguity. In previous researches, short texts are represented utilizing Bag-of-Words(BoW) [34]. For example, the short text "Who is Barbara Jordan?" can be represented as "Jordan: 1" according to a feature template, which indicates that the short text has the feature "Jordan". However, it ignores semantic information hidden in short texts and could not tell the specific meaning of "Jordan", whether it is a country or a person's name. In order to utilize more latent semantic information in short texts for disambiguation, topic models, such as Latent Dirichlet Allocation(LDA) [30] and Biterm Topic Model(BTM) [40], are applied into short text categorization. However, these models still use sparse vector representation, which may bring Curse of Dimensionality problem.

In recent years, how to make use of neural network models to obtain effective representations of short text for classification has attracted considerable attention [16, 19, 38, 45]. Embedding representation which takes contextual information into consideration, has solved the Curse of Dimensionality problem effectively. The most straightforward approach is to use models like Convolutional Neural Network(CNN) to generate whole text representation and send to a softmax layer to obtain label distribution of a given text [16].

However, models like CNN ignore the relatedness of words in short texts, which is significant for classification. Because of the finite length of short text, the links between words seem to be more important. To introduce more contextual information, Wang et al. proposed a semantic based neural network model for short text categorization [37]. Though semantic clustering of word embeddings shows positive effective on categorization accuracy, the distance metrics in the proposed short text categorization model are simple and out-of-vocabulary words in this model cannot be utilized, which may lose some valuable information and bring negative influences on accuracy. What's more, CNN and RNN models are combined to form whole short text representations for classification [19]. But

single direction RNN can only introduce the preceding information, which is unsuitable for abundant representation of short text [45].

In previous researches, the difficulty of short text categorization lies in effectiveness of short text representation. Because of the lack of contextual information, the disambiguation in short text becomes more difficult. An effective short text representation can help us to keep away from the current dilemma. To tackle this issue, we propose an abundant representation model for short text categorization. We propose a neural network model utilizing representation from Bi-RNN and LDA to catch more contextual information and latent semantic information for categorization. Compared with other representation models, the representations of whole short text extrated from Bi-RNN and LDA enrich feature representation. The results show that the proposed model is comparable to state-of-the-art neural network models on large-scale dataset and method proposed is effective.

Our contributions are listed as follows:

(1) We address the challenge of short text representation by utilizing representation from Bi-RNN and LDA to introduce more contextual information and latent semantic information for categorization.
(2) We apply latent semantic information extracted from topic model to enhance the neural network representation of a short text, which significantly mitigates ambiguation issue in short text.
(3) The results of our model show that the proposed model is comparable to state-of-the-art neural network models on large-scale dataset.

2 Related work

For normal texts, traditional categorization approaches are based on Statistical Learning Models. When training classification models, many feature extraction methods are applied to acquire features for classification, e.g. Document Frequency(DF) [41], Mutual Information(MI) [41], Chi-square(CHI) [4], and Information Gain(IG) [1]. Time complexities of these methods are low, so the computing speed is fast. After obtaining the features, classifiers based on the features are constructed. Classification algorithms can be classified into three types: (1) algorithms based on statistical learning: Naïve Bayes [36], Maximum Entropy [21], Support Vector Machine [13], Hidden Markov Model(HMM) [7], and K-Nearest Neighbor(KNN) [9]; (2) algorithms based on rules: Decision Tree [15] and Association Rules Model; (3) algorithms based on neural network: CNN [16] and RNN [45]. Among statistical learning algorithms, SVM is the most classic one, which has achieved better performance comparing to other models. HMM is based on sequence and utilizes word probability distribution to construct model, which consists of an unobservable state transition process and an observable observation generation process. The highlight of HMM is no need of large-scale dictionaries and rules. In addition, combination of different classifiers are utilized to improve classification efficiency. For example, Zheng et al. [44] proposed a collaborative work framework based on a linear classifier and Extreme Learning Machine(ELM), which is a single hidden layer feed forward network where the input weights are chosen randomly and the output weights are calculated analytically. To introduce more knowledge for classification, Lauer et al. [17] incorporated prior knowledge into SVM to improve classifiers' performance. Wu et al. [39] also proposed a Wikipedia semantic matching approach for text classification.

Text categorization techniques assume that features are closely related to document categories. On the basis of this hypothesis, there are two kinds of traditional text representation models, namely, Boolean Model and Vector Space Model. The Boolean Model can be considered as a special case of vector model. Depending on whether the feature is presented in the document, the weight of feature is 1 or 0. In Vector Space Model, a document is represented as a vector in feature space, which is also called document vector. Each dimension in a document vector corresponds to a feature in the document. The similarity of two documents is obtained by calculating the cosine of corresponding document vectors. Decision Tree and Association Rules Model are based on Bollean Model. KNN and SVM depend on Vector Space Model. What's more, text can also be represented as graphs in categorization [29, 42].

Traditional text representation and statistic learning feature extraction methods may cause Curse of Dimensionality problem. To tackle this issue, word embedding is proposed [6, 24, 28]. Word embedding distinguishes traditional text representation from providing more semantic information [2]. Word embedding produces word vectors, which can reduce the Euclidean distance between synonyms and similar words. Meanwhile, Word embedding can solve the problem of high dimension in traditional word vectors and the problem of sparsity. Word embedding contains semantic information compared to one-hot representation. In addition, Word embedding takes context information into consideration to enhance word representation in contrast to topic models.

As mentioned in Section 1, short texts have the characteristics of sparse and ambiguity, which make short texts categorization models cannot achieve good performance when utilizing traditional text representation methods. Models which employing Vector Space Model does not consider the contextual relationship between words and would also ignore semantic relations. In order to make up for the lack of valuable information in short texts, Li et al. combined the Wikipedia and concept dimension extension and introduced information from existing knowledge base to strengthen the performance of short text categorization [20]. However, because the features extracted using feature engineering may be sparse sometimes, it is likely to cause Curse of Dimensionality. Based on word embedding, Le and Mikolov proposed a doc2vec model which adds a paragraph vector into word2vec to represent whole short text [18].

In addition, semantic distance of short text can also be regarded as an approach for classification [22, 27, 38]. Short text can be regarded as a Gaussian distribution utilizing word embddings. Nikolentzos el al. proposed to model each short text as a multivariate Gaussian distribution based on the distributed representations of words [27]. Utilizing the similarity function as kernel for SVM, multivariate Gaussian distribution achieves better performance than BoW representation. Wang et al. exploited more contextual information using semantic clustering of word embeddings for classification [38]. However, the similarity of short texts cannot only be measured by applying word embeddings and ignoring the ambiguity of words in short texts.

Recently, more and more attention has been payed to neural network models in Natural Language Processing. Recurrent Neural Network(RNN) [33] has been proved to be effective in solving sequence problems, which is one of the most common neural network models. It can utilize context information in text. However, there are gradient explosion and gradient vanishing problem in training of RNN [11]. LSTM is proposed to solves these problems. LSTM controls the state of cells by adding three kinds of gates in the hidden layer. Based on LSTM, Tree LSTM and Bi-RNN are often used to complete NLP tasks, such as Machine Translation and Word Segmentation. Based on word embedding, Kalchbrenner

et al. introduced the Dynamic Convolutional Neural Network(DCNN) for modeling short texts [14]. On the basis of CNN, they added a dynamic k-max pooling layer into CNN and utilize two convolution layers. Kim et al. [16] proposed to use two input channels to introduce task-specific and static word embeddings simultaneously in CNN. Ji et al. proposed a model which combines CNN and RNN to train short text representation and utilize Artificial Neural Networks(ANN) as a classifier [19]. The combination proves that the sequential information extracted from RNN improves the quality of prediction. Although they can obtain richer textual representations, these models' structures are rather complex.

In order to obtain the latent semantic information in short texts, researchers use topic models, such as Latent Semantic Analysis(LSA), Probabilistic Latent Semantic Analysis(PLSA) and LDA [5]. LDA model is an unsupervised probability-based model which is suitable for large-scale dataset. LDA model can find out topics in text. Phan et al. proposed to utilize topics extracted from LDA as features and Maximum Entropy classifier to classify short texts [30]. Chen et al. proved that leveraging topics at multiple granularity can model text precisely [5].

Though neural network models can solve the problem of dimensionality, the difficulty of short text categorization lies in effective short text representation. Because of the lack of context, the disambiguation in short text becomes more difficult. We need to design an effective text representation method based on the characteristics of short text. Neural network models like CNN may lose the sequence information between words in a short text. Since short text is very short, the connection between words is more close and its context is not negligible. Our proposed model can obtain contextual information utilizing Bi-RNN. To enhance feature representation, topic model can harness latent semantic information. We capture high-order information and word relations to produce complex features, guarantee the classification performance for very short text.

3 Proposed model

Context information plays an important role in classification. For example, when we meet the short text: "Who is Barbara Jordan?", utilizing traditional methods, it is hard for us to tell what "Jordan" means. With large-scale embedding representation and topic model, we can extract useful latent semantic information for short text categorization. As shown in Figure 1, the length of a short text is quite short, which does not contain enough context information for classification. In order to reserve all valuable information in short text, we take out-of-vocabulary words into consideration. The proposed model utilizes well pre-trained word embeddings and randomly initialized out-of-vocabulary word embeddings to initialize the look-up table for words. For short text $s = \{w_1, w_2, ..., w_n\}$, each word w_i can get an embedding x_i as the input of model from look-up table. Bi-RNN is applied to obtaining the abundant contextual representation of a short text. Meanwhile, all files get its own document-topic distribution through LDA. LDA can give us a document-topic distribution which indicates the category of short text. We choose the current short text's document-topic distribution as an additional feature for categorization. We combine the Bi-RNN output representation with document-topic representation as our final representation of short text. Through a dense layer, the complete representation of short text for classification is constructed. Finally, a softmax function is employed as classifier to predict the category which the short text belongs to.

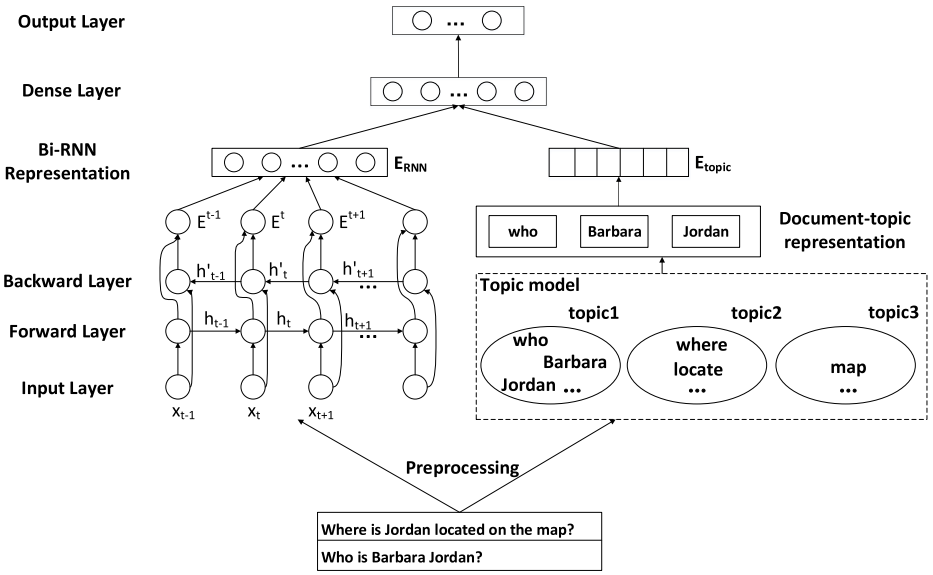


Figure 1 The framework of the proposed deep model

3.1 Input layer

Since short texts often come from Internet, they always contain some out-of-vocabulary words, such as new named entities or abbreviates. For example, the abbreviate "q&a" means "question and answering" in English, which is an out-of-vocabulary word in this task. Usually, we remove these out-of-vocabulary words from traning sets. However, because of the length of short text, we may lose some important information if we do like that. In our model, we use well pre-trained word embeddings from Google News as our lookup dictionary. It has more than 3,000,000 words. For those out-of-vocabulary words, we initialize the embeddings randomly for training. These embeddings are all tuned while training for a strong categorization model. In terms of word vector representation models, CBOW and Skip-gram are two typical models [26]. CBOW can use surrounding word to infer the probability of the intermediate word. However, the probability of surrounding word in Skip-gram model is calculated by the intermediate word. In training, We employ embeddings trained with Skip-gram model.

3.2 Bi-RNN representation

Short text categorization requires the ability to keep track of relevant information that may be arbitrarily far away from current word. Fortunately, Recurrent Neural Network(RNN) is such a neural architecture that employs a structure called short-term memory in order to solve this semantic distance problem. Basic RNN systems have been enhanced with the use of special memory cell units, referred as Long Short-Term Memory neural networks, or LSTMs [10]. In Basic RNN, the state of input transfers from front to back in one direction [8, 12, 25]. However, for a short text, the information it contains is very limited. So we need to enhance the representation of short text.

Bi-directional RNN (Bi-RNN) can keep track of former information and utilize these information to affect latter words representations, which is the combination of two single RNN [32]. Bi-RNN model is composed of input layer, forward layer, backward layer, and output layer, in which the input layer corresponds to the input sequence. The forward layer is a LSTM network structure that is passed from left to right, and the nodes of the hierarchy connect to the nodes of the input layer and historical state of previous input. The backward layer is a LSTM network structure that is passed from right to left, and the nodes of the hierarchy are connected to the same state of the nodes of the input layer and the same level at the same time. The output layer is nodes corresponding to the output sequence, which is connected to the forward transfer layer and the backward transfer layer.

At each time t, we input extracted word embeddings to Bi-RNN, and the output of Bi-RNN is determined by forward and backward layers.

The basic unit, LSTM, utilizes cell state to add and remove information. An unrolled representation of a LSTM Cell is shown in Figure 2. Rectangles represent linear layers followed by the labelled non-linearity. Each cell learns to control the input and previous cell memory. It has three gates to control cell state, including:

(1) Input gate: Restrict the input of current hidden layer cell and determine whether to update the input information of current cell and what information needs to be updated or retained. Given input x_t and hidden state of last word h_{t-1}, the output of the input gate i_t is the value between 0 and 1 through activcation functionσ, which is applies to the input information to determine whether to update the cell status. 1 indicates that the information is allowed to pass and the corresponding values need to be updated. 0 means it is not allowed to pass and values cannot be updated.

$$i_t = \sigma(W^{(i)}x_t + U^{(i)}h_{t-1}) \tag{1}$$

(2) Output gate: Control the current hidden layer node's output and determine whether to output to the next hidden layer or output layer. Through the control of output gate, we can determine which information needs to be output. 1 indicates that the information needs to be output and 0 stands for the opposite.

$$o_t = \sigma(W^{(o)}x_t + U^{(o)}h_{t-1}) \tag{2}$$

(3) Forget gate: Control the stored history information of the hidden layer nodes. Forget gate calculates the value between 0 and 1 based on the state of the last hidden layer and

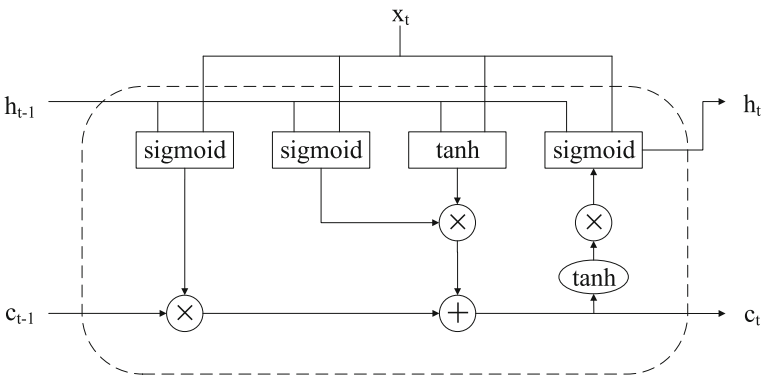


Figure 2 LSTM cell

the input of the current time node, and acts on the state of last cell to determine which information needs to be retained or discarded, where 1 is reserved and 0 is discarded.

$$f_t = \sigma(W^{(f)}x_t + U^{(f)}h_{t-1}) \tag{3}$$

t indicates time step, W and U are parameter matrices. At each time step, the model modifies four states. The forward memory cell is utilized to generate the next memory cell:

$$c_t = f_t \cdot c_{t-1} + i_t \cdot tanh(W^{(c)}x_t + U^{(c)}h_{t-1}) \tag{4}$$

The final hidden state h_t transferred to next time step is defined as:

$$h_t = o_t \cdot tanh(c_t) \tag{5}$$

Bi-RNN representation at time t is composed of forward and backward hidden state:

$$E^t = [\overrightarrow{h_t}; \overleftarrow{h_t}] \tag{6}$$

[;] means concat vectors. For a short text of length n, the final Bi-RNN representation E_{RNN} is the combination result of $\{E^n\}$.

Algorithm 1 Generative process of LDA

```
for all topics k ∈ [1, K] do
    sample mixture components φ_k ~ Dir(β)
end for
for all documents m ∈ [1, M] do
    sample mixture proportion θ_m ~ Dir(α)
    document length N_m ~ Poiss(ξ)
    for all words n ∈ [1, N_m] in document m do
        sample topic index z_{m,n} ~ Mult(θ_m)
        sample term for word w_{m,n} ~ Mult(φ_{m,n})
    end for
end for
```

3.3 Topic representation

As we mentioned in Section 1, an abundant representation needs not only the representation of words it contains, but also the latent semantic information in it. Topic model is usually applied to harness the latent semantic information of text in former researches [3]. Inspired by topic model, we use LDA to help us obtain the latent semantic meaning of a short text.

Given the hyper parameter α and matrix parameter β, we calculate the joint distribution of a topic mixture θ. In training step, we use Gibbs sampling to estimate approximate posterior inference in LDA. The generative process of LDA is shown in Algorithm1. The document-topic distribution is the topic representation E^{topic} in this model.

For example, the distance between short text "who is Barbara Jordan?" and other short texts from different categories is shown in Fig. 3. The calculation of distance is depended on document-topic distribution . The squares indicate the short texts which are belong to the same category of short text "who is Barbara Jordan?". Triangles and circles represent other two categories. As illustrated in the figure, short texts in the same category will gather together. It indicates that document-topic distribution can have some effect on identifing

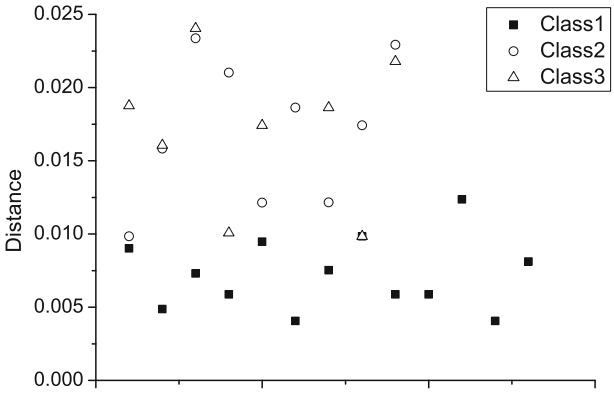


Figure 3 Clusters based on topic distribution of short texts

the categories of short texts, so we utilize document-topic distribution E_{topic} as another representation of short text.

3.4 Output layer and training

As shown in Figure 1, Bi-RNN representation E_{RNN} and topic representation E_{topic} are concatenated to generate the final short text representation E_d by dense layer. A softmax classifier is applied to predict the probability distribution y over categories at output layer. The predicted distribution of category set $\{C\}$ is defined as:

$$E_d = \sigma(W_d \cdot [E_{RNN}; E_{topic}] + b_d) \tag{7}$$

$$y = softmax(W_s \cdot E_d + b_s) \tag{8}$$

$W_d \in \mathbb{R}^{2d_h \times d_e}$; $W_s \in \mathbb{R}^{d_e \times |C|}$; d_h is the dimension of hidden layer; d_e is the dimension of embedding representation; σ is the activation function of dense layer; b_d and b_s are biases of different layers.

Given the label distribution of training examples y', the final training objective is to minimize the cross-entropy loss. A stochastic gradient step is taken on the loss function.

4 Experimental evaluation

4.1 Datasets

Experiments are conducted on Google Snippets dataset [30] and TREC dataset [23]. The specific information of the datasets are shown in Table 1. We divided the training set into 9:1 to reserve a validation set.

Among 8 common short text datasets, Google Snippets has the maximum number of classes. TREC is the second. What's more, the training and test set are already split. In spite of the size of TREC dataset, we still choose the two datasets which are convenient for comparison.

Table 1 Statistics of two datasets

Datasets	Num. of training examples	Num. of test examples	Num. of classes	Vocabulary size
Google snippets	10,060	2,280	8	29,276
TREC	5,452	500	6	9,513

4.2 Evaluation metrics and experiment settings

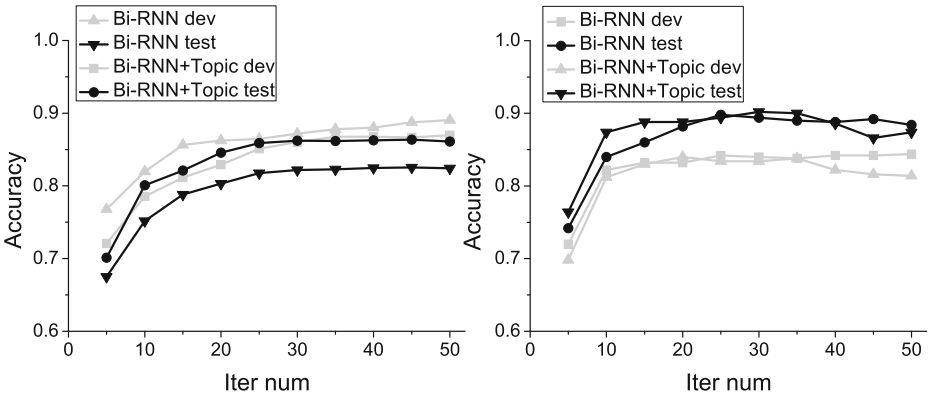
The pre-trained word embedding is trained using word2vec. It contains 3,000,000 words and each word maps to a vector of 300 dimension. We set the hyper parameters as follows: embedding size $d = 300$, hidden layer size $d_{h1} = 200, d_{h2} = 200$, initial learning rate $\alpha = 0.002$. We tune all the parameters on the validation set. Categorization performance is evaluated with the classical evaluation metric: Accuracy, which is defined as:

$$Accuracy = \frac{Num.\ of\ \ correctly\ \ classified\ \ examples}{Num.\ of\ \ all\ \ examples} \tag{9}$$

4.3 Evaluation of performance

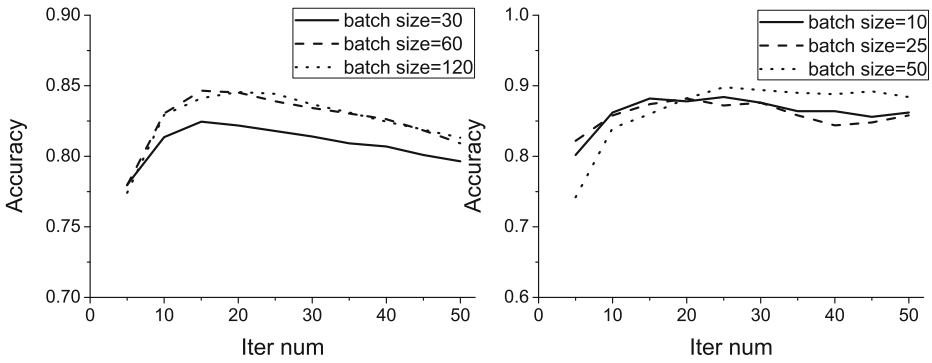
We defined the model only consists of a Bi-RNN as **Bi-RNN** and our proposed model as **Bi-RNN+Topic** in evaluation.

We tune the learning rate and batch size of our model with other fixed parameters. As illustrated in Figure 4, with the increase of iter num, the accuracy of classification also increases on the validation set and test set of both datasets. But the trend of growth gradually slows down and finally decreases. When the curves become smooth, the accuracy of Bi-RNN+Topic is higher than that of Bi-RNN on test sets. With the increase of iter num, the model gradually fits the training data which reflects an increase in accuracy of validation set in the figure. When we meet the threshold, the growth trend of accuracy slows down and even shrinks because the model is over-fitting. We randomly choose hyper-parameters, so Bi-RNN and Bi-RNN+topic is close and the curves are fluctuating on TREC dataset.



(a) Google Snippets (b) TREC

Figure 4 Classification accuracy of the proposed method with respect to parameter iterations on two datasets



(a) Google Snippets (b) TREC

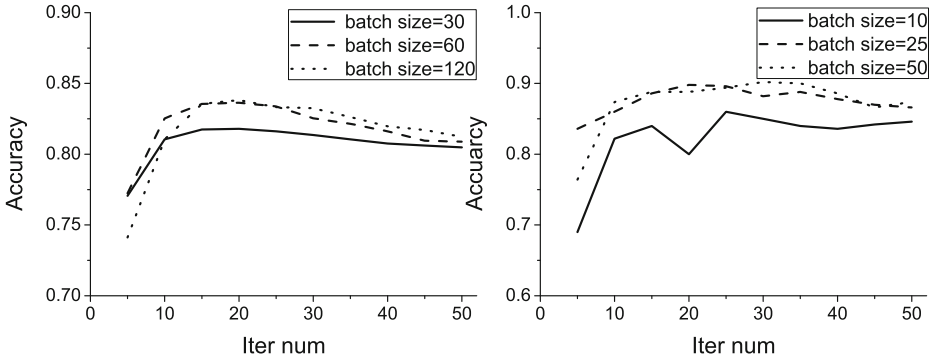
Figure 5 Influence of batch size on accuracy in Bi-RNN

Figures 5 and 6 demonstrate how batch size impacts the cost time of training a converged model. As we can see from the figures, when the accuracy is the same, the model with lower batch size may cost more time. We find that large batch size may bring better performance in a relatively short time and large learning rate may result in over-fitting and miss of well trained model. There may be some fluctuation when training models.

We also computed the sensitivity of the classification to the value of topic num. Specifically, Figure 7 shows how the classification accuracy changes with respect to parameter: topic num on Google Snippets dataset. The highest accuracy is achieved when topic num is close to 8 on Google Snippets dataset, which is the class label num of the dataset. With the increase of topic num, the accuracy of classification will decrease.

Furthermore, we compare our model with previous published models on Google Snippets dataset in Tables 2 and 3. We introduce the following models as baseline, and the details are described:

(1) **LDA+MaxEnt** The model utilizes LDA to discover hidden topics to enrich short text [30].



(a) Google Snippets (b) TREC

Figure 6 Influence of batch size on accuracy in Bi-RNN+Topic

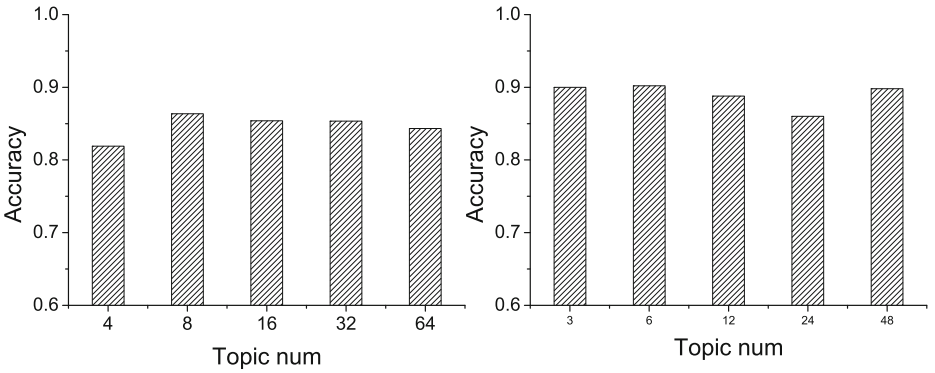


Figure 7 Influence of topic num on accuracy

(2) **Gaussian** Short texts are regarded as multivariate Gaussian distributions based on word embeddings and a similarity method based on multivariate Gaussian distribution is applied for categorization [27].
(3) **Semantic+CNN** The model is based on Convolutional Neural Network (CNN) which employs semantic information from semantic clustering of word embeddings [38].
(4) **DCNN** A Dynamic Convolutional Neural Network(DCNN) for modeling short texts [14]. On the basis of CNN, a dynamic k-max pooling layer is added.
(5) **CNN-Twochannel** A CNN model contains two input channels. One is for task-specific word embedding, and the other is for static word embeddings [16].

As shown in Table 2, our model obtains the highest accuracy on Google Snippets dataset. Using Bi-RNN to get the representation of short text can have a little improvement on accuracy. Abundant representation with Bi-RNN and topic model reduces the influence of data sparsity in short text categorization and achieves the best performance on Google Snippets dataset. In addition, latent semantics in short text can be discovered utilizing topic model, which is benefit for categorization. The accuracy of Bi-RNN+Topic is 1.26% higher than Semantic+CNN, which utilizes semantic clustering to improve the input for CNN. It also adds more semantic information to neural network model, which is proved to be effective. Gaussian is the worst-performing method on Google Snippets. One possible explanation is that Gaussian depends on word embeddings. When a large amount of words do not have pre-trained word embeddings, it does not perform well.

However, the accuracy of our model is lower than traditional statistic learning model on small dataset. As shown in Table 3, neural network models may lose their advantages on

Table 2 The classification accuracy of different models on Google Snippets dataset

Methods	Accuracy(%)
LDA+MaxEnt [30]	82.70
Gaussian [27]	82.24
Semantic+CNN [38]	85.10
Bi-RNN	85.44
Bi-RNN+Topic	86.36

Table 3 The classification accuracy of different models on TREC dataset

Methods	Accuracy(%)
Gaussian [27]	98.20
DCNN [14]	93.00
CNN-TwoChannel [16]	93.60
Bi-RNN	93.20
Bi-RNN+Topic	94.00

small datasets. Compared to traditional models, neural network models have low generalization performance on the small datasets. Though the performance of BiRNN+Topic is not the best on small dataset, the results indicate that this topic-based method is effective when comparing to BiRNN model. As mentioned in last paragraph, Gaussian depends on word embeddings. Short texts in TREC dataset are more related to those articles on which word2vec model was trained than Google Snippets dataset does. Thus Gaussian achieves the best performance on Google Snippets dataset. DCNN and CNN-Twochannel are all CNN models, which cannot utilize continuous contextual information as Bi-RNN does, so the performances of these models are not better than Bi-RNN model.

5 Conclusion

Considering the characteristics of short texts, an abundant representation of text is very important to short text. Previous researches may employ some additional knowledge bases or similarity metrics to enrich representation of a short text. We do not rely on extra knowledge bases and use a simple approach to achieve abundant representation of short texts. This paper proposes a neural network model utilizing representation from Bi-RNN and LDA to introduce more contextual information and latent semantic information for categorization. The results show that the proposed model is comparable to state-of-the-art neural network models on large-scale dataset and method proposed is effective. In the future, we will research on how to improve the efficiency of short text categorization. Meanwhile, we will try to apply multiple representation extracted from models like Bi-RNN and CNN to enhance the representation of short text.

Acknowledgements We would like to thank the anonymous reviewers for their insightful comments. This work is partially supported by National Natural Science Foundation of China under Grant 41571382, U1636116, 11431006, 61472191, 61772278, the Natural Science Research of Jiangsu Higher Education Institutions of China under Grant 15KJA420001, and the Research Fund for International Young Scientists under Grant 61650110510.

References

1. Azhagusundari, B., Thanamani, D.A.S.: Feature selection based on information gain. International Journal of Innovative Technology &, Exploring Engineering **2**(2), 18–21 (2013)
2. Bengio, Y., Schwenk, H., Senécal, J.S., Morin, F., Gauvain, J.L.: Neural probabilistic language models. Springer, Berlin (2006)
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
4. Ceri, S., Bozzon, A., Brambilla, M., Valle, E.D., Fraternali, P., Quarteroni, S.: An introduction to information retrieval. Web Information Retrieval, Springer, Berlin **2013**, 96–102 (2013)

5. Chen, M., Jin, X., Shen, D.: Short text classification improved by learning multi-granularity topics. In: The 22Nd international joint conference on artificial intelligence, IJCAI 2011, Barcelona, July 16-22, pp. 1776–1781 (2011)
6. Collobert, R., Weston, J.: A unified architecture for natural language processing: deep neural networks with multitask learning. In: Machine learning, proceedings of the 25Th international conference, ICML 2008, Helsinki, June 5-9, pp. 160–167 (2008)
7. Ghahramani, Z.: An introduction to hidden markov models and bayesian networks. IJPRAI **15**(1), 9–42 (2001)
8. Graves, A., Mohamed, A., Hinton, G.E.: Speech recognition with deep recurrent neural networks. In: IEEE international conference on acoustics, speech and signal processing, ICASSP 2013, Vancouver, May 26-31, pp. 6645–6649 (2013)
9. Han, E., Karypis, G., Kumar, V.: Text categorization using weight adjusted K-Nearest neighbor classification. In: The 5Th Pacific-Asia conference on knowledge discovery and data mining, PAKDD 2001, Hong Kong, April 16-18, pp. 53–65 (2001)
10. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
11. Hochreiter, S., Younger, A.S., Conwell, P.R.: Learning to learn using gradient descent. In: International conference on artificial neural networks, ICANN 2001, Vienna, August 21-25, pp. 87–94 (2001)
12. Hüsken, M., Stagge, P.: Recurrent neural networks for time series classification. Neurocomputing **50**, 223–235 (2003)
13. Joachims, T.: Text categorization with support vector machines: learning with many relevant features. In: The 10Th European conference on machine learning, ECML 1998, Chemnitz, April 21-23, pp. 137–142 (1998)
14. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences. In: The 52Nd annual meeting of the association for computational linguistics, ACL 2014, June 22-27, Baltimore, vol. 1: Long Papers, pp. 655–665 (2014)
15. Karbassi, A., Mohebi, B., Rezaee, S., Lestuzzi, P.: Damage prediction for regular reinforced concrete buildings using the decision tree algorithm. Comput. Struct. **130**(1), 46–56 (2014)
16. Kim, Y.: Convolutional neural networks for sentence classification. In: The 2014 conference on empirical methods in natural language processing, EMNLP 2014, Doha, October 25-29, pp. 1746–1751 (2014)
17. Lauer, F., Bloch, G.: Incorporating prior knowledge in support vector machines for classification: a review. Neurocomputing **71**(7-9), 1578–1594 (2008)
18. Le, Q.V., Mikolov, T.: Distributed representations of sentences and documents. In: The 31Th international conference on machine learning, ICML 2014, Beijing, June 21-26, pp. 1188–1196 (2014)
19. Lee, J.Y., Dernoncourt, F.: Sequential short-text classification with recurrent and convolutional neural networks. In: The 2016 conference of the North American chapter of the association for computational linguistics: human language technologies, NAACL HLT 2016, San Diego, June 12-17, pp. 515–520 (2016)
20. Li, J., Cai, Y., Cai, Z., Leung, H., Yang, K.: Wikipedia based short text classification method. In: Database systems for advanced applications - DASFAA 2017 international workshops: BDMS, BDQM, SeCoP, and DMMOOC, Suzhou, March 27-30, pp. 275–286 (2017)
21. Li, J., Rao, Y., Jin, F., Chen, H., Xiang, X.: Multi-label maximum entropy model for social emotion classification over short text. Neurocomputing **210**, 247–256 (2016)
22. Li, L., Zhong, L., Xu, G., Kitsuregawa, M.: A feature-free search query classification approach using semantic distance. Expert Systems with Applications **39**(12), 10,739–10,748 (2012)
23. Li, X., Roth, D.: Learning question classifiers. In: 19Th international conference on computational linguistics, COLING 2002, Taipei, August 24 - September 1, pp. 556–562 (2002)
24. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv:1301.3781 (2013)
25. Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., Khudanpur, S.: Recurrent neural network based language model. In: The 11Th annual conference of the international speech communication association, INTERSPEECH 2010, Makuhari, September 26-30, pp. 1045–1048 (2010)
26. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: The 27Th annual conference on neural information processing systems, Lake Tahoe, Nevada, December 5-8, pp. 3111–3119 (2013)
27. Nikolentzos, G., Meladianos, P., Rousseau, F., Vazirginannis, M., Stavrakas, Y.: Multivariate gaussian document representation from word embeddings for text categorization. In: European chapter of the association for computational linguistics, EACL 2017, Barcelona, April 3-7, pp. 450–355 (2017)
28. Paccanaro, A., Hinton, G.E.: Learning distributed representations of concepts using linear relational embedding. IEEE Trans. Knowl. Data Eng. **13**(2), 232–244 (2001)

29. Papadakis, G., Giannakopoulos, G., Paliouras, G.: Graph vs. bag representation models for the topic classification of Web documents. World Wide Web **19**(5), 887–920 (2016)
30. Phan, X.H., Nguyen, M.L., Horiguchi, S.: Learning to classify short and sparse text & Web with hidden topics from large-scale data collections. In: The 17Th international conference on World Wide Web, WWW 2008, Beijing, April 21-25, pp. 91–100 (2008)
31. Salton, G., Wong, A., Yang, C.: A vector space model for automatic indexing. Commun. ACM **18**(11), 613–620 (1975)
32. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Trans. Signal Process. **45**(11), 2673–2681 (1997)
33. Socher, R., Lin, C.C., Ng, A.Y., Manning, C.D.: Parsing natural scenes and natural language with recursive neural networks. In: The 28Th international conference on machine learning, ICML 2011, Bellevue, June 28 - July 2, pp. 129–136 (2011)
34. Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H., Demirbas, M.: Short text classification in twitter to improve information filtering. In: The 33Rd international ACM SIGIR conference on research and development in information retrieval, SIGIR 2010, Geneva, July 19-23, pp. 841–842 (2010)
35. Toh, K., Lu, J., Yau, W.: Global feedforward neural network learning for classification and regression. In: Energy minimization methods in computer vision and pattern recognition, third international workshop, EMM-CVPR 2001, Sophia Antipolis, September 3-5, pp. 407–422 (2001)
36. Troussas, C., Virvou, M., Espinosa, K.J., Llaguno, K., Caro, J.: Sentiment analysis of facebook statuses using naive bayes classifier for language learning. In: The 4Th international conference on information, intelligence, systems and applications, IISA 2013, Piraeus, July 10-12, pp. 1–6 (2013)
37. Wang, P., Xu, B., Xu, J., Tian, G., Liu, C., Hao, H.: Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification. Neurocomputing **174**, 806–814 (2016)
38. Wang, P., Xu, J., Xu, B., Liu, C., Zhang, H., Wang, F., Hao, H.: Semantic clustering and convolutional neural network for short text categorization. In: The 53Rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing of the asian federation of natural language processing, ACL 2015, Beijing, vol. 2: Short Papers, July 26-31, pp. 352–357 (2015)
39. Wu, Z., Zhu, H., Li, G., Cui, Z., Huang, H., Li, J., Chen, E., Xu, G.: An efficient wikipedia semantic matching approach to text document classification. Inform. Sci. **393**, 15–28 (2017)
40. Yan, X., Guo, J., Lan, Y., Cheng, X.: A Biterm topic model for short texts. In: The 22Nd international World Wide Web conference, WWW 2013, Rio De Janeiro, May 13-17, pp. 1445–1456 (2013)
41. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: The 14Th international conference on machine learning, ICML 1997, Nashville, July 8-12, pp. 412–420 (1997)
42. Yao, L., Sheng, Q.Z., Ngu, A.H.H., Gao, B.J., Li, X., Wang, S.: Multi-label classification via learning a unified object-label graph with sparse representation. World Wide Web **19**(6), 1125–1149 (2016)
43. Zhang, Y., Dong, Z., Wu, L., Wang, S.: A hybrid method for MRI brain image classification. Expert Systems with Applications **38**(8), 10,049–10,053 (2011)
44. Zheng, W., Tang, H., Qian, Y.: Collaborative work with linear classifier and extreme learning machine for fast text categorization. World Wide Web **18**(2), 235–252 (2015)
45. Zhou, C., Sun, C., Liu, Z., Lau, F.C.M.: A c-LSTM neural network for text classification. arXiv:1511.08630 (2015)