DSI DATA SCIENCE AND INFORMETRICS
VOL.4, NO.1, FEB. 2024

RESEARCH ARTICLE

A comparative study of Neural Sinkhorn Topic Model based on different word embedding

Fenggao Niu[a*], Sijia Wang[b], Bikun Chen[c]

a. School of Mathematical Sciences, Shanxi University, Taiyuan, China

b. School of Computer and Information Technology, Shanxi University, Taiyuan, China

c. School of Social Science, Soochow University, Suzhou, China

ABSTRACT

Topic models play an important role in many tasks of natural language processing. The early topic models are based on the bag-of-words assumption, which do not consider the context relationship and face the sparsity problem. Word embedding can map words into a dense vector in a low-dimensional space and preserve the relationship information between words. Therefore, word embedding vectors such as Word2Vec, GloVe, and fastText have been introduced into neural topic models to improve the modeling effect. However, current topic models do not fully consider the characteristics of each word embedding, and only use one of them. In order to study the advantages and disadvantages of different word embeddings and their influence on the topic model, and then provide a basis for the reasonable choice of embedding methods, this paper explores the influence mechanism of different word embeddings on the topic model (Neural Sinkhorn Topic Model is selected) and text classification task by changing the word embeddings and their dimensions. The results show that: ⅰ) Word embedding trained by large corpora has the greatest impact on topic modeling and document classification, with an increase of 23% in topic coherence and topic diversity indicators, and an average increase of 68% in classification indicators; ⅱ) Word embedding trained by Skip-gram model is suitable for long text topic modeling, and word embedding trained by GloVe model is suitable for short text topic modeling; ⅲ) Word embedding trained by fastText model has poor performance in the topic model, and the effect of combining with the topic model for document classification is better; ⅳ) The selection of word embedding dimension also has an impact on the topic model, and the most suitable word embedding dimension should be selected according to the actual situation.

KEYWORDS

Word embedding; Topic model; Neural sinkhorn topic model; Comparative study; Text classification

* Corresponding author: nfgao@sxu.edu.cn

1 Introduction

As a successful text analysis technique, the topic model has found its presence in various tasks of natural language processing, such as text classification, sentiment analysis, topic detection, etc. Topic model is a useful tool for discovering latent topics in a collection of documents that can describe an interpretable semantic concept. In the past decade, the three-layer Bayesian probabilistic topic model has been fully studied, but with the development of deep learning, it is gradually overwhelmed by the emerging deep learning technology. In order to solve the problems existing in the traditional probabilistic topic models, such as complex reasoning processes, inability to parallel computing and difficult to realize automation, topic model is combined with a deep neural network to open up new neural topic models (NTMs). So that the topic model can use neural network to improve the performance and efficiency of modeling. Due to the flexibility and scalability that traditional probabilistic topic models do not have, NTMs have been applied to natural language processing tasks such as text generation (Tang et al., 2019), document summarization (Cui et al., 2020) and question-answering systems, which are difficult to apply to traditional topic models.

Recently, word embedding representation techniques combined with NTMs have achieved considerable improvements in topic modeling. Word embedding techniques model each word in a document based on the distribution of words around it and summarize these statistics in terms of low-dimensional embedding representation. This representation is widely used in natural language processing tasks, such as information retrieval (Manning et al., 2008), document classification (Sebastiani, 2002), and question-answering systems (Tellex et al., 2003).

In order to fully combine the advantages of topic model and word embedding representation technology, the existing research is mainly divided into two aspects. On the one hand, the pre-trained word embedding is introduced into topic modeling, and the low-dimensional word embedding is used to better represent the information of words and further improve the performance of topic model. The early topic models mainly use the bag of words model for modeling, without considering the context of words, and short texts also face the problem of data sparsity. Word embedding, as one of the important breakthroughs in natural language processing, can map each word to a dense vector in a low-dimensional space, which not only alleviates the problem of data sparsity, but also retains the relationship information between words. Therefore, pre-trained word embeddings are introduced into the topic model, aiming to improve the performance of the topic model. At present, static word embedding is mainly combined with topic models, and there are three main methods: Word2Vec (divided into two modes: CBOW and Skip-gram), GloVe and fastText. Each of these word embedding methods has its advantages and disadvantages. However, the current models do not fully consider the characteristics of each word embedding, and only select one word embedding method to combine with the topic model. For example, Li et al. (2016) introduced the General Polya Urn (GPU) model and combined it with the Dirichlet Multinomial Mixtures (DMM) model to propose the GPU-DMM method. By introducing the pre-trained 300-dimensional Word2Vec word embedding representation, the GPU model is used to improve the semantic relationship between words. The Embedded Topic Model (ETM) proposed by Dieng et al. (2020) is a document generation model that combines Latent Dirichlet Allocation (LDA) model and Skip-gram model. He et al. (2021) proposed a neural topic model based on optimal transportation, named Neural Sinkhorn Topic Model (NSTM). The model uses

pre-trained 50-dimensional GloVe word embedding as the input of the topic model, which greatly improves the performance of the topic model. It is also the model with the best combination effect of static word embedding and neural topic model.

On the other hand, topic model and word embedding can also be combined by joint learning, which can not only use word embedding to improve the performance of topic models, but also use topic models to train word embedding. Shi et al. (2017) proposed the Skip-gram Topical word Embedding (STE) model, which can simultaneously learn topic representation and word embedding in a single framework. The experimental results show that the STE model can indeed generate effective word embedding representations and latent topics. Law et al. (2018) established an EM algorithm framework that can iteratively optimize the topic representation and word embedding representation. Xu et al. (2018) proposed a new Wasserstein method with a distillation mechanism, which can learn the distribution of topics, the embedding representation of words, and the word distribution of the optimal transfer of topics to documents in a unified framework.

This paper aims to study the influence of different word embedding methods on the topic model and find the best combination of word embedding and topic model. NSTM utilizes the advanced properties of modeling geometric structures on probability distribution spaces using OT, which can achieve a better balance between obtaining good document representations and generating coherent/diverse topics. NSTM also alleviates the burden of designing complex sampling schemes for the posterior of NTM. What's more, NSTM is a natural way to integrate pre-trained word embeddings, which has been proven to alleviate the problem of insufficient word co-occurrence information in short texts. Through a large number of experiments, NSTM can be shown to have state-of-the-art performance in both topic quality and document representation for regular texts and short texts. Therefore, this paper is based on NSTM to study the impact of different word embedding methods on topic models. Through experiments on NSTM with multiple word embedding methods and multiple dimensions, the influence mechanism of different word embedding methods and dimensions on the topic model is discussed, which provides a reference for subsequent topic modeling and application research. The main work of this paper is:

ⅰ) Based on the analysis of word embedding theory, this paper studies the generation mechanism of different word embedding methods.

ⅱ) The experiment compares the influence of different word embedding methods on NSTM, obtains the best combination of word embedding and NSTM, and improves the model.

ⅲ) This paper discusses the influence mechanism of word embedding on topic model from both theoretical and practical aspects.

2 Neural topic models and NSTM

Topic modeling, as an unsupervised approach, aims to mine a set of latent topics from a set of documents, where each topic describes an interpretable semantic concept. Traditional probabilistic topic models are mainly represented by LDA model (David et al., 2003) and its extended models. In the past two decades, deep learning methods have gradually replaced traditional machine learning techniques and made significant breakthroughs in discovering complex structures in large datasets, which have been widely used in image processing, speech

recognition, natural language processing and other fields. With the development of neural networks in the field of text mining, researchers began to use simple neural network structures to reconstruct the generation process of probabilistic topic models, and then variational autoencoders (Kingma & Welling, 2014) were used to construct topic models, and a series of variants of neural topic models appeared. However, there are still some shortcomings in the existing NTMs. ⅰ) For many current NTMs, the training goal is to make the error between the generated samples and the original samples smaller, which means the quality of the obtained topics is not so good. ⅱ) For short documents, the word co-occurrence information of each document is insufficient, so there is a problem of data sparsity. NTM is more susceptible to data sparsity due to the use of encoder and decoder structures. In order to solve the two shortcomings of NTM, He et al. (2021) proposed a neural topic model based on optimal transportation, named NSTM.

NSTM is a modification of neural topic models built on the optimal transport framework. Similar to the standard NTM, it consists of an encoder and a decoder structure. The encoder outputs the topic distribution $\boldsymbol{z}$ of the document through the bag-of-words representation $\boldsymbol{x}$ of the input document, and the decoder projects $\boldsymbol{z}$ back into the word space to reconstruct $\boldsymbol{x}$, where $\boldsymbol{x}$ and $\boldsymbol{z}$ are two discrete probability distributions of the document about words and topics, respectively. Unlike NTM, NSTM directly acts as the loss function of the model by minimizing the OT distance between $\boldsymbol{x}$ and $\boldsymbol{z}$. The loss function of the final model is a combination of the Sinkhorn distance and cross-entropy between $\boldsymbol{x}$ and $\boldsymbol{z}$, where the expected multinomial log-likelihood is used to guide the optimization of the Sinkhorn distance:

$$\max_{\theta,G}\left(\epsilon\tilde{x}^{\mathrm{T}}log\phi(z)-d_{M,\alpha}(\tilde{x},z)\right) \quad (1)$$

where, $z=softmax\left(\theta(\tilde{x})\right)$, θ is the weight matrix of the encoder, $\tilde{x}\in\Delta^{V}$ obtained by normalising x: $\tilde{x}:=x/S$ where $S:=\sum_{v=1}^{V}x$ is the length of a document. Also, each document is associated with a distribution over K topics: $z\in\Delta^{K}$, each entry of which indicates the proportion of one topic in this document. $M\in\mathbb{R}_{>0}^{V\times K}$ is the cost matrix, where m_{vk} indicates the semantic distance between topic k and word v. We specify the following construction of M:

$$m_{vk}=1-\cos(e_v,g_k) \quad (2)$$

where $g_k\in\mathbb{R}^{L}$ and $e_v\in\mathbb{R}^{L}$ are the embeddings of topic k and word v, respectively. ϵ is the hyperparameter that controls the weight of the expected likelihood; α is the hyperparameter for the Sinkhorn distance.

We train the model with an input document x and a pre-trained word embedding E and two hyperparameters ϵ and α, and finally take the output θ as the document-topic distribution and ϕ as the topic-word distribution. The model can additionally obtain an embedding representation G of the topic.

3 Word embedding representation methods

The original word embedding representation originated from the distributed hypothesis proposed by Firth (1957). However, this word embedding has the characteristics of sparse and high-dimensional. Therefore, researchers have proposed dimensionality reduction methods for word embeddings. For example, Deerwester et al. (1990) used singular value decomposition

method to decompose the document matrix to reduce the dimension of word embedding. Muntsa et al. (2014) reduced the dimension of word embedding by deleting the dimensions with lower frequency of word pairs. With the development of deep learning and neural networks, researchers have begun to use neural networks to train low-dimensional representations of words. Google proposed Word2Vec (Mikolov et al., 2013), which includes two modes: CBOW (The Continuous Bag-of-Words) and Skip-Gram, and their objective function is the relationship between the target word and the context word. Since then, many studies have improved the Word2Vec model. Bansal et al. (2014) improved the performance of the model by training word embedding based on syntactic relations of sentences. In order to generate embedded representations more suitable for tasks involving syntax (Wang et al., 2015). Although the word embedding obtained by neural network training can successfully capture fine-grained semantic and syntactic rules, it is a black-box operation, and the statistical ideas contained in it cannot be effectively explained. Therefore, Jeffrey et al. (2014) proposed the GloVe model, which combines two advantages: the global matrix factorization method and local context window method and can generate a meaningful substructure vector space. fastText is a text classification model released by Facebook, which is an improvement of the CBOW model (Joulin et al., 2017).

3.1 Word2Vec

The idea behind Word2Vec is that words with similar contexts also have similar semantics. The model tries to predict words directly by using neighboring words and learns a low-dimensional dense word embedding. To learn these word embeddings through a neural network, Word2Vec comes in two forms: the CBOW model, which predicts target word probability by input context word information, and the Skip-Gram model, which predicts context probability by input target word information. These two prediction methods share the restriction that the probabilities of each word given the same input sum to one.

3.2 GloVe

GloVe model is a word embedding representation obtained by factorizing the word co-occurrence matrix. The model first creates a word co-occurrence matrix through the text corpus, and then uses the gradient descent method to decompose the matrix. The loss function used in the model is the least square loss.

3.3 fastText

fastText is a fast text classification algorithm, which is an improvement on CBOW. The target word predicted by CBOW is changed to the label of the document, so as to realize document classification. Compared with other classification algorithms under neural network architecture, fastText has two important optimizations: the addition of Hierarchical Softmax and N-gram features. The central idea of fastText is to superimpose and average the words and N-gram vectors in the document to obtain the entire text vector, and then use character-level N-gram features as auxiliary features and Hierarchical Softmax to output the category labels corresponding to the words.

3.4 Comparison of word embedding methods

In general, each word embedding representation method has its advantages and disadvantages. Table 1 analyzes the similarities and differences of various word embedding methods in the training process, model input, and loss function.

Table 1 Comparison of the existing word embedding methods

Word embedding methods	Training process	Model input	Loss function
CBOW	Based on the local corpus	Context of the target word	Cross entropy loss
Skip-Gram	Based on the local corpus	Target word	Cross entropy loss
fastText	Based on the local corpus	Multiple words and their n-gram features	Cross entropy loss
GloVe	Based on the global corpus	Word co-occurrence matrix	Least squares loss

ⅰ) The training process. Word2Vec (CBOW & Skip-Gram) and fastText are trained based on the local corpus, which may ignore the global information of the document, while the GloVe model is trained on the global corpus, and the word co-occurrence matrix is used to model, which can express the semantic information between words more accurately.

ⅱ) The model input. The input of the CBOW model is the One-Hot encoded-word vector of the context word, and the input of the Skip-Gram model is the One-Hot encoded-word vector of the target word. The total prediction of the CBOW model is the number of words in the vocabulary, and the total prediction of the Skip-Gram model is the number of customized context words. Therefore, the training time of Skip-Gram is longer, and the word vectors learned by Skip-gram are more detailed than those learned by CBOW. The input of GloVe is a co-word matrix constructed based on the corpus, which fuses contextual information and global information. The input of the fastText model is multiple words and their n-gram features.

ⅲ) The loss function. Both the Word2Vec (CBOW & Skip-Gram) model and the fastText model use weighted cross-entropy as a loss function to measure the difference between the predicted value and the true value. The greater the difference is, the greater the cross-entropy loss. Since these models only consider local semantic information and lack global information, the GloVe model adopts a square loss-based way to incorporate global information.

4 Datasets and data preprocessing

Experiments were conducted using three benchmark text datasets to evaluate the impact of different word embedding methods on NSTM. The dataset contains one long text dataset and two short text datasets.

Dataset 1: 20 Newsgroups (20NG) dataset (Lang, 1995) is one of the international standard datasets used for text classification, text mining, and information retrieval research. It includes

18,846 news documents on 20 different topics. There are three versions of this dataset, and the second version is used in this paper.

Dataset 2: The Web Snippets (WS) dataset is a short text dataset published by Phan et al. in 2008 (Phan et al., 2008). It includes 12,337 search segments in eight categories, and the average length of the segments is about 13 words.

Dataset 3: Tag My News (TMN) dataset is a news dataset published by Vitale et al. in 2012 (Daniele et al., 2012). The data is collected from all news stories published by three newspapers, the New York Times, Reuters, and America Day, from March 2011 to June 2011. The headline and summary are selected, and the average length of each text is about 20 words. There are 7 categories in the dataset: Entertainment, Tech, Sports, US, Health, Business, and World.

First, the three datasets are preprocessed, and the processed datasets are shown in Table 2. We split each dataset into a training set (80%) and a test set (20%).

Table 2 The statistics of the datasets

Datasets	Number of docs	Vocabulary size	Number of labels	Dataset type	Average length of each text	Training set	Testing set
20 Newsgroups	18,846	22,636	20	Long text dataset	284	15,076	3,770
Web Snippets	12,337	10,052	8	Short text dataset	13	9,867	2,470
Tag My News	32,597	13,368	7	Short text dataset	20	26,077	6,520

5 Experiments

5.1 Experimental settings

Datasets: Experiments are conducted on three widely used benchmark text datasets, namely the 20NG dataset, the WS dataset, and the TMN dataset. The details are as described in the previous section.

Settings for NSTM: NSTM is implemented on TensorFlow, and for the encoder θ, a fully connected neural network with a hidden layer of 200 units using ReLU as the activation function, followed by a dropout layer (rate=0.75) and a batch standard layer. For the Sinkhorn algorithm, the maximum number of iterations is 1000 and the stopping tolerance is 0.005. In all experiments, we fix the hyperparameter $\alpha = 20$ of the Sinkhorn distance, and the hyperparameter $\epsilon = 0.07$ of the expected multinomial log-likelihood. The learning rate of NSTM is 0.001, batch size 200 for maximally 50 iterations, and epochs 500. In order to compare topic coherence (TC) with topic diversity (TD) on different datasets, we fix the number of topics $K = 100$.

Selection of word embedding dimension: In the original papers of Word2Vec, GloVe and fastText, 300 is selected as the dimension of word embedding, and this setting is mostly used in subsequent studies. Also commonly used are 200, 100, or 50 dimensions. Therefore, this paper chooses these dimensions for comparison and studies the influence of different word embedding dimension methods on the topic model. We select CBOW, Skip-gram, GloVe, and fastText word

embeddings of four dimensions (50, 100, 200, 300) on each dataset and apply different word embeddings of each dimension to NSTM to obtain the topic word distribution.

5.2 Training word embeddings

To improve comparability, CBOW, Skip-gram, GloVe and fastText are trained on three datasets respectively, so as to avoid the influence of different corpora on word embeddings. In addition, to explore the influence of the word embeddings trained on the dataset in the experiment and the word embeddings pre-trained in the large corpus on the topic model, the GloVe word embeddings trained on the Wikipedia corpus are selected for comparison experiments.

5.3 Evaluation metrics

We use TC and TD as performance measures of topic quality. TC measures the semantic coherence of the top words in a topic. We use the top 10 words for each topic to compute the normalized pointwise mutual information (NPMI). TD measures the diversity of topics found. We take the top 25 words of a topic and calculate the percentage of unique words. TD close to 0 indicates redundant topics; TD close to 1 indicates more diverse topics.

For text classification, we use the commonly used accuracy, precision, recall, and F1-score as evaluation metrics.

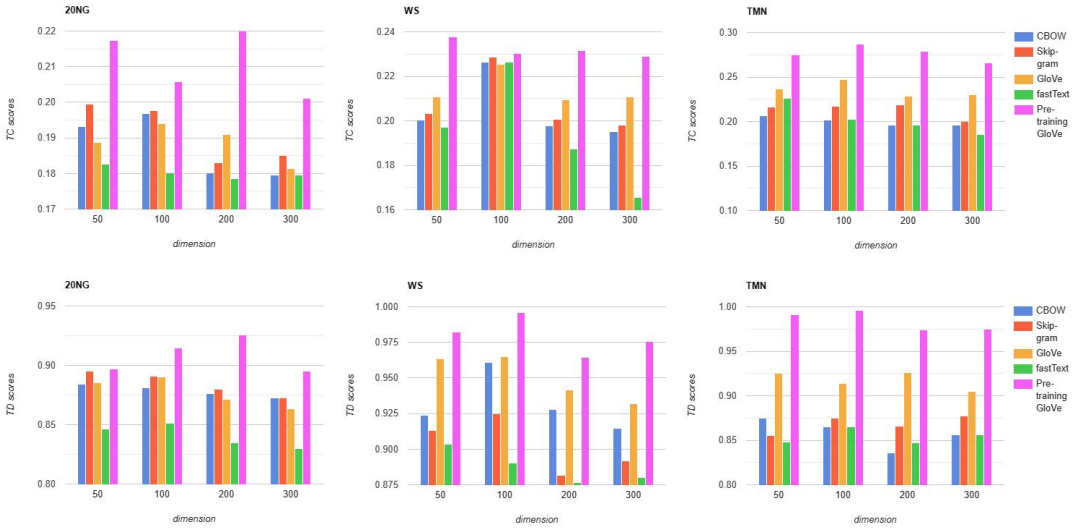
5.4 Results

Topic model results: The comparison is made from three perspectives: dataset, embedding representation method, and embedding dimension. That is, four-word embedding methods of CBOW, Skip-gram, GloVe and fastText with different dimensions trained on the corresponding dataset, and the pre-trained GloVe word embedding is used as the input of NSTM. The experimental evaluation results are shown in Figure 1 and Table 3.

From the comparison of results in Figure 1, it can be seen that: ⅰ) When the pre-trained GloVe word embeddings are used for topic modeling, their results are higher than other methods in both TC and TD indicators; ⅱ) The word embeddings trained by fastText for topic modeling have the worst results; ⅲ) Comparing the two forms of Word2Vec, when the word embeddings trained by Skip-gram are used for topic modeling on the three datasets, the TC and TD indicators are better than the word embeddings trained by CBOW, but the advantages are not obvious. ⅳ) The topic modeling results of GloVe-trained word embeddings on the 20NG dataset are not as good as those of Skip-gram-trained word embeddings, but the results of GloVe-trained word embeddings on WS dataset and TMN dataset are better than those of Skip-gram trained word embeddings; ⅴ) For each word embedding method, most of the results on 100-dimensional word embeddings are higher than the results on other dimensions.

Table 3 only shows the experimental results of NSTM under 100 dimensions of different word embeddings. It can be seen that in the three datasets, in terms of TC and TD indicators, the pre-trained GloVe performs the best, with TC and TD indicators higher than other methods, especially in the TMN dataset, where TC and TD are 0.0853 and 0.1309 higher than the lowest value, respectively. Combined with the characteristics of the TMN dataset itself, we can deduce that because the TMN dataset collects the titles and short summaries of news reports, on the one hand, the length of each text is short, and the words contain less context information. On the other hand, it leads to low similarity of each text and insufficient word co-occurrence information.

This makes the gap between the information contained in the word embeddings trained using the TMN dataset and the word embeddings trained using the large corpus larger than the other two datasets.



20NG: 20 Newsgroups; WS: Web Snippets; TMN:Tag My News; TC: topic coherence; TD: topic diversity.

Figure 1 Experimental results of NSTM with different word embedding methods

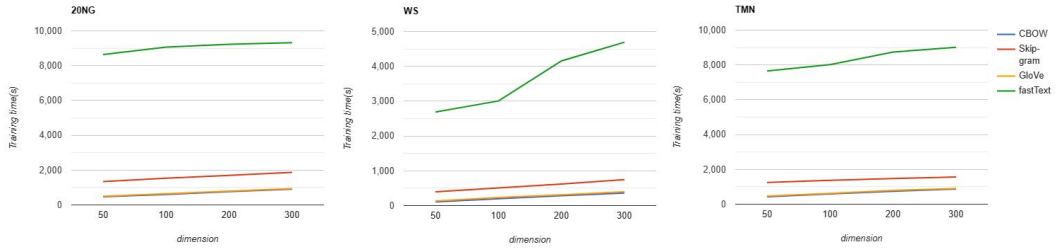
Table 3 Experimental results of NSTM with different word embedding methods of 100 dimensions

Word embedding methods	20NG		WS		TMN	
	TC	TD	TC	TD	TC	TD
CBOW	0.1968	0.8813	0.2263	0.9610	0.2014	0.8652
Skip-gram	0.1976	0.8907	0.2286	0.9250	0.2170	0.8752
GloVe	0.1939	0.8903	0.2254	0.9650	0.2470	0.9136
fastText	0.1801	0.8513	0.2263	0.8900	0.2028	0.8651
pre-trained GloVe	0.2058	0.9147	0.2301	0.9961	0.2867	0.9960

20NG: 20 Newsgroups; WS: Web Snippets; TMN:Tag My News; TC: topic coherence; TD: topic diversity.

Training speed of word embeddings: The training speed of word embeddings is also an important factor affecting the use of the model, Figure 2 and Table 4 show the training time of the four word embedding methods except for pre-trained GloVe under different datasets.

The comparison of training time in Figure 2 and Table 4 shows that: ⅰ) No matter which training method is used, the training time increases with the increase of word embeddings dimension; ⅱ) In the same dimension, the training speed of CBOW model is the fastest, the training speed of GloVe model is slightly slower than CBOW model, the training time of Skip-gram model is about 2.5 times of CBOW model, and the slowest training speed is fastText model, about 15 times of CBOW model.



20NG: 20 Newsgroups; WS: Web Snippets; TMN:Tag My News.

Figure 2 The time cost of training word embeddings with different word embedding methods

Table 4 The time cost of training word embeddings with different word embeddings of 100 dimensions

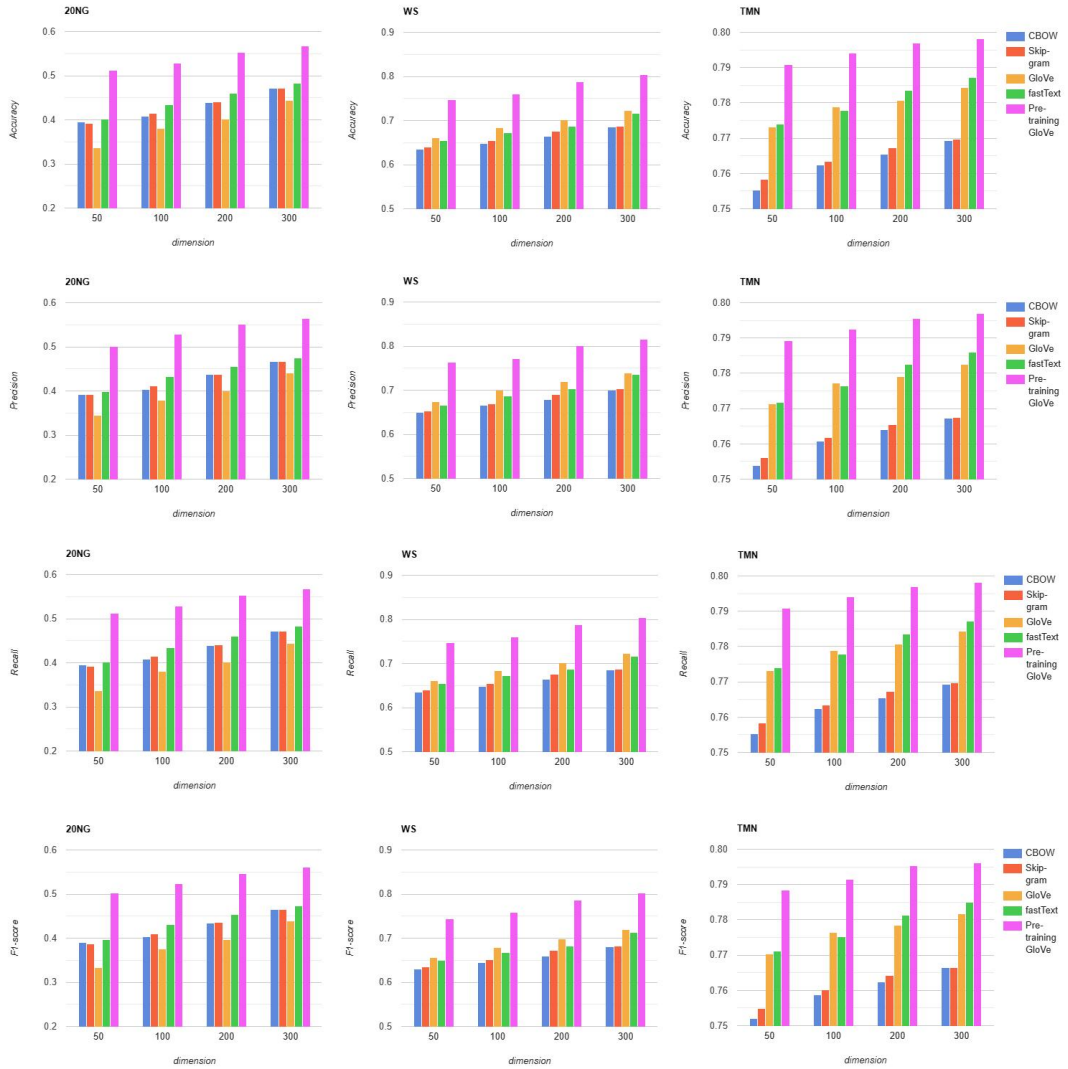
Word embedding methods	Training time (s)		
	20NG	WS	TMN
CBOW	603	199	589
Skip-gram	1530	506	1369
GloVe	637	235	615
fastText	9067	3005	8016

20NG: 20 Newsgroups; WS: Web Snippets; TMN:Tag My News.

5.5 Text classification tasks and evaluation

To compare the extrinsic prediction performance, we use document classification as a downstream task. We used the document-topic matrix obtained by using each word embedding representation for NSTM in Section 5.4 as features for classification and trained a random forest with decision trees of 800 to predict the category of each document.

From the comparison of results in Figure 3, it can be seen that: ⅰ) On the three datasets, when the pre-trained GloVe word embeddings are used for document classification, the results are higher than other methods in the four indicators of accuracy, precision, recall and F1-score. ⅱ) On the 20NG dataset, when the word embeddings trained by GloVe are used for document classification, the results are lower than other methods in four indicators. ⅲ) On WS and TMN datasets, the word embeddings trained by Skip-gram and CBOW for document classification are lower than other methods in four indicators, while the word embeddings trained by GloVe and fastText for document classification have similar results, and the text classification effect is better. ⅳ) On the three datasets, each embedding method shows that the larger the dimension of word embedding, the better the effect of text classification.



20NG: 20 Newsgroups; WS: Web Snippets; TMN:Tag My News.

Figure 3 Experimental results of text classification with different word embedding methods

6 Conclusion

An in-depth study of the influence of different word embedding representations on the results of topic models can provide optimization strategies for the selection of embedding methods and model training. Firstly, by analyzing the generation principle of three word embedding representation methods, Word2Vec, GloVe and fastText, their similarities and differences are compared from three aspects: training process, model input and loss function. Secondly, the NSTM with the best combination effect of word embedding and topic model was selected for experiments, and the influence mechanism of different word embeddings on the topic model was explored by changing the way and dimension of word embedding. Finally, document classification was used as a downstream task to compare the impact of the combination of each word embedding and topic model on document classification. The following main conclusions and strategies are formed:

i) Using word embeddings trained on large corpora provides the greatest performance improvement for topic models and document classification.

ii) The word embeddings trained by the Skip-gram model are suitable for long-text topic modeling (because long documents contain many low-frequency words), and the word embeddings trained by the GloVe model are suitable for short-text topic modeling (Word2Vec and fastText are not as good as GloVe for topic modeling due to insufficient context information).

iii) The word embeddings trained by the fastText model are not effective in the topic model, but the effect of document classification is better when combined with the topic model.

iv) Too small word embedding dimension is easy to cause underfitting, and too large word embedding dimension is easy to cause overfitting. Therefore, when using pre-trained word embeddings in the topic model, it is necessary to compare different dimensions and choose the better size.

v) In this study, we compared the training speeds of different word embedding models and it was observed that: i) No matter which training method is used, the training time cost increases with the increase of word embeddings dimension; ii) In the same dimension, CBOW model has the fastest training speed, and fastText model has the slowest training speed, which is about 15 times of CBOW model. These findings highlight the impact of different word embedding models on training speed and provide valuable insights for future research in this area. Overall, our study contributes to a better understanding of the factors influencing training efficiency in word embedding models and emphasizes the importance of considering these factors in model selection and implementation.

In addition to providing optimization strategies for selecting embedding methods and model training in topic modeling and document classification, our study also highlights the importance of considering the characteristics of the input data when selecting appropriate word embedding models. For example, our findings suggest that the Skip-gram model is suitable for long-text topic modeling, while the GloVe model is suitable for short-text topic modeling. This information could be useful for practitioners who are working with different types of text data. Furthermore, our study also sheds light on the impact of different word embedding models on training speed, which is an important consideration for practical applications. Practitioners with limited computational resources may need to choose a faster word embedding model, such as CBOW, to reduce training time, while those with more resources may opt for models with higher accuracy, such as fastText. Overall, our study provides valuable insights into the factors influencing training efficiency in word embedding models, and we hope that our findings will help guide future research and practical applications in this area.

Since word embedding is widely used, future work will explore the impact of different word embeddings on specific tasks, such as machine translation, question-answering systems, and more. The reason for this is that these tasks require the model to map input sequences to output sequences, which often necessitates understanding the semantics and context within the input sequence. Therefore, by studying these tasks, we hope to gain further insights into the strengths and weaknesses of different word vector representations in semantic understanding and provide better choices and optimization strategies for practical applications.

Acknowledgments

Our work was mainly supported by Fundamental Research Program of Shanxi Province (No. 202203021211305) and Shanxi Scholarship Council of China (2023-013), also supported by National Natural Science Foundation of China (62076156), the Scientific and Technological Innovation Programs of Higher Education Institutions in Shanxi (2021L004), Fundamental Research Program of Shanxi Province (202103021223023), National Social Science Fund in China (22BTQ098) and Youth Interdisciplinary Research Team of Humanities and Social Sciences (202205).

Availability of data and materials

20Newsgroups dataset (http://qwone.com/~jason/20Newsgroups/)
Web Snippets dataset (http://acube.di.unipi.it/dataset/)
Tag My News dataset (http://acube.di.unipi.it/tmn-dataset/)
pre-trained GloVe word embeddings (https://nlp.stanford.edu/projects/GloVe/)

References

Bansal, M., Gimpel, K., & Livescu, K. (2014). Tailoring continuous word representations for dependency parsing. Proceedings of the 52nd Annual *Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 809-815. https://doi.org/10.3115/v1/P14-2131

Cui, P., Hu, L., & Liu, Y. (2020). Enhancing extractive text summarization with topic-aware graph neural networks. *International Conference on Computational Linguistics,* 5360-5371. https://doi.org/10.18653/v1/2020.coling-main.468

Daniele, V., Paolo, F., & Ugo, S. (2012) Classification of short texts by deploying topical annotations. *European Conference on Information Retrieval,* 376-387. https://doi.org/10.1007/978-3-642-28997-2_32

David M., B., Andrew Y., N., Michael I., J., DM, B., & AY, N. (2003) Latent dirichlet allocation, *Journal of Machine Learning Research, 3,* 993–1022. https://doi.org/10.1162/jmlr.2003.3.4-5.993

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the Association for Information Science & Technology, 41*(6)*,* 391-407. https://doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9

Dieng, A. B., Ruiz, F. J. R., & Blei, D. M. (2020). Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics, 8*(2)*,* 439-453. https://doi.org/10.1162/tacl_a_00325

Firth, J. R. (1957). A synopsis of linguistic theory, 1930-1955. *Studies in Linguistic Analysis,* 168-205.

He, Z., Dinh, P., Viet, H., Trung, L., & Wray L., B. (2021) Neural topic model via optimal transport. *International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.2008.13537

Jeffrey, P., Richard, S., & Christopher D., M. (2014) Glove: Global Vectors for Word Representation. *Conference on Empirical Methods in Natural Language Processing,* 1532-1543. https://doi.org/10.3115/v1/D14-1162

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2017). Bag of tricks for efficient text classification. *Computing Research Repository,* 427-431. https://doi.org/10.18653/v1/E17-2068

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational bayes. *International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.1312.6114

Lang, K. (1995). Newsweeder: learning to filter netnews. *Machine Learning Proceedings*, 331-339. https://doi.org/10.5555/3091622.3091662

Law, J., Zhuo, H.H., He, J., Rong, E. (2018). LTSG: Latent topical skip-gram for mutually improving topic model and vector representations. In: Lai, JH., et al. *Pattern Recognition and Computer Vision. PRCV 2018. Lecture Notes in Computer Science, 11258. Springer, Cham.* https://doi.org/10.1007/978-3-030-03338-5_32

Li, C., Wang, H., Zhang, Z., Sun, A., & Ma, Z. (2016). Topic modeling for short texts with auxiliary word embeddings. *International ACM SIGIR Conference on Research and Development in Information Retrieval,* 165-174. https://doi.org/10.1145/2911451.2911499

Manning, C., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. *Cambridge: Cambridge University Press*. https://doi.org/10.1017/CBO9780511809071

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *Computer Science.* https://doi.org/10.48550/arXiv.1301.3781

Muntsa Padró, Idiart, M., Villavicencio, A., & Ramisch, C. (2014). Nothing like good old frequency: studying context filters for distributional thesauri. *Empirical Methods in Natural Language Processing,* 419-424. https://doi.org/10.3115/v1/D14-1047

Phan, X. H. , Nguyen, M. L. , & Horiguchi, S. (2008). Learning to classify short and sparse text & web with hidden topics from large-scale data collections. *Proceedings of the 17th International Conference on World Wide Web,* 91-100. https://doi.org/10.1145/1367497.1367510

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys, 34(1),* 1-47. https://doi.org/10.1145/505282.505283

Shi, B., Lam, W., Jameel, S., Schockaert, S., & Lai, K. P. (2017). Jointly learning word embeddings and latent topics. *International ACM SIGIR Conference on Research and Development in Information Retrieval,* 375–384. https://doi.org/10.1145/3077136.3080806

Tang, H., Li, M., & Jin, B. (2019). A topic augmented text generation model: joint learning of semantics and structural features. *Conference on Empirical Methods in Natural Language Processing, 19(1),* 5089-5098. https://doi.org/10.18653/v1/D19-1513

Tellex, S., Katz, B., Lin, J.J., Fernandes, A., & Marton, G.A. (2003). Quantitative evaluation of passage retrieval algorithms for question answering. *International ACM SIGIR Conference on Research and Development in Information Retrieval*, 41-47. https://doi.org/10.1145/860435.860445

Wang, L., Chris, D., Alan W., B., & Isabel, T. (2015) Two/too simple adaptations of Word2Vec for syntax problems. *North American Chapter of the Association for Computational Linguistics*, 1299-1304. https://doi.org/10.3115/v1/N15-1142

Xu, H., Wang, W., Liu, W., & Carin, L. (2018) Distilled Wasserstein Learning for Word Embedding and Topic Modeling. *Neural Information Processing Systems,* 1723-1732. https://doi.org/10.48550/arXiv.1809.04705