Received: 12 July 2019 | Revised: 20 October 2019 | Accepted: 24 October 2019

DOI: 10.1002/ett.3817

SPECIAL ISSUE ARTICLE

WILEY

Topic sentiment analysis using words embeddings dependency in edge social system

Wenhao Xie[1,2] | Xianghua Fu[1] | Xi Zhang[3] | Yu Lu[1] | Yanzhi Wei[1,2] | Jingying Yang[1,2]

[1]College of Big Data and Internet, Shenzhen Technology University, Shenzhen, China

[2]College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China

[3]Information Center, Shenzhen Technology University, Shenzhen, China

Correspondence
Xianghua Fu, College of Big Data and Internet, Shenzhen Technology University, Shenzhen 518118, China.
Email: fuxianghua@sztu.edu.cn

Present Address
Xianghua Fu, College of Big Data and Internet, Shenzhen Technology University, Shenzhen 518118, China.

Funding information
National Nature Science Foundation of China, Grant/Award Number: 61472258 and 61402294; National Key Technology Research and Development Program of the Ministry of Science and Technology of China, Grant/Award Number: 2014BAH28F05; Technology Planning Project of Guangdong Province, Grant/Award Number: 2014B010118005; Science and Technology Foundation of Shenzhen City, Grant/Award Number: JCYJ20140509172609162 and JCYJ20130329102032059

Abstract

The topic sentiment analysis is really fundamental in detecting potential cyber threats and cyber attacks in edge social systems. We can detect potential cyber threats and cyber attacks by identifying the sentiment orientation and topics of public opinion information in the edge social system. Topic sentiment joint model is an extended model, which aims to deal with the problem of detecting sentiments and topics simultaneously from the online comment. Most existing topic sentiment joint models ignore the dependency among words so that they lose rich semantic information and the resulting distribution might be not very satisfactory. In this paper, we propose a novel topic sentiment joint model with word embeddings dependency based on recurrent neural network. The model introduces the dependency among word embedding and delivers topic information and sentiment information of words by a recurrent neural network. It fully extends the semantic information and redefines the topic sentiment-word distribution. Moreover, we obtain more accurate topic detection and sentiment analysis. Experimental results on online review data set show that the proposed model significantly improves the sentiment classification accuracy and achieved better topic detection compared with previous methods.

1 | INTRODUCTION

In recent years, social media becomes extremely popular in China. People can more frequently and freely to express their attitudes, exchange their opinions, and share their life experience on social media sites such as blogs, microblog, forum, and bulletin board system. These sites are often the best sources of up-to-the-minute information, and quickly and broadly disseminate news and memes across both real world events and cultural trends. In China, WeChat is the most popular social networking application, and it can generate tens of millions of words per second and the amount of public opinion information varies greatly in different times. Figure 1 shows the sentiment orientation of the sixth national cyber security public opinion. We could clearly see that the amount of public opinion information on WeChat fluctuates significantly from day to day. Relevant public opinion information reached its peak on September 3, 2019 and reached its subpeak on

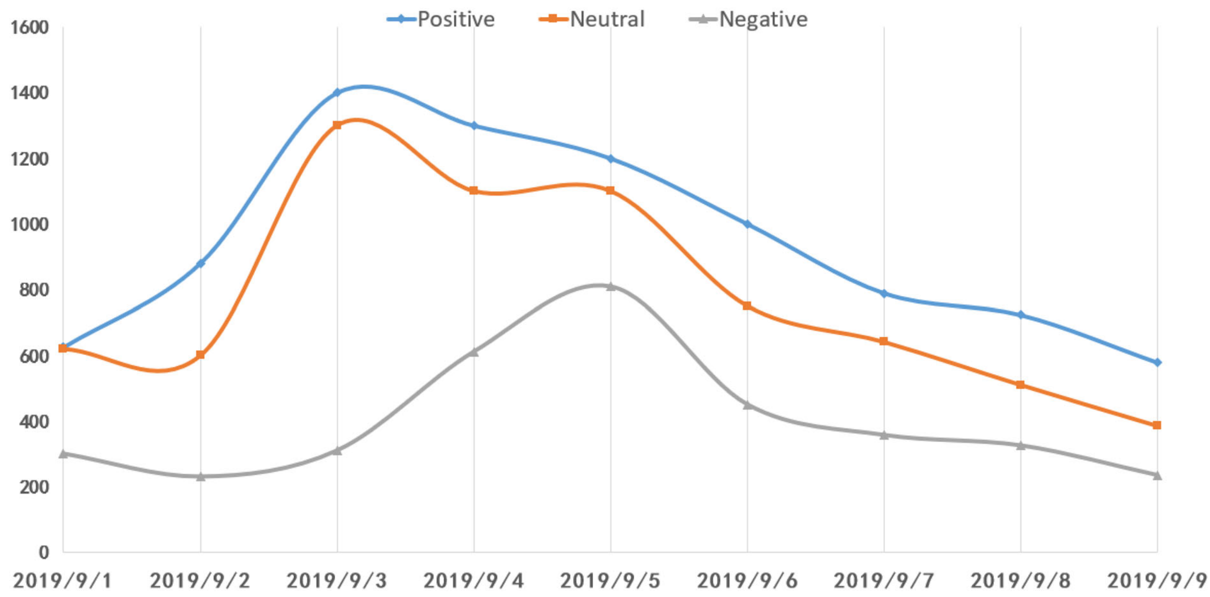

FIGURE 1 Chinese cyber security public opinion sentiment orientation

the 4th. The overall sentiment tendency of China's public opinion information is also shown in Figure 1. Although its overall sentiment orientation is positive, there are still some negative public opinion information. This negative information probably involves some sensitive information and misinformation. However, the spread of commercially sensitive information, misinformation, and malicious remarks in social media may cause significant economic losses and the severe damage to the society. It is well known that it is every government's responsibility to find out as much as negative or fake news that jeopardize social security, so as to prevent tragedy from happening as soon as possible, and finally achieve the maintenance of social security order, strike crimes, protect the people's life property safety, and other legal right. For example, we detected a text information from someone's WeChat, “We are going to hijack a child at the Guangzhou Railway Station at 4pm tomorrow.” If we use the topic sentiment joint model to detect that this text information is offensive, we could prevent a terrorist attack in advance. Therefore, the text information in the network relate to the security of the network and the security of the country. In other words, detecting and analyzing public opinion information in edge social systems, and then predicting the public's sentiment tendencies, is of great significance to cyber security and national security.

There are various public opinion monitoring tasks[1-4] and one of them is sentiment classification. In general, it refers to whether the semantic orientation of a text is positive, negative or neutral. For applying machine learning to sentiment classification, most existing approaches rely on supervised learning models trained from labeled corpora where each document has been labeled as positive or negative prior to training. Such labeled corpora can not be always obtained easily in practical applications. Moreover, sentiment classification models trained on one domain might not work at all when the domain is changed to another. Furthermore, in a more fine-grained sentiment classification problem (eg finding users' opinions for a particular product feature), topic/feature detection and sentiment classification are often performed in a two-stage pipeline process, by first detecting a topic/feature and later assigning a sentiment label to that particular topic.

Intuitively, sentiment polarities are dependent on topics or domains. Therefore, detecting both topic and sentiment simultaneously should play an important role in helping users in terms of opinion mining and summarization. For instance, though the adjective “unpredictable” in a phrase such as “unpredictable steering” may have negative orientation in an automobile review, it could also have positive orientation in a phrase like “unpredictable plot” in a movie review.

Topic modeling and sentiment analysis are two central tasks that deal with textual data. In public opinion monitoring task, most of the previous methods deal with topic identification and sentiment analysis, respectively.[5-7] However, this ignores the relationship between topics and sentiments. More explicitly, the former deals with extracting topics (what is it about?), while the latter is about sentiment and opinion classification (what is the underlying opinion?). In general, these two tasks are complementary to the extent that sentiments are usually issued about topics and topics are often the basis of subjective positions.

Topic and sentiment modeling tasks are largely related to the extent that sentiments are usually issued about a topic at hand. In order to model topic-sentiment conjunction, numerous works make use of topic models, which are statistical models for discovering low-dimensional structures (topics) from text based on word co-occurrence patterns. These patterns are captured using the so-called latent or hidden variables. Earlier topic models, like latent Dirichlet allocation (LDA)[8] and probabilistic latent semantic index,[9] have mainly focused on extracting homogeneous topics but more recently, these models have been extended to capture other aspects of text, like sentiment. For instance, joint sentiment topic model (JST)[10] has been developed for topic extraction under different sentiment labels. This is performed by extending LDA[8] with a new sentiment layer inserted before the topic layer. Thus, to generate a word for a document, a sentiment label s is obtained first, then a topic is obtained conditioned on s. Reverse-JST[11] is a variant of JST where the order of sentiment and topic layers is inverted.

In this paper, we propose a novel topic sentiment joint model with word embeddings dependency based on recurrent neural network (TSJM-WED). Our model use recurrent neural networks to capture the topic and sentiment information and fully extend the semantic information in the text. Compared with other models, the proposed model, which is a significant improvement, can more accurately predict the sentiment orientation expressed by the textual information. Therefore, in the edge social system, our proposed model can more accurately identify text information with negative sentiment and reactionary topic, thus indirectly maintaining network security and national security.

2 | RELATED WORK

Recently, how to accurately monitor public opinion information[1-6] in large-scale social media has become an important research hotspot of cyber security and cyber attacks. As a result, there have been many attempts to develop text mining techniques for accurately detecting topics and sentiments from large volumes of social contents.[7,12-16] Topic identification and sentiment analysis are key research issues in security monitoring of public opinion. For a public opinion monitoring task, most of the previous methods deal with topic identification and sentiment analysis, respectively. However, in practice, sometimes, we need to determine the sentiment tendency of the public under the multiview topic, and identify the topic and sentiment at the same time. Therefore, this is the reason topics and sentiments should be jointly extracted and analyzed. Therefore, a few of researchers have proposed a method of mixed analysis of topics and sentiments. The basic principle of such method is to add sentiment markers based on topic probability models while identifying sentiments related to the topic.

Several works extending probabilistic topic model[8,17-20] have been designed to tackle the problem of the joint extraction of sentiments and latent topics from documents in the recent years.[10,11,21-28] The JST model[10] extended LDA[8] to a four-layer model by adding an additional sentiment layer between the document and the topic layer. Topic sentiment mixture (TSM)[21] jointly modeled topics and sentiments in the corpus on the basis of the probabilistic latent semantic index. Huang et al[28] proposed a multimodal JST (MJST) model for weakly supervised sentiment analysis in microblogging, which applies LDA to simultaneously analyze sentiment and topic hidden in messages based the introduction of emoticons and microbloggers personality. Fu et al[22] introduced extra word embedded into JST to extend semantic information and proposed a topic sentiment joint model with word embeddings (TSWE). The unsupervised topic-sentiment joint probabilistic model UTSJ[19] first employed Gibbs sampling algorithm to approximate parameters of maximum likelihood function offline and obtained topic-sentiment joint probabilistic distribution vector for each review. TDAM[27] assumed that a global topic embedding is shared across documents and employs an attention mechanism to derive local topic embedding for words and sentences. The word-pair sentiment-topic model (WSTM)[20] represented the whole reviews corpus as a bag of word-pairs, and effectively captures sentiment and topic information that are implicit in words co-occurrence pattern by modeling the generative process of the word-pair set. However, these pure TSM models are difficult to obtain the dependency between topics and sentiments and the dependency among topics and sentiments in the context. These sentiment and topic distributions might be not very satisfactory results.

Li et al[17] considered the dependency between topics and sentiments and the dependency among sentiments in context. Sentiment-LDA model and dependency-sentiment-LDA model (DSLDA) are proposed, which decide the source of sentiment label using the value of transfer variable. The DSLDA model did not consider the dependency among topics. Fu et al[1] proposed a novel topic sentiment joint model called weakly supervised TSWE (WS-TSWE), for which word embeddings and HowNet lexicon were incorporated simultaneously to improve the topic identification and sentiment recognition. Liu et al[23] proposed a novel probabilistic generative model (DTSA) to extract topics and the specified sentiments from news streams and analyzed their evolution over time simultaneously. Rahman and Wang[24] took the dependency between topics and sentiments into account and proposed latent TSM model (HTSM). The model assumed that a sentence has a topic and ignored the change of topics and sentiments among words.

Most recently, deep neural networks showed that they have very good performance in a broad range of natural language processing tasks.[29-35] This is especially the case with recurrent neural networks, such as long short-term memory (LSTM)[36] and gated recurrent unit (GRU).[37] These models have more powerful strength keeping the sequence information over time and capturing the long-range dependency in text. Numerous works utilized these models to carry out language and sentiment analysis. Tang et al[38] used LSTM and convolution neural networks to represent text and then execute sentiment classification. Tian et al[39] introduced LSTM into the probabilistic topic model to propose a sentence-level topic model and sentences can be generated when topics are given.[40] However, these models pay little attention to topic sentiment model so far.

In this paper, we propose a topic sentiment model (TSJM-WED) that incorporates word embedded dependency by a recurrent neural network, and combined with topic vectors under the sentiments obtained from the TSWE model. As a result, we can deliver semantic information of a word to the next another word, and then deliver the topic and sentiment information. In general, we consider the dependency among word embedding to fully extend the semantic information, and can do better topic detection and sentiment classification simultaneously. Therefore, we are able to detect the security problems in the edge social system through more reliable sentiment analysis and topic detection technology, and prevent the network attack in advance.

2.1 | Recurrent neural networks

Recurrent neural networks, such as LSTM[36] and GRU,[37] have a powerful ability to keep the sequence information over time and capture the long-range dependency in the text. These models can be utilized to perform language and sentiment analysis.

2.1.1 | Long short-term memory

The principle of LSTM[36] lies in that it introduces a special memory cell that can solve the vanishing gradient problem. The memory cell is composed of three gates and a cell unit. Gates generally use sigmoid activation function and the cell unit often uses activation function tanh transformation. The output values of a sigmoid function are between 0 and 1, where 1 means completely carrying previous information and 0 means completely abandoning it. The LSTM adds or removes information to update the cell state through gates. A diagram of the LSTM model is shown in Figure 2.

Equations of LSTM are expressed as follows.

Input gate is

$$i_t = \sigma\left(w_x^i x_t + w_h^i h_{t-1} + w_c^i c_{t-1} + b_i\right). \tag{1}$$

Forget gate is

$$f_t = \sigma\left(w_x^f x_t + w_h^f h_{t-1} + w_c^f c_{t-1} + b_f\right). \tag{2}$$

Output gate is

$$o_t = \sigma\left(w_x^o x_t + w_h^o h_{t-1} + w_c^o c_{t-1} + b_o\right). \tag{3}$$

Update cell state is

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh\left(w_x^c x_t + w_h^c h_{t-1} + b_c\right) \tag{4}$$

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

where x_t is the input at the current time step t (such as one word in a sentence), σ denotes the logistic sigmoid function, and $\odot$ denotes element wise multiplication. The superscript of the weight matrices and biases indicate what they are used for. For example, w_x^i is a matrix of input. The input gate i_t represents new information we are going to input. i_t multiplied by a tanh layer decides final new information added to the cell state. The forget gate f_t decides which information we are going to throw away from the cell state. f_t multiplied by the last step cell state c_{t-1} filters discarded information. New

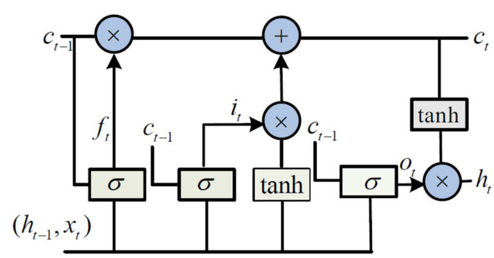

FIGURE 2 Graphical representation of LSTM model

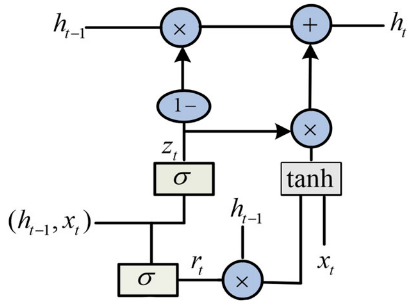

FIGURE 3 Graphical representation of GRU model

cell state c_t is obtained by adding these two items. Finally, the cells state c_t is processed through a tanh layer, and then multiplied by the output gate o_t to decide which sections will be output.

2.1.2 | Gated recurrent unit

The GRU[37] is a variant of LSTM. The recurrent neural network has an update gate and output gate, so the structure is relatively simple. The structure of GRU model is shown in Figure 3.

Equations of LSTM are expressed as follows.

Input gate is

$$r_t = \sigma\left(w_x^r x_t + w_h^r h_{t-1} + b_r\right). \tag{6}$$

Output gate is

$$z_t = \sigma\left(w_x^z x_t + w_h^z h_{t-1} + b_z\right). \tag{7}$$

The candidate value of state is

$$u_t = \tanh\left(w_x^u x_t + w_h^u (r_t \odot h_{t-1}) + b_u\right). \tag{8}$$

Update state is

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot u_t. \tag{9}$$

The define of the tokens mentioned above is similar to the LSTM model in Section 2.1.1.

2.2 | Topic sentiment joint model with word embedding

The TSWE[22] is formed by taking the original topic sentiment model (JST)[10] and replacing their Dirichlet multinomial component with a two component mixture of a sentiment-topic-to-word Dirichlet multinomial component and a word embedding component. The structure of TSWE model is shown in Figure 4.

Assume that we have a corpus with a collection of D documents denoted by $C = \{d_1, d_2, \ldots, d_D\}$; each document in the corpus is a sequence of N_d words denoted by $d = (w_1, w_1, \ldots, w_{N_d})$, and each word in the document is an item from a vocabulary index with V distinct terms denoted by $\{1, 2, \ldots, V\}$. Moreover, let L be the number of distinct sentiment labels, and K be the total number of topics. The procedure of generating a word w_i in document d consists of three stages. First, one chooses a sentiment label l from the document specific sentiment distribution π_d. Second, one chooses a topic randomly from the topic distribution $\theta_{d,l}$, where $\theta_{d,l}$ is chosen conditioned on the sentiment label l. Finally, one draws a word from distribution over words defined by the topic and sentiment label, which is different from JST[10] that a word is sampled from the word distribution only defined by topic.

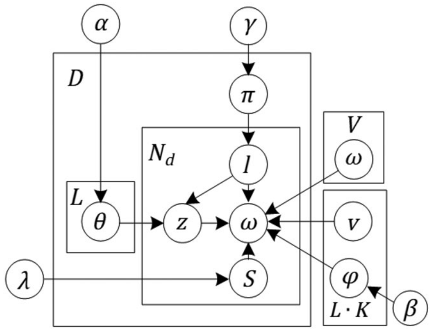

FIGURE 4 Graphical representation of TSWE model

The hyperparameters α and β in TSWE can be treated as the prior observation counting the number of times topic j associated with sentiment label l sampled from a document and the number of times words sampled from topic j associated with sentiment label l respectively, before any actual words are being observed. Similarly, the hyperparameter γ can be interpreted as the prior observation counting the number of times sentiment label l sampled from document before any words from the corpus is observed. In TSWE, there are three sets of latent variables,[22] ie, the joint sentiment/topic-document distribution θ, the joint sentiment/topic-word distribution φ, and the sentiment-document distribution π.

The formal definition of the generative process of TSWE model is as follows.

For each of sentiment-topic pair (l, z)

- generate the word distribution of the sentiment-topic pair $\phi_{l,k} \sim Dir(\beta)$.

For each document d

- draw a document-sentiment distribution $\pi_d \sim Dir(\gamma)$.

For each sentiment label l under document d

- draw a sentiment-topic distribution $\theta_{d,l} \sim Dir(\alpha)$.

For each word w_i in document d

- draw a sentiment label $l_i \sim Mul(\pi_d)$.
- draw a topic $z_i \sim Mul(\theta_{d,l})$.
- draw a binary indicator variable $s_i \sim Ber(\lambda)$.
- draw a word $w_i \sim (1 - s_i)\mathrm{Mul}(\varphi_{z_i}) + s_i \mathrm{Mul}T(v_{z_i} w^T)$.

The TSWE model has the structure of the original JST[10] model, with two additional matrices v and ω of embeddings weight, where v_k and ω_i are the embeddings representations associated with sentiment-topic k and word i, respectively. We also add a binary indicator variable s_i obeying Bernoulli distribution, to determine whether the word ω_i is to be generated by the Dirichlet multinomial or the word embeddings component. Our model defines the probability that a word is generated from embeddings component as the multinomial distribution $MulT$ with

$$\mathrm{MulT}\left(w_i | v_k w^T\right) = \frac{\exp(v_k \cdot w_{w_i})}{\sum_{w_i' \epsilon w} \exp\left(v_k \cdot w_{w_i'}\right)}. \tag{10}$$

$MulT$ is a multinomial distribution with log-space parameters. ω is pretrained word embeddings learned from an external big corpus, which is fixed. ω_i and v_k are word and topic embeddings weights. To approximate the topic embeddings v_k, we apply to the regularized maximum likelihood estimation. Learning loglinear models for topic models with MAP estimation is also used in model.[41,42] The negative log-likelihood L according to TSWE model factorizes topic-wise into factors L_k for each topic associated with the sentiment. With L_2 regularization for each topic z_k under sentiment, we set L_2 to 0.01. N^{k,ω_i} denotes the number of occurrences of word ω_i assigned with topic k and sentiment label l, we derive

$$L_k = \mu \|v_k\|_2^2 - \sum_{w_i \epsilon w} N^{k,w_i} \left(v_k w_{w_i} - log \left(\sum_{w_i' \epsilon w} exp\left(v_k w_{w_i'}\right) \right) \right). \tag{11}$$

We obtain the MAP estimation of topic vectors v_k by minimizing the regularized negative log likelihood. The derivative with respect to the mth element of the embeddings for topic k is

$$\frac{\partial L_k}{\partial v_{k,m}} = 2\mu v_{k,m} - \sum_{w_i \epsilon w} N^{k,w_i} \left(w_{w_i,m} - \sum_{w_i' \epsilon w} w'_{w_i,m} \mathrm{MulT}\left(w_i' | v_k^T\right) \right). \tag{12}$$

Then, the model applies L-BFGS implementation[43] from the Mallet toolkit[44] to derive the topic vector v_k that minimizes L_k. The Gibbs sampling algorithm[45] is introduced into the TSWE model. The detailed derivation process on Gibbs sampling for the topic model can refer to the work of Robert and Casella.[46]

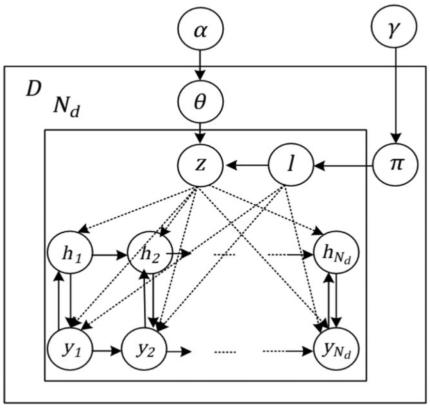

FIGURE 5 Graphical representation of TSJM-WED model

3 | TOPIC SENTIMENT JOINT MODEL WITH WORD EMBEDDED DEPENDENCY

The TSWE model mentioned before introduces extra word embedded into JST to extend semantic information to perform topic detection and sentiment classification simultaneously. However, it ignores the dependency among words. To overcome this limit, we propose the TSJM-WED. The TSJM-WED model not only uses the word embedding trained on the external corpus but also considers the dependency among words to extend the semantic information. It utilizes recurrent neural networks LSTM and GRU for training purposes because it can remember the dependency among words. It delivers semantic information of a word to the next word and then delivers the topic and sentiment information. The dependency is obtained from the training of word embedding and topic vectors in the recurrent neural network. The topic sentiment model with word embedded dependency whose dependency among words obtained from LSTM model is referred to as TSJM-WED-L while when that obtained from GRU model is referred to as TSJM-WED-G. The structure of TSJM-WED model is shown in Figure 5.

The corpus D in the TSJM-WED model shown in Figure 5. contains m documents and $D = \{d_1, d_2, \dots, d_m\}$. Each document contains N_d words, such as $d_i = \{y_1, y_2, \dots, y_{N_d}\}$. $\{h_1, h_2, \dots, h_{N_d}\}$ are the hidden layers. π is the document-sentiment distribution and it obeys the Dirichlet distribution with γ as the parameter. l is the sentiment label. θ is the sentiment distribution under document-sentiment and it follows the Dirichlet distribution with α as the parameter. The hyperparameter γ can be interpreted as the prior observation counting the number of times sentiment label l sampled from document before any words from the corpus is observed. In our model (TSJM-WED), the process of each word in a document is as follows.

1) For each document d
   draw a document-sentiment distribution $\pi_d \sim Dir(\gamma)$.
2) For each sentiment label l under document d
   draw a sentiment-topic distribution $\theta_{d,l} \sim Dir(\alpha)$.
3) For each word x_t in the document d

   i) draw a sentiment label $l \sim Mul(\pi_d)$.
   ii) draw a topic $z \sim Mul(\theta_{d,l})$.
   iii) calculate the hidden layer of the recurrent neural networks $h_t = f(h_{t-1}; x_t; z)$
   iv) generate word $wp(w|x_1, \dots, x_t; z; l) \infty g(w'; h_{t-1}; x_t; z; l)$.

The hidden layer h_t is calculated by the improved recurrent neural network variant LSTM and GRU.

The formulas that LSTM calculates the hidden layer h_t are as follows:

$$i_t = \sigma\left(W_x^i x_t + W_z^i z_t + W_h^i h_{t-1} + W_c^i c_{t-1} + b_i\right) \tag{13}$$

$$f_t = \sigma\left(W_x^f x_t + W_z^f z_t + W_h^f h_{t-1} + W_c^f c_{t-1} + b_f\right) \tag{14}$$

$$o_t = \sigma\left(W_x^o x_t + W_z^o z_t + W_h^o h_{t-1} + W_c^o c_{t-1} + b_o\right) \tag{15}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot tanh\left(W_x^c x_t + W_z^c z_t + W_h^c h_{t-1} + b_c\right) \tag{16}$$

$$h_t = o_t \cdot tanh(c_t). \tag{17}$$

The formulas that GRU calculates the hidden layer h_t are as follows:

$$r_t = \sigma\left(W_x^r x_t + W_z^r z_t + W_h^r h_{t-1} + b_r\right) \tag{18}$$

$$z_t = \sigma\left(W_x^z x_t + W_z^z z_t + W_h^z h_{t-1} + b_z\right) \tag{19}$$

$$u_t = tanh\left(W_x^u x_t + W_z^u z_t + W_h^u (r_t \cdot h_{t-1}) + b_z\right) \tag{20}$$

$$h_t = z_t \cdot h_{t-1} + (1 - z_t) \cdot u_t. \tag{21}$$

x_t is the word embedding of the input word and word embedded are obtained by training the external large corpus with Word2vec model. The superscript of the weight matrices and biases indicate what they are used for. For example, w_x^i is an initial weight matrix of the input. h_{t-1} are hidden layer states and they deliver the sequence information and topic relevant information of words. z_t is the topic vector of the input word and the topic vectors are obtained by maximizing the formula

$$L_t = \mu\|m_t\|_2^2 - \sum_{x_i \epsilon W} N^{t,x_i}\left(m_t\omega_{x_i} - log\left(\sum_{x_i' \epsilon W} exp\left(m_t\omega_i'\right)\right)\right). \tag{22}$$

ω is pretrained word embeddings learned from an external big corpus. m_t and x_i are the embeddings representations associated with sentiment-topic t and word i, respectively. The negative log likelihood L in our model factorizes topic-wise into factors L_t for each topic associated with sentiment. With L_2 regularization for each topic z_t under sentiment, we set L_2 to 0.01. N^{t,x_i} denotes the number of occurrences of word x_i assigned with topic t and sentiment label l. Finally, given the word, sentiment label and the topic, we use softmax function to predict the distribution of the next word and the formula is as shown in (23)

$$p(w|x_1, \ldots, x_t; z; l) = softmax\left(v_t \cdot \left(W_z^y z_t + W_h^y h_t + b_y\right)\right). \tag{23}$$

v_t is the word embedding of the corresponding word. W_*^* is the weight and b_* is the offset term. The softmax function is as follows:

$$p(i) = \frac{exp(x_i)}{\sum_{i=1}^{v} exp\left(x_i'\right)}. \tag{24}$$

In this formula, v is the size of the dictionary. The object function of the model uses the negative logarithmic likelihood loss function. The formula is as follows:

$$L = -\frac{1}{m}\sum_{i=1}^{m} y_i log(p(w|x_1, \ldots x_t; z; l)). \tag{25}$$

y_i is the sentiment label of the input text and m is the number of input text. The negative log-likelihood becomes unhappy at smaller values, where it can reach infinite unhappiness, and becomes less unhappy at larger values. That is, the negative logarithmic likelihood loss function reaches infinity when input is 0, and reaches 0 when input is 1. Because we are summing the loss function to all the correct classes, what is actually happening is that, whenever the network assigns high confidence at the correct class, the unhappiness is low, but when the network assigns low confidence at the correct class, the unhappiness is high.

4 | EXPERIMENTS

In this section, we evaluate the performance of TSJM-WED model on sentiment classification and topic detection based on both English and Chinese data sets. The performance of the sentiment classification is measured based on the probability of a sentiment label given a document. We only take into account the probability of positive and negative label for a given document in the experiment, with the neutral label probability being ignored. Therefore, we define that a document is classified to be a positive-sentiment document if the probability of a positive sentiment label is larger than that of negative sentiment label for a given document, and vice versa.[47] Topic detection presents a word generated from the per-corpus word distribution conditioned on both topics and sentiment labels.

In order to show that two models TSJM-WED-L and TSJM-WED-G can perform sentiment classification and topic detection simultaneously, we compared these two models with JST model,[10] DSLDA model,[17] HTSM model,[24] DSTA

Data set	Doc	Average doc length	V
CR	4000	11	3776
HR	4000	27	8702
BR	3975	18	2473
MR04	1385	183	4365

TABLE 1 The experimental data set

model,[23] UTSJ model,[19] MJST model,[28] WSTM model,[20] TDAM model,[27] WS-TSWE model,[1] and TSWE model[22] proposed in Section 1. We used accuracy rate to evaluate the ability of text sentiment analysis, and the performance of topic detection is evaluated by perplexity[48] and normalized mutual information (NMI).[48] In addition, we also list the words under the sentiments and topics to evaluate the accuracy of the sentiment and topic extraction.

4.1 | Data set

4.1.1 | External data set

We used the Chinese Wikipedia and English Wikipedia corpus as external information to train the initial word embedded. Wikipedia is the largest encyclopedia to date, which is open, web-oriented, and multilingual. Documents in Wikipedia are clearly organized by topics, thus making it very suitable for training our word embedded.

The Chinese Wikipedia consists of 777 961 documents, about 998.5 MB tokens. The English Wikipedia consists of 2 million documents, about 1.7 GB file size. We train the word vectors by using the Google Word2Vec toolkit.[49,50] In this paper, we define the dimension of word embeddings is 300 as in the work of Vargas-Calderón and Camargo.[49,50]

4.1.2 | Experimental data sets

We conduct experiments on two language data sets, Chinese and English, to demonstrate the effectiveness of our TSJM-WED model. The Chinese data set* consists of the product reviews of the computer (CR), the product reviews of the book (BR), and the product reviews of the hotel (HR). They all contain 2000 positive and 2000 negative examples. The English data set† is the MR04 data set, which was introduced by Pang and Lee in 2004, consisting of 1000 positive and 1000 negative movie reviews.

Our experimental data sets contain Chinese and English. The data processing includes word segment and filtering out high/low frequency words, stop words, improper characters, and single words. Empirically, if the word occurs less than 5 times in the document, we regard it as a low-frequency word. If the word's occurrence frequency appears more than 20% of the total tokens, then it is treated as a high-frequency word. For the Chinese corpus, we choose the ICTCLAS2016 as our word segment system, which can not only support the user dictionary but also find new words. The English data sets do not need the word segment processing. We also convert all characters to lower case and remove nonalphabetic characters. After this preprocessing, the distribution of all the experimental data sets is shown in Table 1.

4.2 | Parameter setting

The hyperparameter β is set to 0.01 in the TSJM-WED model, which is a common setting as noted by Griffiths and Steyvers.[47] α is set to a standard setting $\alpha = \frac{50}{k}$. The hyperparameter γ is set to $\frac{0.05 \cdot A}{L}$ according to the work of Griffiths and Steyvers.[47] The is the average length of documents and is the total number of sentiment labels.

4.3 | Experimental results and analysis

In this section, we have conducted a set of experiments on TSJM-WED-L and TSJM-WED-G model. We have carried out the experiment from the sentiment classification and topic detection tasks. We have also compared our models with 10 basic models mentioned before to illustrate better performance of our proposed model TSJM-WED-L and TSJM-WED-G. In addition, we have conducted two sets of ablation experiments to explore the effectiveness of our proposed method. In the first set, we have performed the experiment that removes the word embedding on the TSJM-WED model. We denote these two models as TSJM-WED-L′ and TSJM-WED-G′. The words are trained completely based the Dirichlet multinomial component without word embeddings. In the second set, we have removed the RNN model from the TSJM-WED model. We denote this model as W-TSJM-WED.

*https://github.com/LeSamouraiXWH/Chinese-Comment-Data/
†www.cs.cornell.edu/people/pabo/movie-review-data/

4.3.1 | Sentiment classification evaluation

In the sentiment classification evaluation, we use the metric accuracy to evaluate the sentiment classification. If a model's accuracy is larger than some other models, it shows that its result of sentiment classification is better than that of those models. In order to fully verify the validity of the model results, we conduct the experiment with sentiment classification on CR, HR, BR, and MR04. The TSJM-WED-L and TSJM-WED-G models are compared with JST, DSLDA, HTSM, DTSA, MJST, WSTM, UTSJ, TDAM, WS-TSWE, and TSWE models. Table 2 lists the sentiment classification accuracy of different models under different topic numbers with respect to four different data sets.

In Figure 6, it is intuitive to show that, when we perform the sentiment analysis for online review data sets, the accuracies of both TSJM-WED-L and TSJM-WED-G models are higher than that of the JST, TSWE, DSLDA, DTSA, MJST, WSTM, UTSJ, TDAM, WS-TSWE, and HTSM models, which considers dependency. Furthermore, the accuracy of the TSJM-WED-L model is slightly higher than that of the TSJM-WED-G model on most data sets. For example, for the experiment of Chinese computer review data set CR, when we set the number of topics as 8, the accuracy rates of TSJM-WED-L and TSJM-WED-G model are higher approximately 1% to 5% than that of the other 10 models. The accuracy rate of TSJM-WED-G model is 79.45% and the accuracy rate of TSJM-WED-L model is 79.92%. While, according to the experimental results when using the English MR04 data set, the accuracy rates of TSJM-WED-L and TSJM-WED-G model are higher approximately 0.5% to 7% than that of the other 10 models. The accuracy rate of the TSJM-WED-L model is 84.91% and the TSJM-WED-G model is 84.23%. Regarding the ablation experiment results, performance of both TSJM-WED-L and TSJM-WED-G models are significantly improved compared with the W-TSJM-WED model. However, the TSJM-WED-L model and the TSJM-WED-G model have only slightly improved compared with the TSJM-WED-L′ and TSJM-WED-G′ models, respectively, based on the majority of the data sets, sometimes even degrades performance. In general, the RNN models have more improvement than word embeddings. Although the word embeddings contain sentiment and topic contextual information, it neglects the dependency characteristics among words. The RNN models can capture the dependency characteristics among words to enrich the semantic information. In addition, the results between TSJM-WED-L model and the TSJM-WED-G model are not very different. They can both store the sequence information among words in the document through various kinds of gates and make full use of the semantic information of the dependency among words. It combines the sentiment information and topic information of the words, and then improves the sentiment classification accuracy of the model. This fully illustrates that considering the dependency among words can be better to express document features in the topic sentiment joint analysis. Therefore, we can more accurately detect the opinions and sentiments expressed by the masses in the edge social system, so as to better maintain network security and prevent network attacks in advance.

4.3.2 | Topic detection evaluation

Another task is to identify topics and sentiment from the data sets, and evaluate the effectiveness of sentiment topic captured by these models. Unlike the sentiment identification task, the topic-sentiment identification evaluates the word distribution over topics under positive and negative sentiment label for each document. Therefore, we need to evaluate the topic clustering performance under the corresponding sentiment polarity.

We use the perplexity and NMI[48] to evaluate topic detection. The perplexity traditionally used in language modeling is monotonically decreasing in the likelihood of the test data sets. Lower perplexity scores reflect better generalization performance. More formally, for a test set of D documents, the perplexity function is as follows:

$$\text{perplexity}(D) = \exp\left(-\frac{\sum_{d=1}^{D}\log(p(w_d))}{\sum_{d=1}^{D}N_d}\right) \tag{26}$$

$$p(w_d) = \prod_{n=1}^{N}\sum_{z} p(z|d)\times p(w_n|z). \tag{27}$$

According to Equation (26), N_d is the total number of words in each document in the test set. $p(w_d)$ refers to the probability of occurrence of each document in the test set. According to Equation (27), $p(z|d)$ represents the probability of occurrence of each topic in a document, and $p(w_n|z)$ represents the probability that each word in the dictionary appears under a certain topic. In terms of perplexity, we use the CR data set to carry out the training of topic sentiment joint analysis on the JST, TSWE, DSLDA, HTSM, DTSA, MJST, WSTM, UTSJ, TDAM, WS-TSWE, TSJM-WED-L, and TSJM-WED-G models. The perplexity of every model on CR data set is shown in Figure 7.

TABLE 2 The accuracy of sentiment classification of different models under different topic numbers

Data set	Model	T = 8 Acc(%)	T = 20 Acc(%)	T = 40 Acc(%)	T = 80 Acc(%)
CR	JST	74.11	71.51	72.36	73.32
	DSLDA	76.05	78.25	77.36	77.67
	HTSM	77.93	78.67	77.95	77.86
	TSWE	78.21	78.76	79.15	78.36
	DTSA	78.81	78.23	79.14	77.81
	MJST	78.31	77.65	79.08	78.55
	WSTM	78.43	78.37	79.09	77.87
	TDAM	77.67	78.02	79.11	78.16
	UTSJ	78.21	77.66	79.01	78.36
	WS-TSWE	78.55	78.38	79.25	78.98
	W-TSJM-WED	77.36	76.88	78.35	77.98
	TSJM-WED-L′	79.85	78.34	79.89	80.59
	TSJM-WED-G′	78.34	**78.88**	**79.75**	79.23
	TSJM-WED-L	**79.92**	**78.98**	**80.15**	**80.97**
	TSJM-WED-G	**79.45**	78.56	79.69	**79.58**
HR	JST	73.33	72.31	72.16	67.85
	DSLDA	72.02	73.87	73.51	74.23
	HTSM	72.77	74.35	73.25	71.26
	TSWE	76.61	75.86	75.94	76.67
	DTSA	77.02	75.89	76.74	76.83
	MJST	77.22	75.44	76.56	76.28
	WSTM	77.09	76.21	76.11	76.06
	TDAM	77.13	76.95	76.12	76.81
	UTSJ	76.89	75.92	76.32	76.23
	WS-TSWE	77.24	75.89	**76.79**	77.08
	W-TSJM-WED	75.58	75.23	75.35	75.89
	TSJM-WED-L′	77.34	76.05	76.95	78.54
	TSJM-WED-G′	76.98	76.54	75.68	77.23
	TSJM-WED-L	**77.73**	**76.32**	**77.02**	**78.95**
	TSJM-WED-G	**77.68**	**76.95**	75.86	**77.56**
BR	JST	66.56	62.36	66.54	65.39
	DSLDA	67.51	67.49	66.12	66.39
	HTSM	67.99	71.23	70.36	68.37
	TSWE	74.28	72.31	72.11	69.12
	DTSA	74.13	72.21	71.45	70.11
	MJST	73.56	72.24	71.29	70.13
	WSTM	74.88	72.85	72.17	70.13
	TDAM	73.56	72.33	71.23	69.56
	UTSJ	74.12	72.13	72.05	70.89
	WS-TSWE	74.29	72.38	72.31	70.85
	W-TSJM-WED	72.87	71.03	71.68	69.23
	TSJM-WED-L′	73.88	73.07	71.87	70.98
	TSJM-WED-G′	74.87	72.84	72.83	71.45
	TSJM-WED-L	**74.69**	**73.65**	**72.85**	**71.23**
	TSJM-WED-G	**74.93**	**72.95**	**73.02**	**71.95**
MR04	JST	71.21	71.69	62.39	56.89
	DSLDA	82.67	81.36	80.95	79.69
	HTSM	78.53	81.26	79.56	78.13
	TSWE	82.32	84.32	82.41	78.23
	DTSA	81.47	83.14	82.31	79.24
	MJST	82.13	83.23	82.11	79.14
	WSTM	82.76	82.15	82.32	79.43
	TDAM	82.44	83.45	82.65	78.26
	UTSJ	82.98	82.74	81.47	79.62
	WS-TSWE	83.88	83.05	83.04	79.97
	W-TSJM-WED	82.16	82.12	81.76	79.35
	TSJM-WED-L′	**84.98**	83.75	83.26	81.03
	TSJM-WED-G′	84.05	83.46	82.95	80.22
	TSJM-WED-L	84.91	**83.95**	**83.69**	**81.25**
	TSJM-WED-G	**84.23**	**83.56**	**83.56**	**80.56**

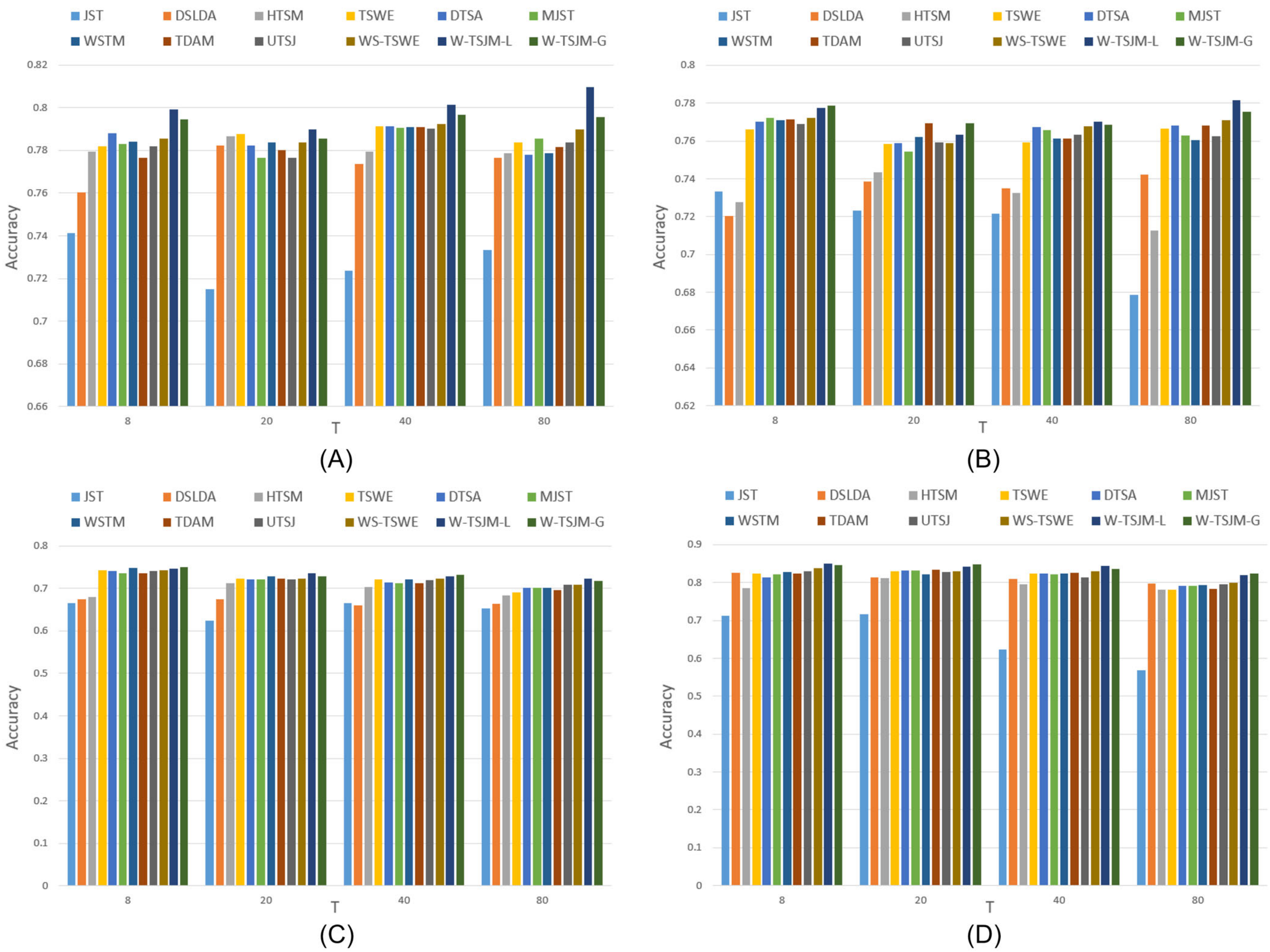

FIGURE 6 The accuracy of sentiment classification of different models under different topic numbers on the four different data sets. A, Accuracy on different T in the CR data set; B, Accuracy on different T in the HR data set; C, Accuracy on different T in the BR data set; D, Accuracy on different T in the MR04 data set

From the experimental results in Figure 7, we can derive the following conclusions:

$$\text{Perplexity}(L-\text{TSWE}) < \text{Perplexity}(\text{JST}) \tag{28}$$

$$\text{Perplexity}(L-\text{TSWE}) < \text{Perplexity}(G-\text{TSWE}) \tag{29}$$

$$\text{Perplexity}(G-\text{TSWE}) < \text{Perplexity}(\text{JST}) \tag{30}$$

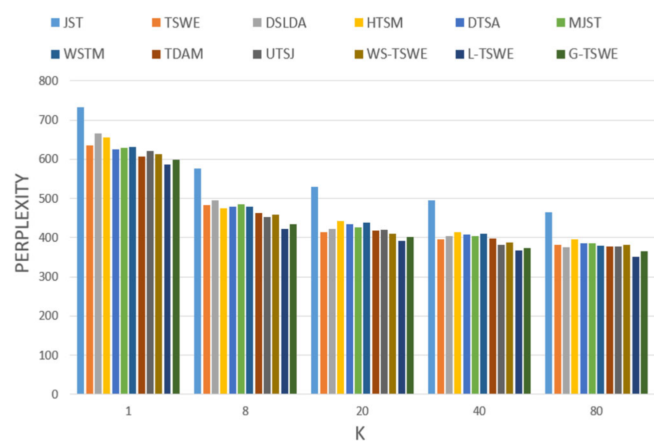

FIGURE 7 The perplexity of every model on CR data set

$$\text{Perplexity}(L - \text{TSWE}) < \text{Perplexity}(\text{TSWE}) \tag{31}$$

$$\text{Perplexity}(\text{TSWE}) < \text{Perplexity}(\text{JST}) \tag{32}$$

$$\text{Perplexity}(G - \text{TSWE}) < \text{Perplexity}(\text{HTSM}) \tag{33}$$

$$\text{Perplexity}(\text{HTSM}) < \text{Perplexity}(\text{DSLDA}) \tag{34}$$

$$\text{Perplexity}(L - \text{TSWE}) < \text{Perplexity}(WS - \text{TSWE}) \tag{35}$$

$$\text{Perplexity}(WS - \text{TSWE}) < \text{Perplexity}(\text{JST}) \tag{36}$$

$$\text{Perplexity}(\text{DSTA}) < \text{Perplexity}(\text{JST}) \tag{37}$$

$$\text{Perplexity}(\text{UTSJ}) < \text{Perplexity}(\text{JST}) \tag{38}$$

$$\text{Perplexity}(\text{TDAM}) < \text{Perplexity}(\text{JST}) \tag{39}$$

$$\text{Perplexity}(\text{MJST}) < \text{Perplexity}(\text{JST}) \tag{40}$$

$$\text{Perplexity}(\text{WSTM}) < \text{Perplexity}(\text{JST}). \tag{41}$$

According to Equations (28), (30), (32), (36), (37), (38), (39), (40), and (41), we can see that the classification results of TSJM-WED-L, TSJM-WED-G, DSTA, MJST, WSTM, UTSJ, TDAM, WS-TSWE, and TSWE are better than the JST based on word frequency statistics. From Equations (29), (33), (34), and (35), it is shown that the TSJM-WED-L model has better classification result in the four topic sentiment joint models with dependency. According to Equation (31), with dependency among word embeddings is better than the topic sentiment joint model, which only considers word embedding. We can see the perplexity values decrease with larger topic number. The generalization performance increases when multiple topics are considered. However, the perplexity value begins to be stable after K = 40.

For a test set of D documents, and topic number K, the NMI is

$$\text{NMI} = \sum_{k=1}^{K} \sum_{1\leqslant i\leqslant j\leq D} \frac{log\left(\frac{p(w_i.w_j)}{p(w_i)p(w_j)}\right)}{-log(p(w_i, w_j))}. \tag{42}$$

The NMI scores always range from 0.0 to 1.0, and a higher score shows better clustering performance. As shown in Figure 6, when the number of the topic is set to 40, the perplexity begins to stabilize. In order to further verify the validity of the model, we set the number of topic as 40 and compare the perplexity and NMI of different models on four different data sets. Table 3 lists the experimental results.

Table 3 shows the TSJM-WED-L and TSJM-WED-G models perform better in perplexity and NMI compared to the JST, TSWE, DSLDA, HTSM, DTSA, MJST, WSTM, UTSJ, TDAM, and WS-TSWE models. We can see that the perplexity on the MR04 data set is higher than that on the other data sets. The reason is that the word number in the corpus is more than others. It also proves that it is appropriate to analyze the topic and sentiment in a unified way. The perplexity and NMI of TSWE, which use word embeddings, are also superior to the original topic sentiment joint model JST. This is because the TSWE model introduces the external word embeddings based on the JST model. It enriches the semantic representation of the words so that the semantic links of the sentiments-topics and words and words and words get closer. Compared with the TSWE model, the TSJM-WED-L model and the TSJM-WED-G model use the long-short-term memory networks to incorporate the dependency among words on the basis of the TSWE model, which makes the semantic representation of words richer. They can detect the sentiment-topic more accurately and get better classification results. It can be seen from the ablation experiment, the perplexity and NMI of TSJM-WED-L and TSJM-WED-G are slightly improved compared with the TSJM-WED-L′ and TSJM-WED-G′, respectively, while the perplexity and NMI of TSJM-WED-L and TSJM-WED-G are significantly improved compared with the W-TSJM-WED model. Moreover, we can find that using RNN model has more improvement than using word embeddings, because using RNN models can be better at capturing the dependency characteristics among words. While comparing the NMI and perplexity of the TSJM-WED-L model and the TSJM-WED-G model, TSJM-WED-L model is a little better than the TSJM-WED-G model on most data sets. However, the TSJM-WED-G model is relatively simple and the training time is slightly shorter than that of the TSJM-WED-L model.

TABLE 3 The perplexity and NMI results of different models on four different data sets

Data sets	Model	NMI(%)	Perplexity
CR	JST	39.12	494.35
	DSLDA	54.95	412.68
	HTSM	54.32	410.36
	TSWE	55.61	405.36
	DTSA	54.21	401.55
	MJST	54.41	401.13
	WSTM	54.63	400.92
	UTSJ	54.78	399.56
	TDAM	54.96	398.79
	WS-TSWE	55.03	392.15
	W-TSJM-WED	53.33	417.56
	TSJM-WED-L′	54.11	390.34
	TSJM-WED-G′	55.23	397.64
	TSJM-WED-L	54.23	389.05
	TSJM-WED-G	55.34	395.39
HR	JST	39.53	734.56
	DSLDA	41.35	714.65
	HTSM	42.68	706.69
	TSWE	45.14	693.02
	DTSA	44.89	687.31
	MJST	44.93	697.12
	WSTM	45.07	695.79
	UTSJ	45.89	689.33
	TDAM	44.31	691.59
	WS-TSWE	45.03	685.33
	W-TSJM-WED	43.12	708.43
	TSJM-WED-L′	43.54	685.44
	TSJM-WED-G′	45.65	697.36
	TSJM-WED-L	44.35	680.96
	TSJM-WED-G	45.26	695.43
BR	JST	27.36	721.36
	DSLDA	29.65	648.69
	HTSM	32.39	658.34
	TSWE	34.34	644.47
	DTSA	33.45	640.56
	MJST	33.51	639.44
	WSTM	34.73	639.12
	UTSJ	34.16	639.51
	TDAM	33.56	638.92
	WS-TSWE	34.87	638.34
	W-TSJM-WED	32.12	678.85
	TSJM-WED-L′	35.09	638.54
	TSJM-WED-G′	33.47	642.77
	TSJM-WED-L	35.43	635.66
	TSJM-WED-G	33.97	639.59
MR04	JST	20.11	1468.51
	DSLDA	52.96	1338.36
	HTSM	50.36	1346.39
	TSWE	52.73	1348.31
	DTSA	52.37	1344.28
	MJST	52.43	1341.92
	WSTM	52.69	1339.47
	UTSJ	52.96	1338.62
	TDAM	53.09	1331.44
	WS-TSWE	53.12	1325.49
	W-TSJM-WED	51.23	1398.54
	TSJM-WED-L′	52.89	1367.34
	TSJM-WED-G′	53.17	1345.78
	TSJM-WED-L	53.06	1312.89
	TSJM-WED-G	53.15	1328.36

In addition, the main purpose of the topic sentiment joint model is to extract the sentiment of the text and the topic under each sentiment. The sentiment labels in the experiment are positive and negative. There are top 15 words of the partial sentiment-topic probabilities, which trained by the TSJM-WED-L model and the DSLDA model on the CR data set in Table 4.

TABLE 4 Top 15 words of the partial sentiment-topic probabilities of TSJM-WED-L and DSLDA on the CR

	TSJM-WED-L	
	Positive	
Topic 1	**Topic 2**	**Topic 3**
pretty	performance	heat dissipation
appearance	internet access	not bad
run	game	satisfy
overall	system	interface
stable	not bad	fan
girl	software	suitable
bright	family use	select
majestic	genuine	calorific value
compare	majestic	power supply
to start	compatible	charge
hard disk	filling at the price	carry
mall	problem	technology
Domestic goods	Web page	Configuration
body	reasonable	introduction
On the grade	Texture	Speed
	Negative	
Topic 1	**Topic 2**	**Topic 3**
jingdong	screen	calorific value
service	feeling	sound
delivery	depressed	normal
order	not good	start up
express delivery	hardware	fan
packaging	disadvantages	heat sink
telephone	return	strenuous
ship	not enough	heat
disappointed	difficult	hot
freight	sign up	not good
hope	strange	battery
invoice	responsibility	power supply
endure	gap	touch
quality	bad luck	bottom
company	malfunction	charge
	DSLDA	
	Positive	
Topic 1	**Topic 2**	**Topic 3**
pretty	performance	compare
speed	internet access	heat dissipation
machine	normal	laptop
running	cheap	game
features	brand	design
camera	computer	sound
suitable	price	overall
appearance	weight	interface
white	already	indeed
imagine	friend	piano
fashion	software	noise
office	enough	fingerprint
girl	series	aspect
stable	lenovo	practical
entirely	accept	complete

(Continues)

TABLE 4 Continued

	Negative	
Topic 1	**Topic 2**	**Topic 3**
question	system	compare
know	trouble	heat dissipation
jingdong	preloaded	keyboard
thing	advice	online
service	all	machine
appear	ship	interface
reason	not good	lenovo
solve	refit	pyrexia
unknown	screen	disadvantage
delivery	sound	machine
finally	depressed	laptop
already	affect	edition
genuine	operate	price
express delivery	use	complete
telephone	find	trouble

In the TSJM-WED-L model, Topic 1 is a positive comment mainly about the appearance of computer under the positive sentiment label. There are topical words such as "appearance, overall, body, run," and sentimental words such as "pretty, bright, majestic, On the grade." Topic 2 mainly makes comments about the computer's system and performance. Topic 3 mainly talks about computer's heat dissipation problems. In the DSLDA model, under the positive sentiment label and negative sentiment label, the topical words and sentimental words are not obvious. It cannot clearly identify the main content of the topic, such as the Topic 3 under the positive model. It is difficult for us to determine what the topic is by these words under the topic. Moreover, there are fewer sentimental words.

In the TSJM-WED-L model under the negative sentiment label, Topic 1 is mainly about the evaluation of logistics service to online shopping computer, such as "jingdong, service, delivery, order, express delivery" and other words about logistics, and "disappointed, endure" sentimental words. Topic 2 is a comment mainly about the screen, hardware issues, and some negative sentimental words. Topic 3 is mainly about computer heat dissipation problems. In the DSLDA model, Topic 1 also describes logistics problem, but it has less sentimental words.

In conclusion, our proposed model (TSJM-WED) can identify more sentiment words more accurately than other models. Therefore, we can detect more sensitive information and negative words in edge social system, so as to better maintain network security and national security.

5 | CONCLUSIONS AND FUTURE WORKS

5.1 | Conclusions

Identifying sentiment polarity and detecting topics from social reviews have been widely and separately studied in recent years. However, in many cases, we concern about sentiment and topics simultaneously. There have been some recent works being aware of this limitation; thus, they attempt to capture sentiments and mixture of topics simultaneously. In this paper, we proposed a novel topic sentiment joint model TSJM-WED (TSJM-WED-L and TSJM-WED-G) to overcome the defect of TSWE model for performing the sentiment classification and topic detection on online review data sets. Although the word embeddings are learned from external corpus, which contains certain semantic information, it has not introduced the sequence information of words. As a result, the dependency among words are not considered, which leads to the results that sentiment classification and topic detection are not ideal. We utilized the topic vectors obtained from the TSWE model and the recurrent neural networks LSTM and GRU to deliver the semantic information of a word to the next word. Meanwhile, the topic and sentiment information can be also delivered to the next word. We consider the dependency among words in this way to fully extend semantic information, and then we compare with several other recent models, such as JST, HTSM, DSLDA, DSTA, UTSJ, MJST, WSTM, TDAM, WS-TSWE, and TSWE. The experimental results show that TSJM-WED is effective in discovering sentiment and extracting sentiment-topics. By this way, it enables us to detect the security problems in the edge social system through more reliable sentiment analysis and topic detection technology, thus prevent the network attack in advance.

5.2 | Future works

Although the proposed method takes the dependency information among words and semantics associations in the text into account, it abstracts away from the influence of time on the sentiment and topic. The following study will consider the use of the existing inference algorithms such as the collapse of Gibbs sampling[45,46] and variational Bayesian[51-57] to detect dynamic topic and sentiment. We also consider developing a more flexible and efficient inference algorithm to accommodate the training requirements of the large-scale corpus. In addition, as for the social media content analysis, the collaborative training among the topic detection, sentiment analysis, and the deep learning can be a very promising solution in the future large-scale social media data analysis and processing for the particularity and complexity of their data. Furthermore, with respect to cyber security, sentiment analysis and topic detection are also important methods to detect cyber security problems and prevent cyber attacks, which provides another example for future work.

ACKNOWLEDGMENTS

This research was supported by the National Natural Science Foundation of China under grants 61472258 and 61402294, National Key Technology Research and Development Program of the Ministry of Science and Technology of China under grant 2014BAH28F05, Technology Planning Project of Guangdong Province under grant 2014B010118005, and Science and Technology Foundation of Shenzhen City under grants JCYJ20140509172609162 and JCYJ20130329102032059.

ORCID

Wenhao Xie https://orcid.org/0000-0002-8531-4169

REFERENCES

1. Fu X, Sun X, Wu H, Cui L, Huang JZ. Weakly supervised topic sentiment joint model with word embeddings. *Knowl Based Syst*. 2018;147:43-54.
2. Han X, Wang L, Cui C, Ma J, Zhang S. Linking multiple online identities in criminal investigations: a spectral co-clustering framework. *IEEE Trans Inf Forensics Secur*. 2017;12(9):2242-2255.
3. Gorodnichenko Y, Pham T, Talavera O. Social media, sentiment and public opinions: evidence from #Brexit and #USElection. Cambridge, MA: National Bureau of Economic Research; 2018.
4. Alomari A. Distance impact on quality of video streaming services in cloud environments. *Int J Space Based Situated Comput*. 2017;7(3):119-128.
5. Xu K, Qi G, Huang J, Wu T, Fu X. Detecting bursts in sentiment-aware topics from social media. *Knowl Based Syst*. 2018;141:44-54.
6. Rout JK, Choo K-KR, Dash AK, Bakshi S, Jena SK, Williams KL. A model for sentiment and emotion analysis of unstructured social media text. *Electron Commer Res*. 2018;18(1):181-199.
7. Ibrahim R, Elbagoury A, Kamel MS, Karray F. Tools and approaches for topic detection from twitter streams: survey. *Knowl Inf Syst*. 2018;54(3):511-539.
8. Blei DM, Ng AY, Jordan MI. Latent dirichlet allocation. *J Mach Learn Res*. 2003;3(Jan):993-1022.
9. Hofmann T. Probabilistic latent semantic indexing. Paper presented at: International ACM SIGIR Conference on Research and Development in Information Retrieval; 1999; Berkeley, CA.
10. Lin C, He Y. Joint sentiment/topic model for sentiment analysis. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management; 2009; Hong Kong.
11. Lin C, He Y, Everson R, Ruger S. Weakly supervised joint sentiment-topic detection from text. *IEEE Trans Knowl Data Eng*. 2012;24(6):1134-1145.
12. Xu G, Meng Y, Chen Z, Qiu X, Wang C, Yao H. Research on topic detection and tracking for online news texts. *IEEE Access*. 2019;7:58407-58418.
13. Peng M, Ouyang S, Zhu J, Huang J, Wang H, Yong J. Emerging topic detection from microblog streams based on emerging pattern mining. Paper presented at: 2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design (CSCWD); 2018; Nanjing, China.
14. Zhang L, Wang S, Liu B. Deep learning for sentiment analysis: a survey. *Wiley Interdiscip Rev Data Min Knowl Discov*. 2018;8(4):e1253.
15. Dragoni M, Poria S, Cambria E. Ontosenticnet: a commonsense ontology for sentiment analysis. *IEEE Intell Syst*. 2018;33(3):77-85.
16. Majumder N, Hazarika D, Gelbukh A, Cambria E, Poria S. Multimodal sentiment analysis using hierarchical fusion with context modeling. *Knowl Based Syst*. 2018;161:124-133.
17. Li F, Huang M, Zhu X. Sentiment analysis with global topics and local dependency. In: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence; 2010; Atlanta, GA.
18. Li J, Qin Y, Huang R. A user-oriented semi-supervised probabilistic topic model. Paper presented at: 2016 2nd IEEE International Conference on Computer and Communications (ICCC); 2016; Chengdu, China.

19. Dong L, Ji S, Zhang C, Zhang Q, Chiu DW, Qiu L, Li D. An unsupervised topic-sentiment joint probabilistic model for detecting deceptive reviews. *Expert Syst Appl*. 2018;114:210-223.
20. Xiong S, Wang K, Ji D, Wang B. A short text sentiment-topic model for product reviews. *Neurocomputing*. 2018;297:94-102.
21. Mei Q, Ling X, Wondra M, Su H, Zhai C. Topic sentiment mixture: modeling facets and opinions in weblogs. In: Proceedings of the 16th International Conference on World Wide Web; 2007; Banff, Canada.
22. Fu X, Wu H, Cui L. Topic sentiment joint model with word embeddings. In: Proceedings of DMNLP, Workshop at ECML/PKDD; 2016; Riva del Garda, Italy.
23. Liu P, Gulla JA, Zhang L. A joint model for analyzing topic and sentiment dynamics from large-scale online news (retraction of vol 21, pg 1117, 2018). New York, NY: Springer; 2019.
24. Rahman MM, Wang H. Hidden topic sentiment model. In: Proceedings of the 25th International Conference on World Wide Web; 2016; Montréal, Canada.
25. Hai Z, Cong G, Chang K, Cheng P, Miao C. Analyzing sentiments in one go: a supervised joint topic modeling approach. *IEEE Trans Knowl Data Eng*. 2017;29(6):1172-1185.
26. Dermouche M, Kouas L, Velcin J, Loudcher S. A joint model for topic-sentiment modeling from text. In: Proceedings of the 30th Annual ACM Symposium on Applied Computing; 2015; Salamanca, Spain.
27. Pergola G, Gui L, He Y. TDAM: a topic-dependent attention model for sentiment analysis. *Inf Process Manag*. 2019;56(6):102084.
28. Huang F, Zhang S, Zhang J, Ge Y. Multimodal learning for topic sentiment analysis in microblogging. *Neurocomputing*. 2017;253:144-153.
29. Young T, Hazarika D, Poria S, Cambria E. Recent trends in deep learning based natural language processing. *IEEE Comput Intell Mag*. 2018;13(3):55-75.
30. Gardner M, Grus J, Neumann M, et al. AllenNLP: a deep semantic natural language processing platform. arXiv preprint arXiv:1803.07640. 2018.
31. Miikkulainen R, Liang J, Meyerson E, et al. Evolving deep neural networks. In: *Artificial Intelligence in the Age of Neural Networks and Brain Computing*. Cambridge, MA: Elsevier; 2019:293-312.
32. Aroyehun ST, Gelbukh A. Aggression detection in social media: using deep neural networks, data augmentation, and pseudo labeling. In: Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying; 2018; Santa Fe, NM.
33. Monti F, Frasca F, Eynard D, Mannion D, Bronstein MM. Fake news detection on social media using geometric deep learning. arXiv preprint arXiv:1902.06673. 2019.
34. Zhang Z, Robinson D, Tepper J. Detecting hate speech on twitter using a convolution-gru based deep neural network. Paper presented at: European Semantic Web Conference; 2018; Heraklion, Greece.
35. Chakraborty A, Kolya A, Dutta S. A fuzzy logic inspired approach for social media sentiment analysis via deep neural network. In: *Advanced Computational and Communication Paradigms*. Singapore: Springer; 2018.
36. Graves A. Long short-term memory. In: *Supervised Sequence Labelling with Recurrent Neural Networks*. Berlin, Germany: Springer; 2012:37-46.
37. Nabil M, Atyia A, Aly M. Cufe at semeval-2016 task 4: a gated recurrent model for sentiment classification. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016); 2016; San Diego, CA.
38. Tang D, Qin B, Liu T. Document modeling with gated recurrent neural network for sentiment classification. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing; 2015; Lisbon, Portugal.
39. Tian F, Gao B, He D, Liu T-Y. Sentence level recurrent topic model: letting topics speak for themselves. arXiv preprint arXiv:1604.02038. 2016.
40. Wang X, Liu Y, Chengjie S, Wang B, Wang X. Predicting polarities of tweets by composing word embeddings with long short-term memory. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing; 2015; Beijing, China.
41. Goto M. A predominant-F/sub 0/ estimation method for CD recordings: MAP estimation using EM algorithm for adaptive tone models. Paper presented at: 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing; 2001; Salt Lake City, UT.
42. Paul MJ, Dredze M. SPRITE: generalizing topic models with structured priors. *Trans Assoc Comput Linguist*. 2015;3:43-57.
43. Zhu C, Byrd RH, Lu P, Nocedal J. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Trans Math Softw*. 1997;23(4):550-560.
44. McCallum AK. Mallet: a machine learning for language toolkit (2002); 2002.
45. Park H, Park T, Lee YS. Partially collapsed Gibbs sampling for latent Dirichlet allocation. *Expert Syst Appl*. 2019;131:208-218.
46. Robert C, Casella G. *Monte Carlo Statistical Methods*. New York, NY: Springer Science & Business Media; 2013.
47. Griffiths TL, Steyvers M. Finding scientific topics. *Proc Natl Acad Sci*. 2004;101(suppl 1):5228-5235.
48. Manning C, Raghavan P, Schütze H. Introduction to information retrieval. *Nat Lang Eng*. 2010;16(1):100-103.
49. Kutuzov A, Kuzmenko E. WebVectors: a toolkit for building web interfaces for vector semantic models. Paper presented at: International Conference on Analysis of Images, Social Networks and Texts; 2016; Yekaterinburg, Russia.
50. Vargas-Calderón V, Camargo JE. Characterization of citizens using word2vec and latent topic analysis in a large set of tweets. *Cities*. 2019;92:187-196.
51. Glynn C, Tokdar ST, Howard B, Banks DL, et al. Bayesian analysis of dynamic linear topic models. *Bayesian Analysis*. 2019;14(1):53-80.
52. Liu Y, Wang J, Jiang Y, Sun J, Shang J. Identifying impact of intrinsic factors on topic preferences in online social media: a nonparametric hierarchical Bayesian approach. *Information Sciences*. 2018;423:219-234.
53. Peng M, Xie Q, Wang H, Zhang Y, Tian G. Bayesian sparse topical coding. *IEEE Trans Knowl Data Eng*. 2018;31(6):1080-1093.

54. Bai H, Chen Z, Lyu MR, King I, Xu Z. Neural relational topic models for scientific article analysis. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management; 2018; Turin, Italy.
55. Wang C, Blei DM, et al. A general Method for Robust Bayesian Modeling. *Bayesian Analysis*. 2018;13(4):1159-1187.
56. Xue J, Eguchi K. Sequential Bayesian nonparametric multimodal topic models for video data analysis. *IEICE Trans Inf Syst*. 2018;101(4):1079-1087.
57. Tang YK, Mao XL, Huang H, Shi X, Wen G. Conceptualization topic modeling. *Multimed Tools Appl*. 2018;77(3):3455-3471.

How to cite this article: Xie W, Fu X, Zhang X, Lu Y, Wei Y, Yang J. Topic sentiment analysis using words embeddings dependency in edge social system. *Trans Emerging Tel Tech*. 2022;33:e3817. https://doi.org/10.1002/ett.3817