Mobile Networks and Applications (2021) 26:2298–2314
https://doi.org/10.1007/s11036-021-01847-w

TA-BiLSTM: An Interpretable Topic-Aware Model for Misleading Information Detection in Mobile Social Networks

Shuyu Chang[1] · **Rui Wang**[1] · **Haiping Huang**[2] · **Jian Luo**[1]

Accepted: 8 September 2021 / Published online: 10 November 2021

Abstract

As essential information acquisition tools in our lives, mobile social networks have brought us great convenience for communication. However, misleading information such as spam emails, clickbait links, and false health information appears everywhere in mobile social networks. Prior studies have adopted various approaches to detecting this information but ignored global semantic features of the corpus and lacked interpretability. In this paper, we propose a novel end-to-end model called Topic-Aware BiLSTM (TA-BiLSTM) to handle the problems above. We firstly design a neural topic model for mining global semantic patterns, which encodes word relatedness into topic embeddings. Simultaneously, a detection model extracts local hidden states from text content with LSTM layers. Then, the model fuses those global and local representations with the Topic-Aware attention mechanism and performs misleading information detection. Experiments on three real datasets prove that the TA-BiLSTM could generate more coherent topics and improve the detecting performance jointly. Furthermore, case study and visualization demonstrate that the proposed TA-BiLSTM could discover latent topics and help in enhancing interpretability.

Keywords Misleading information detection · Deep representation learning · Neural topic model · Attention mechanism · Mobile social networks

1 Introduction

Mobile social networks have brought us great facilities for acquiring information. Inevitably, a vast amount of useless misleading information, such as spam emails, clickbait links, and false health information, is created. This information will deceive us to do things with ill consequences. Table 1 gives two examples of how the meanings of content mislead people and impact categories in the Webis-Clickbait-17 dataset. In general, misleading information is deceptive, which makes it hard to distinguish the difference between two kinds of posts (positive and negative). Thus, how to detect misleading information effectively is challenging. Also, developing an efficient approach with high performance for misleading information detection is particularly essential.

Existing work on misleading information detection could be categorized into two types: machine learning-based approaches and deep learning-based approaches. Approaches based on machine learning often build document representations depending on different feature engineering techniques [10, 26, 35]. Various algorithms such as Labeled-LDA [35] and GBDT [2] also help enhance detection accuracy. Unfortunately, these approaches heavily rely on people to design sophisticated features and will cause lousy performance in a complex context. Deep learning-based approaches extract semantic features from content by multiple non-linear units to solve the above problems. Convolutional neural networks [1, 17], recurrent neural networks [23], and a combination of the two [22]

S. Chang and R. Wang contributed equally to this work.

✉ Haiping Huang
hhp@njupt.edu.cn

Shuyu Chang
csy_njupt@163.com

Rui Wang
rui_wang@njupt.edu.cn

Jian Luo
luoj@njupt.edu.cn

[1] School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210003, Jiangsu, China

[2] Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210003, Jiangsu, China

Table 1 Content from different websites may carry normal or misleading information

Category	Content
(a) Misleading Information	*- Log in or sign up to create your own posts.* *- Just a comforting reminder that we all have these thoughts.* *- BuzzFeed Home ©2017 BuzzFeed, Inc.*
(b) Normal Information	*- A family has been rescued from their truck that was dangling over a cliff-edge in southern China.* *- The father, who was driving, said the road was slippery.*

are commonly used frameworks. Still, these approaches are limited to local semantic information and severely lack interpretability due to the complex structures.

To address the above limitations, we propose a novel model called Topic-Aware BiLSTM (TA-BiLSTM) to add corpus-level topic relatedness and enhance interpretability. Specifically, the TA-BiLSTM is decomposed into two parts: a neural topic model module and a text classification module. Assuming that a multi-layer neural network can approximate the document's topic distribution, we model the topic by Wasserstein autoencoder (WAE) [37]. Neural topic model module constructs the topic distribution on latent space and reconstructs the document representation. The topic distribution could be transformed into the topic embedding provided for the attention mechanism concurrently. Unlike variational autoencoder-based approaches previously [29, 36], our model minimizes the Maximum Mean Discrepancy regularizer [15] based on Optimal Transport theory [39] to reduce Wasserstein distance between the topic distribution and Dirichlet prior.

Furthermore, the text classification module utilizes a two-layer bidirectional LSTM based on the Topic-Aware attention mechanism to extract semantic features. This attention mechanism incorporates topic relatedness information while calculating the representation. Finally, we input representations to the classifier for misleading information detection. To balance the two task learning, we leverage a dynamic strategy to control the importance of these objectives. We concentrate on the neural topic model preferentially, then simultaneously train the classification objective and topic modeling objective.

The main contributions of our work are as follows:

- We propose a novel end-to-end framework Topic-Aware BiLSTM for misleading information detection.
- We introduce a new Topic-Aware attention mechanism to encode the document's local semantic and global topical representation.
- Experiments are conducted on three public datasets to verify the effectiveness of our Topic-Aware BiLSTM model in terms of topic coherence measures and classification metrics.
- We select representative cases from different datasets for visualization, demonstrating that the Topic-Aware BiLSTM enhances interpretability than other traditional approaches.

The remainder of the paper is organized as follows: Section 2 reviews the relevant work, and Section 3 introduces preliminary techniques. Section 4 introduces the methodology of Topic-Aware BiLSTM model. Experiments and result analysis are given in Section 5. Lastly, in Section 6, we conclude the paper.

2 Related Work

Our work is related to three lines of research which are misleading information detection, topic modeling and attention mechanism.

2.1 Misleading Information Detection

Misleading information detection models could be categorized as two streams based on implementation techniques: machine learning-based approaches and deep learning-based approaches.

Generally, machine learning-based approaches need to design the specific representation of texts. For example, Liu et al. [26] employs both the local and the global features via Latent Dirichlet Allocation and utilizes Adaboost to detect spammer. Likewise, Chakraborty et al. [7] uses multinomial Naive Bayes classifiers for pruned features of Clickbait data. Different models of this branch could also result in different detection performance. Song et al. [35] proposes the labeled latent Dirichlet allocation to mine the latent topics from user-generated comments and filter social spam. Biyani et al. [2] uses Gradient Boosted Decision Trees [11] to detect clickbait in news streams. Similarly, Elhadad et al. [10] detects misleading information about COVID-19 through constructing a voting mechanism. However, approaches of this branch often require sophisticated feature engineering and could not capture deep semantic patterns.

Thanks to the rapid development of deep representation learning, approaches such as convolutional neural networks, recurrent neural networks have been applied to extract

semantic representation from text directly. Agrawal [1] and Hai-Tao et al. [17] utilize a convolutional neural network to detect misleading information from clickbait. Kumar et al. [23] adopts a bidirectional LSTM with an attention mechanism to learn a word contributing to the clickbait score in a different manner. Jain et al. [22] constructs a deep learning architecture based on convolutional layers and long short-term memory layers. Nevertheless, deep learning-based approaches often have complex structures and severely lack interpretability. Thus, we integrate the neural topic model to provide corpus-level semantic information and enhance interpretability.

2.2 Topic Modeling

Given a collection of documents, each document will discuss different topics. Topic modeling is an efficient technique which could mine latent semantic patterns from corpus.

Latent Dirichlet Allocation (LDA) [3] is the most publicly used traditional probabilistic generative model that can perform topic mining. Unlike traditional graphical topic models, Miao et al. [29] proposes a neural topic model NVDM based on variational autoencoders (VAE). Variational autoencoders use KL divergence to measure the distance between the topic distribution and Gaussian prior. ProdLDA [36] utilizes the approximated Dirichlet prior through Laplace approximation and improves the topic quality. On the other hand, Wang et al. proposes ATM [43], BAT, and Gaussian-BAT [44] in an adversarial manner. Wang et al. [42] also extends the ATM model for open event extraction. Inspired by ATM model, Hu et al. [20] attempts to improve topic modeling with cycle-consistent adversarial training and names this approach ToMCAT. Zhou et al. [49] extends this line of work by taking into account documents and words as nodes in the graph. Further, autoencoders could be trained stably and reduce the document's representation dimensionally [25] to extract the most effective information [48]. So Nan et al. [31] incorporates adversarial training into Wasserstein autoencoder framework and proposes W-LDA model for unsupervised topic extraction.

2.3 Attention Mechanism

The attention mechanism is a brain processing mechanism unique to human vision originally. When we see a picture in life, our brain will prioritize the main content in the image, ignoring the background and other irrelevant information.

Inspired by this mechanism of the human brain, various attention mechanisms have achieved success in natural language processing tasks, such as sentiment analysis [45] and machine translation [27]. The typical attention mechanism only pays attention to word-level dependencies and assigns weights so that the model could highlight key elements of sentences [18]. Further, the hierarchical attention mechanism [47] uses two-layer attention, which is successively applied at the word level and sentence level to generate the document representation with rich semantics. Besides, Vaswani et al. [38] proposes a self-attention mechanism to deal with the increasing length of text. Self-attention calculates associations between words in a sentence directly. Previous work [16, 41] has shown that topic information could improve the semantic representation of text with the help of attention mechanisms. Nevertheless, to our best knowledge, no relevant work has been conducted on misleading information detection, so we explore and study in this work.

3 Preliminaries

3.1 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is the most commonly used generative model for topic extraction. Assuming that a document can be represented by the probability distribution over topics, and each topic can be represented by the probability distribution over words. To learn the topic better, LDA utilizes the Dirichlet distribution as prior over latent space.

LDA uses $\boldsymbol{\theta}_d$ to denote the topic distribution of a document d and z_n to represent a topic allocation of the word w_n. Thus, the generative process of documents is shown in Algorithm 1.

Algorithm 1 The generative process of LDA.

for each document d **do**
  Draw topic distribution $\boldsymbol{\theta}_d \sim Dir(\boldsymbol{\alpha}')$
  for each word at position n **do**
    Allocate topic $z_n \sim Multi(\boldsymbol{\theta}_d)$
    Sample word $w_n \sim Multi(\varphi_{z_n})$
  end for
end for

Here, $Dir(\boldsymbol{\alpha}')$ is the Dirichlet prior distribution, $\boldsymbol{\alpha}'$ signifies the hyper-parameter of Dirichlet prior, and $\boldsymbol{\theta}_d$ is the topic distribution of document d sampled from Dirichlet prior. z_n denotes the topic allocation of each position n in the document, and w_n is a word randomly generate from multinomial distribution. φ_i is a topic-word distribution of the i-th topic, and φ_{z_n} is one column in the matrix. LDA infers these parameters in an unsupervised manner. After model training, we can obtain representative words with

high probabilities in each topic, and these words represent the semantic meaning of each topic.

3.2 Long Short-Term Memory

As text is sequential data, and small changes of word order will affect the meaning of the entire sentence. However, traditional feedforward neural networks cannot directly extract the word dependency of context. Thus, researchers develop sequential models such as Recurrent Neural Networks (RNN) to extract sequential and contextual features from these data [21]. The RNN comprises an input layer, a hidden layer and an output layer. However, as the length of sentences increases, the training process will appear gradient disappearance and gradient explosion. The Long Short-Term Memory (LSTM) [19] adds a cell state to store long-term memory [13], which could deal with this problem.

Assuming that $\mathbf{x}_j \in \mathbb{R}^{D_w}$ represents a word embedding of the j-th word in the content and D_w is the dimension of word embeddings. LSTM feeds in word embeddings as a sequence and calculates the hidden state $\mathbf{h}_j \in \mathbb{R}^{D_h}$ for each word, where D_h is the dimension of hidden states. The calculation procedure follows below equations:

$$\mathbf{f}_j = \sigma\left(\mathbf{W}_f \cdot [\mathbf{h}_{j-1}, \mathbf{x}_j] + \mathbf{b}_f\right) \tag{1}$$

$$\mathbf{i}_j = \sigma\left(\mathbf{W}_i \cdot [\mathbf{h}_{j-1}, \mathbf{x}_j] + \mathbf{b}_i\right) \tag{2}$$

$$\mathbf{C}'_j = \tanh\left(\mathbf{W}_C \cdot [\mathbf{h}_{j-1}, \mathbf{x}_j] + \mathbf{b}_C\right) \tag{3}$$

$$\mathbf{C}_j = \mathbf{f}_j \cdot \mathbf{C}_{j-1} + \mathbf{i}_j \cdot \mathbf{C}'_j \tag{4}$$

$$\mathbf{o}_j = \sigma\left(\mathbf{W}_o \cdot [\mathbf{h}_{j-1}, \mathbf{x}_j] + \mathbf{b}_o\right) \tag{5}$$

$$\mathbf{h}_j = \mathbf{o}_j \cdot \tanh(\mathbf{C}_j) \tag{6}$$

where $\mathbf{W}_f$, $\mathbf{W}_i$, $\mathbf{W}_C$, $\mathbf{W}_o$, $\mathbf{b}_f$, $\mathbf{b}_i$, $\mathbf{b}_C$ and $\mathbf{b}_o$ are learnable parameters, and $\sigma(\cdot)$ is sigmoid function. Forget gate $\mathbf{f}_j$ determines the information that needs to be retained from the cell state $\mathbf{C}_{j-1}$. Input gate $\mathbf{i}_j$ controls the proportion of new information stored in the new candidate $\mathbf{C}_j$. Lastly, LSTM constrains the hidden state of the current node through output gate $\mathbf{o}_j$. The elaborated design of its structure enables LSTM could learn longer dependencies and better semantic representation.

4 Methodology

In this section, we first introduce the Topic-Aware BiLSTM (TA-BiLSTM) model. As depicted in Fig. 1, our proposed TA-BiLSTM could be divided into two parts: a neural topic model and a text classification model. The topic module employs a neural topic model to discover latent topics from

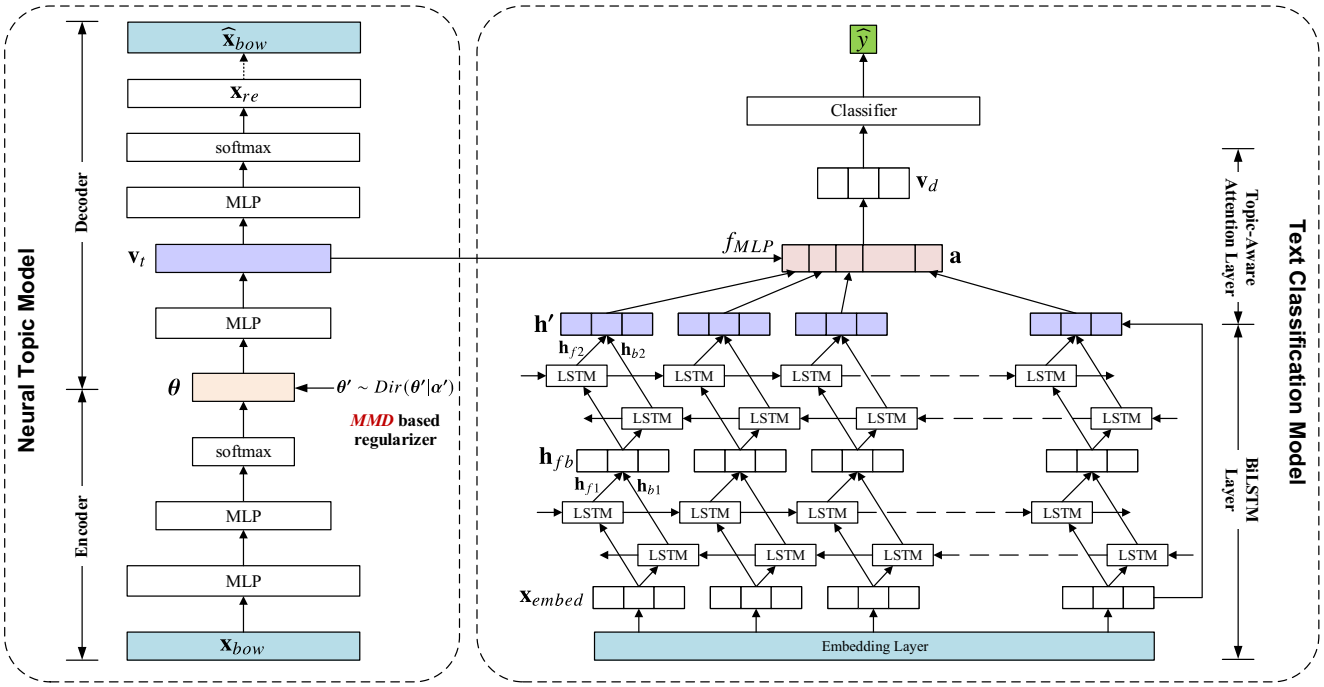


Fig. 1 The overall architecture of TA-BiLSTM: (a)Neural Topic Model on the left; (b)Text Classification Model on the right. MLP and f_{MLP} are multilayer perceptron, $\mathbf{v}_t$ denotes the topic embedding, and $\mathbf{v}_d$ means the document's representation, which is computed through attention weights $\mathbf{a}$

text corpus. The text classification module utilizes a two-layer BiLSTM network based on the Topic-Aware attention mechanism to detect misleading information from text.

4.1 Neural Topic Model

As shown in the left panel of Fig. 1, its structure is composed of an encoder and a decoder. (1) **Encoder** takes the V-dimensional $\mathbf{x}_{bow}$ of the document as the input and transforms it into a topic distribution $\boldsymbol{\theta}$ with K dimension through two fully connected layers. (2) **Decoder** takes the encoded topic distribution $\boldsymbol{\theta}$ as the input, then reconstructs the document $\hat{\mathbf{x}}_{bow}$ with reconstruction distribution $\mathbf{x}_{re}$. After decoded by the first layer, the topic embedding $\mathbf{v}_t$ is collected. Besides, to ensure the quality of extracted topics, we use the Wasserstein distance to conduct prior matching in latent topic space.

4.1.1 Encoder Network

For each document $d = \{w_1, w_2, ..., w_m\}$ in the corpus $C_d = \{d_1, d_2, ..., d_n\}$, the encoder utilizes its bag-of-words representation $\mathbf{x}_{bow}$ as input, where the weights are calculated by TF-IDF formulation:

$$tf_{ij} = \frac{c_{ij}}{\sum_k c_{kj}}, \qquad idf_i = \log \frac{|C_d|}{\left|\left\{j : w_i \in d_j\right\}\right| + 1} \tag{7}$$

where c_{ij} indicates the number of the word w_i appearing in document d_j, and $\sum_k c_{kj}$ is the total number of words in document d_j. $|C_d|$ indicates the total number of documents in the corpus, and $\left|\left\{j : w_i \in d_j\right\}\right|$ represents the number of documents containing word w_i.

$$x_{bow}^{(i)} = tf_{ij} \times idf_i \tag{8}$$

where $x_{bow}^{(i)}$ refers to the semantic relevance of the i-th word in the vocabulary in document d_j.

According to Eqs. 7 and 8, each document could be represented as $\mathbf{x}_{bow} \in \mathbb{R}^V$, where V indicates the vocabulary size.

The encoder firstly maps $\mathbf{x}_{bow}$ into the D_s-dimensional semantic space through following transformation:

$$\mathbf{h}_s = \mathrm{BN}(\mathbf{W}_s \mathbf{x}_{bow} + \mathbf{b}_s) \tag{9}$$

$$\mathbf{o}_s = \max(\mathbf{h}_s, leak * \mathbf{h}_s) \tag{10}$$

where $\mathbf{W}_s \in \mathbb{R}^{D_s \times V}$ and $\mathbf{b}_s \in \mathbb{R}^{D_s}$ are the weight matrix and bias term of the fully connected layer, $\mathbf{h}_s$ is the hidden state normalized by batch normalization $\mathrm{BN}(\cdot)$, $leak$ denotes the hyper-parameter of LeakyReLU activation, and $\mathbf{o}_s$ represents the output of the layer.

Subsequently, the encoder projects the output vector $\mathbf{o}_s$ into a K-dimensional document-topic distribution $\boldsymbol{\theta}_e$:

$$\boldsymbol{\theta}_e = \mathrm{softmax}\left(\mathrm{BN}(\mathbf{W}_o \mathbf{o}_s + \mathbf{b}_o)\right) \tag{11}$$

where $\mathbf{W}_o \in \mathbb{R}^{K \times D_s}$ and $\mathbf{b}_o \in \mathbb{R}^K$ are the weight matrix and bias term of the fully connected layer, $\boldsymbol{\theta}_e$ denotes the topic distribution corresponding to the input $\mathbf{x}_{bow}$ and the k-th ($k \in \{1, 2, ..., K\}$) dimension $\theta_e^{(k)}$ means the proportion of k-th topic in the document.

We add noise to document-topic distribution to draw more consistent topics. We randomly sample a noise vector $\boldsymbol{\theta}_n$ from the Dirichlet prior and merge it with $\boldsymbol{\theta}_e$. The calculation is defined as:

$$\boldsymbol{\theta} = (1 - \eta)\boldsymbol{\theta}_e + \eta \boldsymbol{\theta}_n \tag{12}$$

where $\eta \in [0, 1]$ denotes the mixing proportion of noise.

The encoder transforms the bag-of-words representation into topic distribution which perceives the semantic information in latent space.

4.1.2 Decoder Network

The decoder takes the topic distribution $\boldsymbol{\theta}$ as input. And then, two fully connected layers reconstruct the document's word representation $\hat{\mathbf{x}}_{bow}$. After the transformation of first layer, $\mathbf{v}_t$ serves as the topic embedding of the input document and is provided for the attention mechanism.

The decoder firstly transforms the topic distribution $\boldsymbol{\theta}$ into the D_t-dimensional topic embedding space:

$$\mathbf{h}_t = \mathrm{BN}(\mathbf{W}_t \boldsymbol{\theta} + \mathbf{b}_t) \tag{13}$$

$$\mathbf{v}_t = \max(\mathbf{h}_t, leak * \mathbf{h}_t) \tag{14}$$

where $\mathbf{W}_t \in \mathbb{R}^{D_t \times K}$ and $\mathbf{b}_t \in \mathbb{R}^{D_t}$ are the weight matrix and bias of the fully connected layer, $\mathbf{h}_t$ is the hidden vector normalized by batch normalization $\mathrm{BN}(\cdot)$. The $\mathbf{v}_t$ is activated by the LeakyReLU and then used in Topic-Aware attention mechanism.

Subsequently, the decoder transforms the hidden vector $\mathbf{h}_t$ into V-dimensional reconstruction distribution:

$$\mathbf{x}_{re} = \mathrm{softmax}\left(\mathrm{BN}(\mathbf{W}_r \mathbf{h}_t + \mathbf{b}_r)\right) \tag{15}$$

where $\mathbf{W}_r \in \mathbb{R}^{V \times D_t}$ and $\mathbf{b}_r \in \mathbb{R}^V$ are the weight matrix and bias, and $\mathbf{x}_{re}$ is the reconstruction distribution.

The decoder is an essential part of the neural topic model. After model training, it could generate the words corresponding to each topic. We input one-hot vectors into the decoder to obtain the word distribution of each topic. Here, we use 10 words with the highest probability of each topic to represent its semantic meaning. Based on the topic distribution and the semantics of topics, interpretable word-level information could be provided for classifying documents in the detection process.

4.1.3 Prior Distribution Matching

Since the Dirichlet distribution is commonly regarded as the prior of multinomial distribution, choosing this prior has substantial advantages [40]. To match the encoded topic distribution to Dirichlet prior, we add a regularizer in TA-BiLSTM. Thus, the training process minimizes the regularization term based on the Maximum Mean Discrepancy (MMD) [15] to reduce the Wasserstein distance, which measures the divergence between the topic distribution $\boldsymbol{\theta}$ and randomly samples $\boldsymbol{\theta'}$ from prior.

Regarding the kernel function is $\mathbf{k} : \Theta \times \Theta \rightarrow \Re$, the MMD based regularizer could be defined as:

$$\begin{aligned}\mathcal{D}_\Theta &= \mathrm{MMD}_{\mathbf{k}}(Q_\Theta, P_\Theta) \\ &= \left\| \int_\Theta \mathbf{k}(\boldsymbol{\theta}, \cdot) dP_\Theta(\boldsymbol{\theta}) - \int_\Theta \mathbf{k}(\boldsymbol{\theta}, \cdot) dQ_\Theta(\boldsymbol{\theta}) \right\|_{\mathcal{H}_{\mathbf{k}}}\end{aligned} \tag{16}$$

where $\mathcal{H}$ is the Reproducing Kernel Hilbert Space (RKHS) of real-valued functions mapping Θ to $\Re$. $\mathbf{k}(\cdot, \cdot)$ implies the kernel function of this space, and $\mathbf{k}(\boldsymbol{\theta}, \cdot)$ maps $\boldsymbol{\theta}$ to the features on the high-dimensional space.

As distributions in the latent space are matched with the Dirichlet prior on the simplex, we choose the information diffusion kernel [24] as the kernel function. This function is susceptible to points near the simplex boundary and has better effects on sparse data. The detailed calculation equation is:

$$\mathbf{k}(\boldsymbol{\theta}, \boldsymbol{\theta'}) = \exp\left(-\arccos^2 \left(\sum_{i=1}^{K} \sqrt{\theta^{(i)} \theta'^{(i)}} \right) \right) \tag{17}$$

When performing distribution matching, we employ the Dirichlet distribution, $\boldsymbol{\alpha'}$ means hyper-parameter, then $\boldsymbol{\theta'}$ can be sampled by the following equations:

$$p(\boldsymbol{\theta'} \mid \boldsymbol{\alpha'}) = Dir(\boldsymbol{\theta'} \mid \boldsymbol{\alpha'}) \triangleq \frac{1}{\mathrm{B}(\boldsymbol{\alpha'})} \prod_{i=1}^{K} \left(\theta'^{(i)} \right)^{\alpha'^{(i)} - 1} \tag{18}$$

where $\theta'^{(i)}$ denotes the value of the i-th dimension of $\boldsymbol{\theta'}$, $\alpha'^{(i)}$ means the hyper-parameter of the i-th dimension of the Dirichlet distribution, $\boldsymbol{\theta'}$ represents a sample sampled from the Dirichlet prior, and $\mathrm{B}(\boldsymbol{\alpha'}) = \frac{\prod_{i=1}^{K} \Gamma(\alpha'^{(i)})}{\Gamma\left(\sum_{i=1}^{K} \alpha'^{(i)}\right)}$.

Given M encoded samples and M samples sampled from Dirichlet prior, MMD could be calculated by the following unbiased estimation:

$$\begin{aligned}\widehat{\mathrm{MMD}}(Q_\Theta, P_\Theta) = &\frac{1}{M(M-1)} \sum_{i \neq j} \mathbf{k}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) \\ &+ \frac{1}{M(M-1)} \sum_{i \neq j} \mathbf{k}(\boldsymbol{\theta'_i}, \boldsymbol{\theta'_j}) - \frac{2}{M^2} \sum_{i,j} \mathbf{k}(\boldsymbol{\theta}_i, \boldsymbol{\theta'_j})\end{aligned} \tag{19}$$

where $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_M\} \sim Q_\Theta$ are the samples collected from the encoder, and Q_Θ is the encoded distribution of samples. $\{\boldsymbol{\theta'}_1, \boldsymbol{\theta'}_2, ..., \boldsymbol{\theta'}_M\} \sim P_\Theta$ are sampled from the prior distribution P_Θ.

4.2 Text Classification Model

In this subsection, we will introduce the text classification module. As depicted in the right panel of Fig. 1, we utilize a two-layer BiLSTM based on the Topic-Aware attention mechanism. Because of the complex context of misleading information, we incorporate corpus-level topic features by this mechanism to obtain richer semantic representation. Then, we use a classifier with two fully connected layers to detect misleading information.

4.2.1 BiLSTM

Bag-of-words representation is sparse, and the typical solution approach to the sparsity problem is computational intelligence [46] like word embedding technology. Word2vec [30] and GloVe [32] utilize words as the smallest unit for training, while the fastText [4] splits words into n-gram subwords to construct vectors.

Considering that there are many out-of-vocabulary words in misleading information, we use the embedding layer initialized by the pre-trained fastText. Suppose the word sequence of a document $d = \{w_1, w_2, ..., w_m\}$, w_i represents the i-th word in the content. After transforming each word to a one-hot vector, the embedding layer could map words to their corresponding vectors $\mathbf{x}_{embed} \in \mathbb{R}^{D_w}$, where D_w is the dimension of embedding space.

Then, we utilize a two-layer BiLSTM to extract semantic features, and each layer contains bidirectional LSTM units. This bidirectional structure implements the semantic contextual representation of misleading information. The network takes $\mathbf{x}_{embed}$ in the order of the content as input and gets each word's hidden state. If the definition of LSTM unit is simplified as LSTM$(\cdot)$, the hidden state $\mathbf{h}'$ of each word could be calculated by:

$$\mathbf{h}_{f1} = \overrightarrow{\mathrm{LSTM}}(\mathbf{x}_{embed}), \quad \mathbf{h}_{b1} = \overleftarrow{\mathrm{LSTM}}(\mathbf{x}_{embed}) \tag{20}$$

$$\mathbf{h}_{fb} = [\mathbf{h}_{f1}, \mathbf{h}_{b1}] \tag{21}$$

$$\mathbf{h}_{f2} = \overrightarrow{\mathrm{LSTM}}(\mathbf{h}_{fb}), \quad \mathbf{h}_{b2} = \overleftarrow{\mathrm{LSTM}}(\mathbf{h}_{fb}) \tag{22}$$

$$\mathbf{h}' = \mathrm{BN}\left([\mathbf{h}_{f2}, \mathbf{h}_{b2}, \mathbf{x}_{embed}]\right) \tag{23}$$

where $\mathbf{h}_{f1}, \mathbf{h}_{f2} \in \mathbb{R}^{D_h}$ are vectors calculated by the forward LSTM, and $\mathbf{h}_{b1}, \mathbf{h}_{b2} \in \mathbb{R}^{D_h}$ are vectors calculated by the backward LSTM. $\mathbf{h}' \in \mathbb{R}^{2 \times D_h + D_w}$ is the hidden state that combines the word embedding and the bidirectional LSTM.

4.2.2 Topic-Aware Attention Mechanism

Generally, the attention mechanism is similar to human behavior when reading a sentence, evaluating how important each word is by giving a weight to each part [50]; the higher value is, the more important the word will be. In the typical attention-based model, the alignment score of each word is calculated as:

$$f(\mathbf{h}') = \mathbf{q}^T \tanh(\mathbf{W}_q \mathbf{h}' + \mathbf{b}_q) \tag{24}$$

where $\mathbf{q} \in \mathbb{R}^{D_h}$ are learnable parameters.

However, typical attention mechanisms could not utilize external information, so we design the Topic-Aware attention mechanism to incorporate topic features while calculating the misleading information representation. In this way, we integrate the neural topic module and the text classification module to train the entire model end-to-end.

The attention weights $\mathbf{a}$ for each word are calculated based on the similarity between the topic embedding $\mathbf{v}_t$ and hidden states $H = \{\mathbf{h}'_1, \mathbf{h}'_2, ..., \mathbf{h}'_L\}$ in the last layer of BiLSTM, where L represents the max sentence length in batch.

Specifically, TA-BiLSTM counts the attention weight a_i based on the alignment score between the hidden state $\mathbf{h}'_i$ and the topic embedding $\mathbf{v}_t$, where $i = \{1, 2, ..., L\}$. We set $D_t = D_h$ and use the following equation to calculate the alignment score:

$$f(\mathbf{h}'_i, \mathbf{v}_t) = \mathbf{v}_t^{\mathbf{T}} \tanh(\mathbf{W}_a \mathbf{h}'_i + \mathbf{b}_a) \tag{25}$$

where $\mathbf{W}_a \in \mathbb{R}^{D_h \times D_h}$ and $\mathbf{b}_a \in \mathbb{R}^{D_h}$ are learnable parameters. The larger the value of $f(\mathbf{h}', \mathbf{v}_t)$, the greater the probability of misleading information implied by the corresponding word. Then, the document representation could be summarized based on the alignment scores above:

$$a^{(i)} = \frac{\exp\left(f(\mathbf{h}'_i, \mathbf{v}_t)\right)}{\sum_{j=1}^{L} \exp\left(f(\mathbf{h}'_j, \mathbf{v}_t)\right)} \tag{26}$$

$$\mathbf{v}_d = \sum_{i=1}^{L} a^{(i)} \mathbf{h}'_i \tag{27}$$

where $a^{(i)}$ is the weight of the hidden state $\mathbf{h}'_i$ of the i-th word, and $\mathbf{v}_d \in \mathbb{R}^{D_h}$ contains both semantics of hidden states and topic information embedded by the neural topic model.

4.2.3 Classifier

In this paper, the text which contains misleading information is taken as a positive example. We apply two fully connected layers and a sigmoid activation function to convert the document representation $\mathbf{v}_d$ into the probability for classification. Therefore, the higher value of the output, the more possible this document containing misleading information. The prediction process could be defined as:

$$\mathbf{h}_d = \mathrm{BN}(\mathbf{W}_d \mathbf{v}_d + \mathbf{b}_d) \tag{28}$$

$$\mathbf{o}_d = \max(\mathbf{h}_d, leak * \mathbf{h}_d) \tag{29}$$

$$\hat{y} = \sigma(\mathbf{W}_c \mathbf{o}_d + b_c) \tag{30}$$

where $\mathbf{W}_d \in \mathbb{R}^{D_m \times D_h}$, $\mathbf{b}_d \in \mathbb{R}^{D_m}$, $\mathbf{W}_c \in \mathbb{R}^{D_m}$ and $b_c \in \mathbb{R}$ are learnable parameters, and $\hat{y}$ is the predicted probability.

4.3 Training Objective

In multi-task learning framework, models are optimized for multiple objectives jointly. Our proposed framework mainly has two training objectives: neural topic modeling objective and misleading information detection objective.

For the neural topic modeling, its objective includes the reconstruction term and the MMD based regularization term. It is defined as follows:

$$\begin{aligned} \mathcal{L}_t &= \mu \cdot \mathbb{E}_{P_{\mathbf{x}_{bow}}} \mathbb{E}_{Q(\theta|\mathbf{x}_{bow})} \mathrm{c}(\mathbf{x}_{bow}, \mathbf{x}_{re}) + \mathcal{D}_\Theta \\ &= -\mu \cdot \sum_{i=1}^{V} x_{bow}^{(i)} \log x_{re}^{(i)} + \widehat{\mathrm{MMD}}(Q_\Theta, P_\Theta) \end{aligned} \tag{31}$$

where $\mathrm{c}(\mathbf{x}_{bow}, \mathbf{x}_{re})$ is the reconstruction loss, $x_{bow}^{(i)}$ denotes the weight of the i-th word in the vocabulary, and $x_{re}^{(i)}$ denotes the probability of the i-th word in reconstruction distribution. In our implementation, we follow W-LDA and multiply a scaling factor $\mu = 1/(l \log V)$ to balance the two terms, where l indicates the average sentence length in each batch and V indicates the vocabulary size.

For classification objective, we measure the binary cross-entropy between the target label and the predicted output:

$$\mathcal{L}_c = -\frac{1}{N} \sum_{i=1}^{N} \left(y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)\right) \tag{32}$$

where y_i is the ground truth, and $\hat{y}_i$ represents the predicted probability of the i-th document. N means the total number of document in the corpus.

To balance the two task specific objectives, we adopt a dynamic strategy to control the weights of objectives above. The neural topic model is mainly concerned in the early stage, and then we pay more attention to train the classification objective. Thus, the total training objective is formed as:

$$\mathcal{L}_{total} = \lambda \cdot \mathcal{L}_c + \mathcal{L}_t \tag{33}$$

where λ is a hyper-parameter that dynamically balances the two objectives.

We set λ to a slight value in the early stage, allowing the framework to train neural topic model preferentially. Later, we change λ to 1, shifting the focus to multi-task learning, and train the classifier and the neural topic model jointly.

5 Experiments and Results Analysis

5.1 Experimental Setup

5.1.1 Datasets

We conduct experiments on three public datasets about misleading information to evaluate the effectiveness of the proposed TA-BiLSTM model.

Enron Spam [28] is an English public spam dataset compiled in 2006. Ham emails are collected from the mailboxes of six employees in Enron Corporation. Spam messages are obtained from four sources: SpamAssassin corpus, Honeypot project, spam collection of Bruce Guenter, and spam collected by third parties. These emails were sent and received between 2001 and 2005. The dataset consists of six sub-datasets, which are combined into a whole dataset for experiments.

2007 TREC Public Spam [9]. The Text Retrieval Conference (TREC) is a series of seminars, which mainly focuses on the problems and challenges in information retrieval research. The 2007 TREC conference held a spam filtering competition and published this dataset. The dataset includes complete mail information such as sending and receiving addresses, time, HTML code. In the experiments, we retain content in the main body and ignore other information.

Webis-Clickbait-17 [33] contains a total of 19,538 Twitter posts with links from 27 major news publishers in the United States. These posts were published between November 2016 and June 2017. Five annotators from Amazon Mechanical Turk marked whether articles in these links were misleading information. We use the content of articles linked in the post for detection.

Due to noisy data such as blanks and duplicate documents in three datasets, the statistics of preprocessed datasets are listed in Table 2. We arrange 2/3 of the data as the training set and 1/3 of the data as the test set.

5.1.2 Model Configuration

In the experiments, all datasets use package *enchant* to check the spelling of words. Each word is reverted to base form with no inflectional suffixes by the *en_core_web_lg* model of package *spacy*. We utilize package *gensim* to obtain the word embedding matrix and initialize the embedding layer.

For the neural topic model, we set the number of topics K to 50 and the dimension D_s of the fully connected layer in the encoder to 256. The dimension D_t of the topic embedding is equal to the dimension D_h of the hidden state $\mathbf{h}'$. We make Dirichlet prior as sparse as possible and set the Dirichlet hyper-parameter $\boldsymbol{\alpha}'$ to 0.001. The proportion of noise η that adds to topic distribution is defined as 0.1.

For text classification model, we apply 300-dimensional pre-training fastText word embeddings [14], that is, D_w is set to 300. The dropout of the BiLSTM layer is 0.3, and the dimension D_m in the classifier is 64. The weight matrixes in BiLSTM are initialized by orthogonal initialization, and the parameters in the Topic-Aware attention mechanism are initialized by uniform initialization.

During model training, the hyper-parameter λ is set to 1e-8 initially, and when the training reaches the last 20 Epochs, λ is set to 1. Adam optimizer with a learning rate of 1e-4 to train the parameters of the neural topic model and with a learning rate of 5e-5 to train other parameters. The batch size is 16. The computer CPU is Intel Xeon (Skylake) Platinum 8163, and the operating system is Ubuntu 20.04 64-bit. All models are implemented with PyTorch and run on an NVIDIA V100 32G graphic card.

5.1.3 Baselines

We choose Naive Bayes, Support Vector Machine, Decision Tree, Random Forest four machine learning models for comparison.

Naive Bayes [28] is a probabilistic model. By learning the joint probability distribution of the input and output of the training data, the model computes the label with the largest posterior probability of the predicted data.

Table 2 Statistics of three preprocessed datasets

Datasets	Total	Positive	Negative
Enron Spam	27832	13594	14238
2007 TREC Public Spam	49037	27036	22001
Webis-Clickbait-17	19062	4637	14425

Positive samples refer to misleading information, while negative ones are opposite

SVM [8] is a linear binary classification model defined in the feature space. It uses a kernel function to find a hyperplane to separate the two categories, and maximizes the interval between the data and the plane.

Decision Tree [6] adopts a tree structure and uses layered inferences on the data to achieve the final classification, so it has good interpretability.

Random Forest [5] is an ensemble learning method containing multiple decision trees. The model trains each decision tree independently, and the result is determined by the category with the most output of decision trees.

Besides, we also compare our model with following deep learning-based baselines.

BiLSTM uses a BiLSTM network without attention mechanism. The hidden state of words in the document is averaged as the classifier's input.

Attention-BiLSTM uses a BiLSTM network based on a traditional attention mechanism and inputs the classifier after the weighted summation of each word's hidden state.

In the aspect of topic modeling, we compare our model with the following neural topic models.

LDA [1] [3] extracts topics based on the co-occurrence information of words in the document. We use package *gensim* to implement this model.

NVDM [2] [29] comprises an encoder network and a decoder network, inspired by the variational autoencoder based on Gaussian prior distribution.

W-LDA [3] [31] is the prototype of our model, which uses Wasserstein autoencoder and Dirichlet prior distribution to mine topic information.

BAT [44] applies bidirectional adversarial training with Dirichlet prior for neural topic modeling.

The last three neural topic models mentioned above adopt a neural network structure similar to our model.

5.1.4 Evaluation Metrics

In the experiments, we mainly evaluate the classification performance of the text classification model and the topic quality of the neural topic model.

For classification, we compare three widely used performance metrics: accuracy, precision, and F1-score. Accuracy refers to the proportion of correctly classified samples to the total number. The calculation is:

$$Accuracy = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}[(\hat{y}_i = y_i)] \tag{34}$$

where N is the total number of samples, and $\mathbb{I}(\cdot)$ depicts the indicator function. When $\cdot$ is true, the function equals 1; otherwise, it is equal to 0. In binary classification, we generally divide the combination of predicted labels and ground truths into four types, namely True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). True or False means whether the prediction is correct, Positive or Negative means whether the forecast result is a positive or negative sample. These four categories respectively correspond to the number of samples that meet the condition, so the sum of four values equals N. Based on the above, the definition of precision is:

$$Precision = \frac{TP}{TP + FP} \tag{35}$$

$$Recall = \frac{TP}{TP + FN} \tag{36}$$

Precision is the number of correct labels divided by the number of all predicted positive results, and recall is the fraction of true positive samples predicted to be positive. So the precision and recall are a set of contradictory measures. To comprehensively consider the precision and recall metrics, we also evaluate the effectiveness with the F1-score. The definition is below:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2 \times TP}{m + TP - TN} \tag{37}$$

Under the same experimental conditions, the higher above metrics, the better classification performance.

For topic quality, we utilize two standard metrics C_V and C_A of topic coherence [4][34]. Here we choose 10 representative words for each topic as word sets and respectively compute C_V to measure semantical support for one word in each set. Variously, C_A compares pairs of single words in each topic's set to evaluate the coherence between words. To this end, we apply the two metrics to quantify the quality of topic modeling comprehensively.

5.2 Results and Analysis

In this section, we present the experimental results and corresponding analysis of proposed TA-BiLSTM model in terms of classification performance and topic quality.

[1] https://github.com/RaRe-Technologies/gensim

[2] https://github.com/ysmiao/nvdm

[3] https://github.com/awslabs/w-lda

[4] https://github.com/dice-group/Palmetto

Table 3 Misleading information detection performance on the three datasets

Models	Enron spam			2007 TREC			Clickbait-17		
	Accuracy	Precision	F1-score	Accuracy	Precision	F1-score	Accuracy	Precision	F1-score
Naive Bayes	0.9628	0.9805	0.9612	0.7665	0.9774	0.7373	0.7460	0.4642	0.3568
SVM	0.9737	0.9557	0.9736	0.9780	0.9706	0.9804	0.7173	0.4209	0.4270
Decision Tree	0.9359	0.9319	0.9345	0.9758	0.9762	0.9782	0.7334	0.4478	0.4299
Random Forest	0.9665	0.9769	0.9652	0.9771	0.9908	0.9790	0.7779	**0.6547**	0.3046
BiLSTM	0.9829	0.9762	0.9827	0.9781	0.9744	0.9803	0.7524	0.4940	0.4551
Attention-BiLSTM	0.9838	0.9764	0.9837	0.9810	0.9784	0.9830	0.7764	0.5593	0.4743
TA-BiLSTM	**0.9901**	**0.9880**	**0.9899**	**0.9920**	**0.9937**	**0.9928**	**0.8006**	0.6501	**0.4978**

The first four items are machine learning models, and the last two items are deep learning models for ablation study
All significant information has been bolded

5.2.1 Classification Performance

Table 3 lists the results of classification performance on three used public datasets compared with different baselines. We could observe that the TA-BiLSTM model could obtain better results in accuracy, precision and F1-score.

Specifically, the bag-of-words representation limits the traditional machine learning approaches. The precision of Random Forest on the Clickbait-17 dataset is higher because the model only selects confirmed positive samples to minimize the number of FP. Therefore, the accuracy of Random Forest is not high, and the F1-score is lower than other approaches.

Moreover, we conduct ablation study by comparing BiLSTM and Attention-BiLSTM to verify the outperforming of the Topic-Aware attention mechanism. We could observe that the results are better than those of machine learning-based approaches, indicating that richer semantic feature representation, especially context information, could improve classification performance. Compared with the BiLSTM, the results of Attention-BiLSTM show slight improvements, indicating that the attention mechanism assigns more weights to specific words to provide a more suitable document representation.

Furthermore, in the comparison of Attention-BiLSTM and TA-BiLSTM, we observe that accuracy increases 0.64%, 1.12%, 3.11% and F1-score increases 0.63%, 0.99%, 4.95% for the latter on the three datasets, respectively. The significant improvements show that Topic-Aware attention mechanism could incorporate topic information into classification module. Moreover, the topic information could indeed help TA-BiLSTM to provide more suitable representations for misleading information detection.

5.2.2 Topic Quality Comparison

The calculation of attention mechanism often incorporates supervision signal from a document, which will be helpful for mining latent semantic patterns in topic modeling procedure. Thus, we also evaluate the quality of topics in this subsection. Table 4 presents the results of different topic coherence metrics C_A and C_V comparing with other topic modeling baselines on three datasets.

Table 4 Topic coherence scores of various topic models on the three datasets, a higher value means more coherent topics

Models	Enron spam		2007 TREC		Clickbait-17	
	C_A	C_V	C_A	C_V	C_A	C_V
LDA	0.1483	0.3671	0.1468	0.3719	0.2193	0.4096
NVDM	0.1335	0.3614	0.1485	0.3897	0.1216	0.3411
W-LDA	0.1548	0.3910	**0.1503**	0.4230	0.2132	0.4092
BAT	0.1564	0.3835	0.1378	0.3913	0.2298	0.4177
TA-BiLSTM	**0.1638**	**0.4361**	0.1497	**0.4780**	**0.2351**	**0.4305**

All significant information has been bolded

Table 5 Topic models top-10 words of five same topics on the three datasets, where italics indicate irrelevant words to the topic

Datasets	Models	Topics
Enron spam	LDA	research university conference visit presentation program *shall* finance *dear* school conference *insight* attend *industry ken* everybody discussion discuss reading topic state *project account* policy government notice *board* committee *wind update* claim lottery *program* win *international* agent draw prize promotion *address* *est* mortgage *image arm logo* bad *vol coastal heaven* qualify
	W-LDA	resume interview candidate internship summer research intern student job *crenshaw* conference presentation speaker paper professor university *finance* chair fax visit state governor senate legislature assembly *utility* vote committee republican *burton* lottery claim *batch* winner prize congratulation win lucky agent promotional *free click* remove mortgage *opt* removal life refinance money *advertisement*
	BAT	university finance *rice* professor *martin* department school paper site *shall* meeting meet question discuss room schedule agenda hold attend *draft* assembly vote senate state *utility* legislature governor *burton* republican *bankruptcy* lottery win claim agent *batch international address* congratulation *ref avoid* credit mortgage rate bad loan refinance broker *month link* low
	TA-BiLSTM	student university internship school graduate faculty professor *rice* interview summer conference guest hotel registration event speaker room session lunch attend senate committee assembly senator republican legislature governor vote *bond burton* award lottery prize lucky winning agent winner claim *international* win mortgage loan refinance qualify lender rate bad removal *unsubscribe* consultation
2007 TREC	LDA	weather shower *credit* sunny map thunder *deal* forecast cloudy wind win club *lewis* race sport lead beat *grand* round compete int static method void create object class patch *parrot the* *win* office suite edition pro cloud flash creative *undefined* acrobat *package life* model *private* error *now version feel lead* estimate
	W-LDA	sunny cloudy *variable* forecast cloudiness alert shower weather map *subscribe* football *formula* golf sport championship win cup *fantasy* athletic victory int void modify *samba* static branch node domain unsigned null acrobat professional pro studio office creative illustrator suite premiere professionally plot *apply size* function output model *version* error median efficient
	BAT	sunny weather cloudy shower map forecast cloudiness period program *jun* *(Do not appear)* int void static *context bullish status* result program null flag acrobat professional office suite creative pro studio vista premiere illustrator model matrix *package mixed* residual function random variance estimate datum
	TA-BiLSTM	sunny cloudy weather precipitation cloudiness hourly shower forecast temperature wind golf football tour cycling playoff league cup athletic victory race int void static node unsigned lock branch merge recovery daemon premiere illustrator enterprise suite acrobat creative studio professional edition pro variance coefficient linear regression correlation matrix estimate calculate observation vector
	LDA	republican senate vote bill democrat senator committee *sen* congress democratic coach football sport yard final lose player championship title *loss* health care patient medical *food* doctor disease *reduce* mental *increase* flight travel airport airline plane passenger fly *ban board return* charge arrest border crime *car* criminal *driver* prison *county* prosecutor
	W-LDA	repeal republican care bill *health* insurance affordable lawmaker legislation coverage baseball league player club pitch *major* sport fan hit *minor*

Table 5 (continued)

Datasets	Models	Topics
		medication prescription med drug *generic* doctor pharmacy *sexual ship* medicine
		flight airline passenger plane airport pilot fly aircraft carrier crew
Clickbait-17		suspect arrest *officer* injure shoot *authority truck* kill gun wound
	BAT	*health* care insurance bill republican repeal affordable tax vote law
		tournament ball coach player shot basketball final league *seed* guard
		die doctor *condition* cancer hospital medical brain *tweet* surgery staff
		flight airline passenger plane airport pilot *board* fly seat air
		shooting suspect shoot arrest kill gun prison murder charge incident
	TA-BiLSTM	freedom democracy inauguration speech crowd party politician protester supporter protest
		ball baseball basketball player tournament court *supreme* shoot hall shooting
		disease patient cancer medicine medical diagnose drug doctor treatment *study*
		flight airline passenger plane airport pilot aircraft seat crew fly
		prison crime sentence prosecutor drug jail murder inmate convict arrest

The five topics on the Enron Spam dataset are ”college”, ”conference”, ”politics”, ”prize-winning”, and ”loan”, on the 2007 TREC dataset are ”weather”, ”sports”, ”computer”, ”software” and ”mathematics”, and on the Clickbait-17 dataset are ”politics”, ”sports”, ”medicine”, ”flight” and ”crime”

Compared with the topics extracted by W-LDA on Enron Spam dataset, the C_A of TA-BiLSTM has increased by 5.81%, and the C_V metric has risen by 11.53%. On the 2007 TREC dataset, C_A is almost the same as the W-LDA, but the C_V has increased by 13%. We also present the comparison with BAT. It obtains slightly higher than W-LDA and LDA on Clickbait-17, but our model improves C_A and C_V by 2.31% and 3.06%.

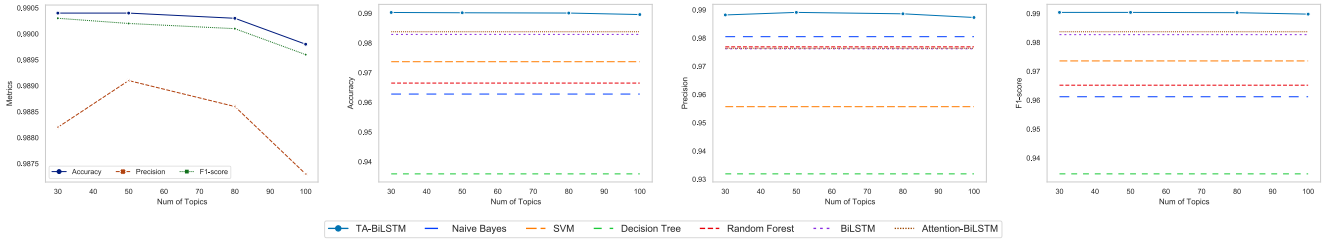
Ignoring NVDM with poor performance, Table 5 lists the top-10 representative words with the highest probability for each topic on three datasets. Thus, we could compare the quality of performance intuitively. Generally, compared with other models, we could realize that the topics generated by TA-BiLSTM have fewer irrelevant words and higher semantic coherence.

The topic words of NVDM are not very consistent because it employs Gaussian prior to mimic Dirichlet in topic distribution space. As the proposed TA-BiLSTM utilizes Dirichlet as prior distribution in topic space, it could obtain coherent topics than NVDM. Meanwhile, the supervision signal also helps the TA-BiLSTM to surpass LDA, W-LDA and BAT in topic modeling evaluation.

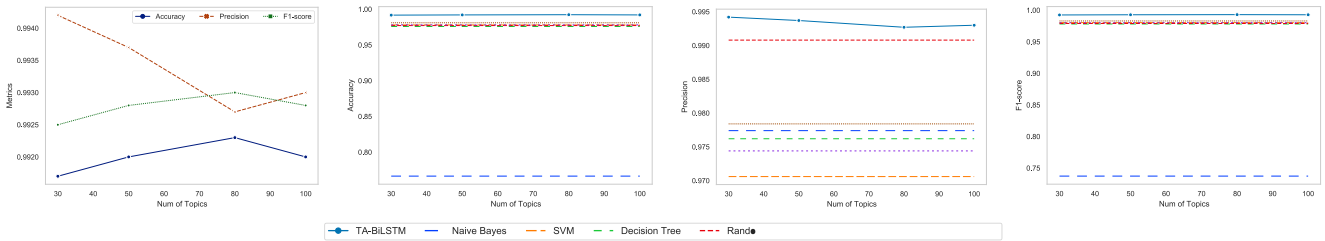
Table 6 Parameter analysis of the number of topics K on three datasets

Datasets	Topics	Accuracy	Precision	F1-score
	30	**0.9904**	0.9882	**0.9903**
	50	**0.9904**	**0.9891**	0.9902
Enron spam	80	0.9903	0.9886	0.9901
	100	0.9898	0.9873	0.9896
	30	0.9917	**0.9942**	0.9925
	50	0.9920	0.9937	0.9928
2007 TREC	80	**0.9923**	0.9927	**0.9930**
	100	0.9920	0.9926	0.9928
	30	0.7937	0.6028	**0.5238**
	50	**0.8047**	0.7010	0.4703
Clickbait-17	80	0.7877	0.6158	0.4505
	100	0.8015	**0.7120**	0.4408

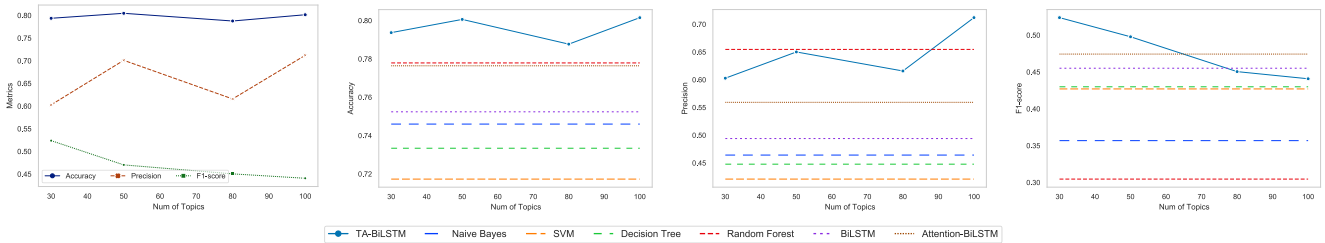
All significant information has been bolded



(a) Enron Spam



(b) 2007 TREC



(c) Clickbait-17

Fig. 2 Illustration of different numbers of topics K on three datasets. Each of these subfigures is constituted by four components. The first one depicts how TA-BiLSTM performance varies with different numbers of topics and others depict the comparison with baselines on three classification metrics Accuracy, Precision and F1-score

Table 7 Parameter analysis of the dimension of hidden states $\mathbf{h}'$ on three datasets

Datasets	Hiddens	Accuracy	Precision	F1-score
Enron Spam	25	0.9900	0.9890	0.9898
	50	**0.9904**	0.9893	**0.9902**
	75	0.9897	0.9848	0.9895
	100	0.9902	**0.9906**	0.9900
	150	0.9885	0.9843	0.9883
2007 TREC	25	**0.9922**	0.9913	**0.9929**
	50	**0.9922**	0.9928	**0.9929**
	75	0.9920	0.9910	0.9928
	100	0.9917	**0.9931**	0.9925
	150	0.9913	0.9914	0.9921
Clickbait-17	25	0.7885	0.5996	0.4882
	50	0.7951	0.6400	0.4724
	75	0.7959	0.6308	0.4916
	100	0.7960	0.6271	**0.4985**
	150	**0.8037**	**0.6832**	0.4811

All significant information has been bolded

Springer

5.2.3 Hyper-Parameter Analysis

To further validate the robustness of TA-BiLSTM, we conduct hyper-parameter analysis in this subsection. Concretely, parameter analysis on three parameters (the number of topics K, the dimension of hidden states $\mathbf{h}'$ and the proportion of noise η) has been carried out.

Firstly, the number of topics K is set to 30, 50, 80 and 100, respectively. The quantitative results on three datasets are reported in Table 6 and visualized in Fig. 2.

For Enron Spam and 2007 TREC datasets, we could observe that TA-BiLSTM performs fairly stable on three metrics. For Clickbait-17 dataset, the classification performance is more sensitive to changes of K, which may be caused by the complicity of the dataset. It is worth mentioning that optimal numbers of topics over datasets are different (50 on Enron Spam, 80 on 2007 TREC and 50 on Clickbait-17). If this number is too large, the model is not interpretable, and if the number is too small, the model training will be negatively affected [12]. Thus, we set the number of topics K to 50 in our experiments.

Similarly, we conduct parameter analysis on the dimension of hidden states $\mathbf{h}'$. It has been set to 25, 50, 75, 100 and 150 respectively. And the corresponding statistics are listed in Table 7. By comparing the results, we could observe that simple models perform better on Enron Spam and 2007 TREC datasets. While dealing with Clickbait-17, classification performance improves with the increasing of model complexity. This may be also caused by the complexity of Clickbait-17 dataset which needs a more complicated model to fit the data.

We further investigate the impact of different proportions of noise η on the performance. In detail, we compute the metrics of classification and topic modeling separately with five proportion settings [0, 0.1, 0.2, 0.3, 0.4]. The detailed comparison is shown in Table 8. It can be concluded that adding a proper proportion of noise to the topic distribution upgrades the quality of topic modeling on all datasets. However, not the optimal parameter for the topic mining has the same consequence on classification performance. Topic coherence is better when the proportion is set to 0.1 or 0.2, while less noise is helpful for the Topic-Aware attention mechanism to preserve topic features and prediction. Hence we set the proportion of noise to 0.1 for better comprehensive results in the experiments.

5.2.4 Case Study and Visualization

To validate that proposed TA-BiLSTM could indeed improve the model interpretability, we conduct case study and visualization in this subsection.

Figure 3a shows an advertising email for an online pharmacy in the Enron Spam dataset. As Topic 8 represents drugs, we could infer that this email may discuss related topics. Also, we could find various drug names appeared in its text content. Likewise, Fig. 3b depicts a web page content from Clickbait-17 which entices people to buy cosmetics. We can also find relevant words from Topic 15 and Topic 45, such as ‘carpet’, ‘fashion’, ‘beauty’, ‘makeup’.

Thus, the above two examples show that corpus-level topic relatedness could really improve model interpretability.

Table 8 Parameter analysis of the proportion of noise η on three datasets

Datasets	Noise	Accuracy	F1-score	C_A	C_V
Enron Spam	0	**0.9904**	**0.9902**	0.1477	0.4132
	0.1	0.9901	0.9899	0.1638	**0.4361**
	0.2	0.9891	0.9889	**0.1678**	0.4285
	0.3	0.9890	0.9989	0.1587	0.4005
	0.4	0.9880	0.9879	0.1588	0.3797
2007 TREC	0	0.9916	0.9924	0.1450	0.4241
	0.1	**0.9918**	**0.9926**	**0.1497**	0.4780
	0.2	0.9912	0.9921	0.1491	**0.4842**
	0.3	0.9909	0.9918	0.1441	0.4487
	0.4	0.9911	0.9919	0.1499	0.4154
Clickbait-17	0	0.7937	0.4816	0.1940	0.3951
	0.1	0.8006	**0.4978**	0.2351	0.4305
	0.2	**0.8014**	0.4711	**0.2493**	**0.4353**
	0.3	0.7871	0.4444	0.2278	0.4275
	0.4	0.7912	0.4505	0.1943	0.3957

All significant information has been bolded

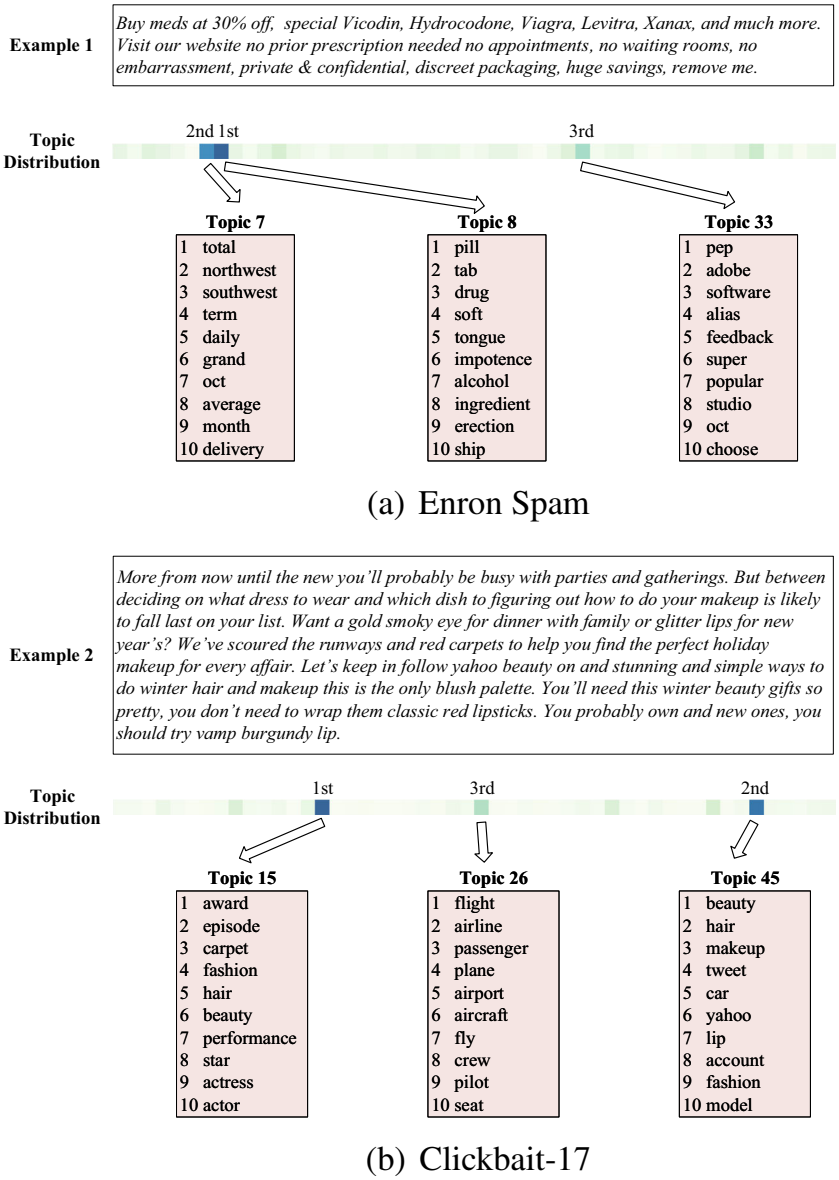


Fig. 3 Case study of two misleading examples from the test sets of Enron Spam (subfigure (a)) and Clickbait-17 (subfigure (b)). Color shade indicates the proportion of topic distribution. A higher proportion in topic distribution will result in a darker color in the figure. Representative top-10 words for crucial topics are listed below the bar

6 Conclusion

In this paper, we proposed the Topic-Aware BiLSTM (TA-BiLSTM) model, an end-to-end framework. TA-BiLSTM contains a neural topic model and a text classification model, which explores corpus-level topic relatedness to enhance misleading information detection. Meanwhile, the supervision signal could be incorporated into topic modeling process to further improve the topic quality. Experiments on three English misleading information datasets demonstrate the superiority of TA-BiLSTM compared with baseline approaches. Additionally, we analyze multiple hyper-parameters in detail and select specific topic examples for visualization. More recently, classification and topic modeling on short texts are still challenging tasks. Our future study would pay more attention to detect misleading information from the short text on social media platforms.

Acknowledgments This work was supported in part by the National Key Research and Development Program (2019YFB2101704 and 2018YFB0803403), National Natural Science Foundation of China (No.61872194, No.62072252 and No.62102192).

References

1. Agrawal A (2016) Clickbait Detection Using Deep Learning. In: 2016 2nd International Conference on Next Generation Computing Technologies (NGCT). IEEE, pp 268–272
2. Biyani P, Tsioutsiouliklis K, Blackmer J (2016) 8 Amazing Secrets for Getting More Clicks: Detecting Clickbaits in News Streams Using Article Informality. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 30

3. Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res 3(Jan):993–1022
4. Bojanowski P, Grave E, Joulin A, Mikolov T (2017) Enriching word vectors with subword information. Transa Assoc Comput Linguist 5:135–146
5. Breiman L (2001) Random Forests. Mach Learn 45:5–32
6. Breiman L, Friedman J, Stone CJ, Olshen RA (1984) Classification and Regression Trees. CRC press
7. Chakraborty A, Paranjape B, Kakarla S, Ganguly N (2016) Stop clickbait: Detecting and preventing clickbaits in online news media. In: 2016 IEEE/ACM International conference on advances in social networks analysis and mining (ASONAM). IEEE, pp. 9–16
8. Chang CC, Lin CJ (2011) LIBSVM: A library for support vector machines. ACM Trans Intell Syst Technol (TIST) 2(3):1–27
9. Cormack GV (2007) TREC 2007 Spam Track Overview. In: In The Sixteenth Text RETrieval Conference (TREC 2007). Proceedings
10. Elhadad MK, Li KF, Gebali F (2020) Detecting Misleading Information on COVID-19. IEEE Access 8:165201–165215
11. Friedman JH (2002) Stochastic gradient boosting. Comput Stat Data Anal 38(4):367–378
12. Gao H, Qin X, Barroso RJD, Hussain W, Xu Y, Yin Y (2020) Collaborative Learning-based Industrial IoT API Recommendation for Software-defined Devices. The Implicit Knowledge Discovery Perspective. IEEE Transactions on Emerging Topics in Computational Intelligence
13. Gao H, Huang W, Duan Y (2021) The Cloud-edge-based Dynamic Reconfiguration to Service Workflow for Mobile Ecommerce Environments: A QoS Prediction Perspective. ACM Trans Internet Technol (TOIT) 21(1):1–23
14. Grave E, Bojanowski P, Gupta P, Joulin A, Mikolov T (2018) Learning word vectors for 157 languages. in: Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018). European Language Resources Association (ELRA)
15. Gretton A, Borgwardt KM, Rasch MJ, Schölkopf B, Smola A (2012) A kernel Two-Sample test. J Mach Learn Res 13(25):723–773
16. Gui L, Jia L, Zhou J, Xu R, He Y (2020) Multi-Task Learning with mutual learning for joint sentiment classification and topic detection. IEEE Trans Knowl Data Eng:1–1
17. Hai-Tao Z, Jin-Yuan C, Yao X, Sangaiah AK, Jiang Y, Zhao CZ (2018) Clickbait convolutional neural network. Symmetry 10(5):138
18. Han X, Li B, Wang Z (2019) An attention-based neural framework for uncertainty identification on social media texts. Tsinghua Sci Technol 25(1):117–126
19. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
20. Hu X, Wang R, Zhou D, Xiong Y (2020) Neural topic modeling with cycle-consistent adversarial training. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp 9018–9030
21. Huang Y, Xu H, Gao H, Ma X, Hussain W (2021) Ssur: an approach to optimizing virtual machine allocation strategy based on user requirements for cloud data center. IEEE Trans Green Commun Netw 5(2):670–681
22. Jain G, Sharma M, Agarwal B (2019) Spam detection in social media using convolutional and long short term memory neural network. Ann Math Artif Intell 85(1):21–44
23. Kumar V, Khattar D, Gairola S, Kumar Lal Y, Varma V (2018) Identifying Clickbait: A Multi-Strategy Approach Using Neural Networks. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp 1225–1228
24. Lafferty J, Lebanon G (2002) Information Diffusion Kernels. In: Proceedings of the 15th International Conference on Neural Information Processing Systems, pp 391–398
25. Li G, Peng S, Wang C, Niu J, Yuan Y (2018) An energy-efficient data collection scheme using denoising autoencoder in wireless sensor networks. Tsinghua Sci Technol 24(1):86–96
26. Liu L, Lu Y, Luo Y, Zhang R, Itti L, Lu J (2016) Detecting "Smart" Spammers on Social Network: A Topic Model Approach. In: Proceedings of the NAACL Student Research Workshop, pp 45–50
27. Luong MT, Pham H, Manning CD (2015) Effective Approaches to Attention-based Neural Machine Translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp 1412–1421
28. Metsis V, Androutsopoulos I, Paliouras G (2006) Spam filtering with naive Bayes-Which naive bayes? In: CEAS, vol 17, pp 28–69. Mountain View, CA
29. Miao Y, Yu L, Blunsom P (2016) Neural variational inference for text processing. In: International Conference on Machine Learning. PMLR, pp 1727–1736
30. Mikolov T, Sutskever I, Chen K, Corrado G, Dean J (2013) Distributed Representations of Words and Phrases and their Compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems, vol 2, pp 3111–3119
31. Nan F, Ding R, Nallapati R, Xiang B (2019) Topic modeling with wasserstein autoencoders. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics. Florence, Italy, pp 6345-6381
32. Pennington J, Socher R, Manning CD (2014) GloVe: Global Vectors for Word Representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp 1532–1543
33. Potthast M, Gollub T, Komlossy K, Schuster S, Wiegmann M, Fernandez EPG, Hagen M, Stein B (2018) Crowdsourcing a large corpus of clickbait on twitter. In: Proceedings of the 27th International Conference on Computational Linguistics, pp 1498–1507
34. Röder M, Both A, Hinneburg A (2015) Exploring the Space of Topic Coherence Measures. In: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, pp 399–408
35. Song L, Lau RYK, Kwok RCW, Mirkovski K, Dou W (2017) Who are the spoilers in social media marketing? Incremental learning of latent semantics for social spam detection. Electron Commerce Res 17(1):51–81
36. Srivastava A, Sutton C (2017) Autoencoding variational inference for topic models. In: International Conference on Learning Representations
37. Tolstikhin I, Bousquet O, Gelly S, Schoelkopf B (2018) Wasserstein Auto-Encoders. In: International Conference on Learning Representations
38. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Polosukhin I (2017) Attention is All you Need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp 6000–6010
39. Villani C (2003) Topics in Optimal Transportation, vol 58. American Mathematical Society
40. Wallach HM, Mimno D, McCallum A (2009) Rethinking LDA: Why Priors Matter. In: Proceedings of the 22nd International Conference on Neural Information Processing Systems, pp 1973–1981

41. Wang C, Wang B (2020) An End-to-end Topic-Enhanced Self-Attention Network for Social Emotion Classification. In: Proceedings of The Web Conference, vol 2020, pp 2210–2219
42. Wang R, Deyu Z, He Y (2019a) Open Event Extraction from Online Text using a Generative Adversarial Network. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp 282–291
43. Wang R, Zhou D, He Y (2019b) ATM: Adversarial-Neural Topic Model. Inf Process Manag 56(6):102098
44. Wang R, Hu X, Zhou D, He Y, Xiong Y, Ye C, Xu H (2020) Neural topic modeling with bidirectional adversarial training. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics. Online, pp 340–350
45. Wang Y, Huang M, Zhu X, Zhao L (2016) Attention-based LSTM for Aspect-level Sentiment Classification. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp 606–615
46. Yang X, Zhou S, Cao M (2020) An approach to alleviate the sparsity problem of hybrid collaborative filtering based recommendations: The product-attribute perspective from user reviews. Mob Netw Appl 25(2)
47. Yang Z, Yang D, Dyer C, He X, Smola A, Hovy E (2016) Hierarchical attention networks for document classification. Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp 1480–1489
48. Yin Y, Cao Z, Xu Y, Gao H, Li R, Mai Z (2020) Qos Prediction for Service Recommendation With Features Learning in Mobile Edge Computing Environment. IEEE Trans Cogn Commun Netw 6(4):1136–1145
49. Zhou D, Hu X, Wang R (2020) Neural topic modeling by incorporating document relationship graph. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp 3790–3796
50. Zhu Y, Zhang W, Chen Y, Gao H (2019) A novel approach to workload prediction using attention-based lstm encoder-decoder network in cloud environment. EURASIP J Wirel Commun Netw 2019(1):1–18