Engineering Applications of Artificial Intelligence 91 (2020) 103595

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

A CLSTM-TMN for marketing intention detection[☆,☆☆]

Yufeng Wang[a], Kun Ma[a,b,*], Laura Garcia-Hernandez[c], Jing Chen[a], Zhihao Hou[a], Ke Ji[a,b], Zhenxiang Chen[a,b], Ajith Abraham[d]

[a] School of Information Science and Engineering, University of Jinan, Jinan 250022, China
[b] Shandong Provincial Key Laboratory of Network Based Intelligent Computing, University of Jinan, Jinan 250022, China
[c] Department of Rural Engineering, University of Cordoba, Spain
[d] Machine Intelligence Research Labs, Scientific Network for Innovation and Research Excellence, Auburn, USA

ARTICLE INFO

Keywords:
Text classification
Marketing intention
Topic memory
News

ABSTRACT

In recent years, neural network-based models such as machine learning and deep learning have achieved excellent results in text classification. On the research of marketing intention detection, classification measures are adopted to identify news with marketing intent. However, most of current news appears in the form of dialogs. There are some challenges to find potential relevance between news sentences to determine the latent semantics. In order to address this issue, this paper has proposed a CLSTM-based topic memory network (called CLSTM-TMN for short) for marketing intention detection. A ReLU-Neuro Topic Model (RNTM) is proposed. A hidden layer is constructed to efficiently capture the subject document representation, Potential variables are applied to enhance the granularity of subject model learning. We have changed the structure of current Neural Topic Model (NTM) to add CLSTM classifier. This method is a new combination ensemble both long and short term memory (LSTM) and convolution neural network (CNN). The CLSTM structure has the ability to find relationships from a sequence of text input, and the ability to extract local and dense features through convolution operations. The effectiveness of the method for marketing intention detection is illustrated in the experiments. Our detection model has a more significant improvement in F1 (7%) than other compared models.

1. Introduction

1.1. Background and challenges

In the context of more and more extensive news dissemination, more and more marketing news is being released by some manufacturers. Marketing pop-up ads and news webpages have brought a lot of trouble to people's lives, it affects in a positive or a negative way the interest of the possible consumers towards the advertised product (Madera et al., 2016). We need to increase the efficacy of an advertising text (Madera et al., 2017). It is especially important to accurately identify marketing news.

Text classification is a work to assign categories to text by its content (Zhou et al., 2015). There are various downstream applications such as spam detection (Zhang et al., 2019), sentiment analysis (Zhang et al., 2018c), and malicious language attacks (Selvam, 2018). Marketing intention detection is to process and analyze such news text automatically and make marketing classification or clustering by characteristics (Wang et al., 2019b). Current deep learning methods attempt to achieve remarkable performance to address this issue in a new way. For example, convolutional neural network (CNN) (Kim, 2014) and recurrent neural network (RNN) (Zaremba et al., 2014) are used for such text classification tasks, which adopt totally different ways of understanding natural languages. CNN is capable to learn local response from temporal data but has a lack on learning sequential correlations (Kim, 2014). RNN is capable to specialize for sequential modeling but has a lack on extract features in a parallel way (Zaremba et al., 2014). In an improved way of long short-term memory recurrent neural network (LSTM) (Yao et al., 2019b), CLSTM uses the locality and density of the convolution operation to store information and learn the time structure. This is proved to be demonstrated their success in

☆ This document is the results of the National Natural Science Foundation of China (61772231), the Shandong Provincial Natural Science Foundation, China (ZR2017MF025), and Project of Shandong Provincial Social Science Program, China (18CHLJ39).

☆☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.engappai.2020.103595.

* Corresponding author at: School of Information Science and Engineering, University of Jinan, Jinan 250022, China.
E-mail addresses: 464017311@qq.com (Y. Wang), ise_mak@ujn.edu.cn (K. Ma), ir1gahel@uco.es (L. Garcia-Hernandez), 454758614@qq.com (J. Chen), 1610176255@qq.com (Z. Hou), ise_jik@ujn.edu.cn (K. Ji), czx@ujn.edu.cn (Z. Chen), ajith.abraham@ieee.org (A. Abraham).

https://doi.org/10.1016/j.engappai.2020.103595
Received 2 December 2019; Received in revised form 12 February 2020; Accepted 2 March 2020
Available online xxxx
0952-1976/

processing formal and well-edited texts on sentiment classification and question classification tasks (Zhou et al., 2015). However, the performance is inevitably compromised when directly applied to informal and short texts to analyze semantic meaning.

Probabilistic models, such as latent semantic analysis (LSA) (Hofmann, 1999) and latent Dirichlet allocation (LDA) (Hoffman et al., 2010), are one of the great success stories of unsupervised learning for text classification. However, inference methods of topic models have become increasingly complex with more data (Ma et al., 2019). Motivated by the observations, topic memory network (TMN) was proposed to identify the indicative words for classification using topic models (Zeng et al., 2018). But it remains much scope for improvement. In this paper, we attempt to improve TMN to construct CLSTM on Classifier part. The text feature vector is further extracted by increasing the position of the word. The performance of the improved model is compared with other models in the experiment section. The experimental results show that our model can effectively improve the functioning of text classification.

Recently, there are many new classification methods in processing text, such as ensemble learning (Zhang and Wang, 2016; Wang et al., 2019b) and hidden Markov models (Kang et al., 2018). However, these methods have inferior performance on short texts with limited features available for classifiers due to the sparsity of the short text data. Sequential correlations and semantic relationships of text are difficult to capture. To establish stronger semantic relationships, some methods rely on feature extraction to enhance semantic information, which may be not clear due to the problem of data sparsity. Topic modeling technology is widely used to construct potential semantics and infer keywords. Due to the sparsity of short texts, documents and words are represented as a unified whole, Deep learning can be used to construct semantic expressions and topic modeling technology can be used to explain hidden variables.

1.2. Contributions

The inspiration of this method comes from the fact that the text features represented by the topic model can improve the performance of text classification, and then the classification model that can effectively initialize the features and space can be used to identify the marketing intention. And our contribution mainly focuses on two parts, one is the classifier model, and the other is the topic extraction method The contributions of our work are extract topics with RNTM and CLSTM-TMN marketing intention filtering model.

- CLSTM-TMN. The CLSTM-TMN model for text classification based on TMN was proposed. CLSTM has the feature of storing historical information. The maximum pooling layer is built in the lower part of CNN to obtain the maximum value of feature map. When LSTM is used to obtain context information semantics, the problem of gradient disappearance can be solved.
- RNTM. A novel Neural Topic Model based on NTM is proposed. The modified linear unit layer in the model extracts topics, and the topics and documents are represented as a whole. This method is not easy to cause the loss of key information. The output of some neurons in RNTM is 0, which reduces the interdependence of parameters and the occurrence of overfitting.

1.3. Organization

The rest of the paper is organized as follows: Section 2 introduces the relevant work on deep learning. Section 3 is the introduction of the three components of the overall process and its methods, (1) Topic extraction (described in Section 3.1), (2) Topic Memory network (described in Section 3.2), and (3) CLSTM (described in Section 3.3). The acquisition of the data set and the configuration of the experiment are described in this Section 4. In order to verify the validity of the proposed model, the deep learning-based model proposed in this paper is compared with some topic models with excellent performance, and the popular topic extraction method is compared with the neural-topic model in this Section 5. Section 6 summarizes and discusses the research content of this paper.

2. Related work

Marketing intention recognition is a kind of text classification. The purpose of topic extraction is to condense features for classification. Our work in this chapter is mainly based on two aspects of analysis: text classification and topic model.

2.1. Text classification

Text categorization can be broadly divided into three categories: traditional machine learning methods; ensemble learning; and deep learning. This section is developed in order of three categories.

Machine learning (Sebastiani, 2002) has achieved remarkable achievements in text classification in recent years. The fast algorithm QKNN (Chen, 2018) is proposed to query the nearest k adjacent data, greatly reducing the similarity calculation. The algorithm can improve the calculation speed of the algorithm and speed up the search time of k adjacent data. A new type of active learning machine learning method (Goudjil et al., 2018) was proposed. This method selects a batch of data samples using the posterior probability provided by the multi-layer SVM classifier, and then intelligently selects efficient samples to mark, reducing the intensity of the annotation work. But they are sensitive to missing data.

Ensemble learning (Kowsari et al., 2019) is widely used in classification tasks. A text classification framework combining word vectors and AE1-WELM was proposed. This method assigns different weights to different samples, and this method can improve the classification accuracy to some extent. An improved performance grouping (PBagging) algorithm was proposed (Zhao et al., 2008). The algorithm relies on reliability as a measure of the quality of classification learning. The news corpus of Kyoto News Agency is classified using the PBagging method. PBagging has a higher accuracy rate than Bagging. The ensemble approach is fairly robust and scalable when dealing with anomalous data (Xi et al., 2018). The effect of text categorization depends on the characteristics, and short texts have the characteristics of sparse data and features that are difficult to express. Feature representation is very difficult because of the short nature of the news text, resulting in the ineffectiveness of classification (Li et al., 2018).

In recent years, deep learning methods have been used for text categorization in order to solve the above problems (Lai et al., 2015; Joulin et al., 2016). Deep learning has performed well in text categorization due to the strong learning ability of neural networks. For example, Some improved models based on convolutional neural networks (CNN) (Yao et al., 2019b) and recurrent neural networks (RNN) (Yao et al., 2019a), long-term and short-term memory (LSTM) (Zhang et al., 2018a) and other neural network models.

The DSCNN model was proposed by Zhang et al. (2016). Its idea is that a depth-sensitive convolutional neural network is used to model documents and sentences. Its process is to use LSTM to pre-process word embeddings. In the first process, The depth-sensitive convolutional network is used for feature extraction optimization. In the second process, the softmax function is used for object classification at the end. The RCNN model was improved by Fang et al. (2019). The output vector of the model is composed of the word vector of the network, the forward output and the backward output, After processing by the pooling layer, the Softmax function is used for object classification. AC-BiLSTM was proposed by Liu and others (Liu and Guo, 2019). The convolutional layer in AC-BiLSTM is utilized to extract high-level phrase representation from the word embedding vector, and the BiLSTM layer is utilized to access the before–after context representation, The attention mechanism is utilized to focus on different

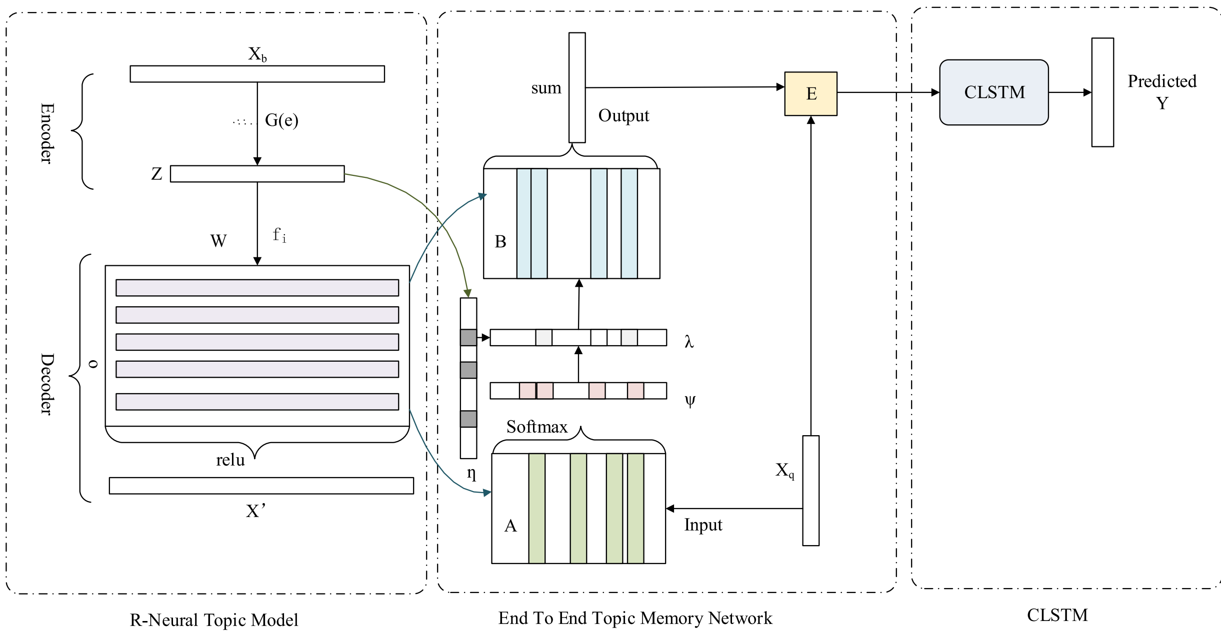

Fig. 1. The overall framework of our CLSTM-TMN model. The dotted box from left to right shows the topic extraction model, topic memory mechanism, and CLSTM classifier.

information in the BiLSTM hidden layer, Finally, the softmax function is used to analyze and categorize the target. AC-BiLSTM can capture not only the global semantic information of a sentence, but also the local characteristics of a word.

CNN performs well in dealing with high dimensional data, and CNN can perform feature extraction automatically. However, when the network layer of the CNN is deep, the training result is easy to fall into a local optimum, and the correlation between the local and the whole is ignored. As an improvement of RNN, LSTM can increase the temporal relationship between elements. However, short-range information between sentences cannot be captured by LSTM. In order to solve the above problems, combined with CNN and RNN, CLSTM utilizes CNN to extract a sequence of higher-level phrase representations, and are fed into a LSTM to obtain the sentence representation (Zhou et al., 2015). Besides, neural network methods are time- consuming in processing large-scale data.

2.2. Topic extraction

Methods of topic extraction mainly include traditional feature extraction methods and deep learning extraction methods, which will be analyzed in this section.

The effective identification of topic models is the key to categorize text correctly. Feature sparsity is considered to be the biggest difficulty in the text processing step, extracting potential topics can make up for the lack of data sparseness. In recent years, many methods of topic extraction have been proposed. One of the more representative topic models is the probabilistic latent semantic index (Landauer et al., 2013), the LSA (Geeganage and Tharanga, 2018) method is proposed to establish an implicit semantic space, but the LSA model cannot statistically predict the probability and cannot update the model when a new model appears. The probabilistic topic model PLSI (Fang et al., 2018; Hofmann, 2017) was proposed, the algorithm is used to fit specific domain synonyms and polysemous words in the training text, but the PLSI model did not provide a uniform probability model. The LDA (Wang and Xu, 2018; Yuan et al., 2015) model mines semantic information hid under text by introducing hyperparameters, but LDA-based models require prior probabilities, which are difficult to determine and LDA is more suitable for classification of large data sets.

Recently, Deep learning has become as a strong machine learning technique that learns multiple layers of representations or features of the data (Zhang et al., 2018c). Documents are modeled by blending topics such as VAE (Kipf and Welling, 2016). However, deep learning also has some problems in extracting feature words, each dimension of distributed representation is abstract and cannot be explained in detail (Young et al., 2018). Current topic extraction methods and deep learning methods exhibit advantages and disadvantages in both documents and vocabulary (Cao et al., 2015). The ability to combine topic models and neural networks with appropriate combinations can efficiently extract topic models and provide detailed probabilistic interpretations of potential variables (Zeng et al., 2018).

In our previous work (Ma et al., 2019), the topic extraction method is inspired by NTM (Miao et al., 2017), which has the advantages of neural network model and probability topic model. However, the activation layer of the model is prone to disappearing gradients, resulting in loss of information and inability to complete deep network training (Athalye et al., 2018). We propose a stream-based live probabilistic topic computing and matching for public opinion monitoring. This model jointly explores topic inference and text classification with memory networks in an end-to-end manner (Zeng et al., 2018).

3. CLSTM-TMN

In this section, we describe our marketing intention identification model (RLSTM-TMN), the overall architecture of which is shown in Fig. 1. There are three main components: (1) Topic extraction model (RNTM) to extract potential topics; (2) A memory network mechanism to map inferred potential topics to feature space; (3) A text classifier (CLSTM) to identify marketing intention.

The text is represented by X, and $\mathrm{X} = \{\mathrm{X}_1, \mathrm{X}_2, \ldots \mathrm{X}_\mathrm{N}\}$ are represented in two forms, including the Bag-of-words (Yao et al., 2019b) and seq2seq (Wu et al., 2019) forms, which are represented as $X_b \in E^s$ and $X_q \in E^l$ respectively. E is a feature vector with classification characteristics, s is the size of the vocabulary, and l is the length of the sequence. X_b is entered into the R-Neural Topic Model to generate the topic. The formed topic is further matched with the embedded X_q to learn the classification characteristics of the topic memory mechanism. Classifiers are used to predict results.

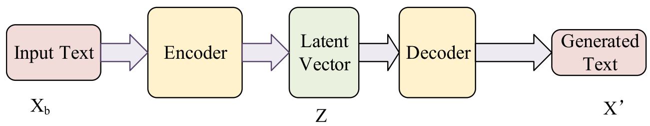

Fig. 2. Encoding-decoding mechanism.

3.1. R-neural topic model

The reasoning process of the model we propose will be described. As shown in Fig. 2, let X as the input of the encoder, Z as the intermediate vector, and X' as the output of the decoder. The context vector is constructed to obtain a single word in the original sequence, and each hidden state of the encoder is scored by the neural network during the context construction process. The probability of the encoder's hidden state is formed in the process of normalization using the Sigmoid function, and the probability is referenced to calculate the weighted sum in the final process. The length of the input and output is not fixed during the machine learning process, and the curve function is required to map X' with the intermediate variable Z. A certain amount of training data is input during the encoding process, and data X is input through the encoder for automatic learning. Different data are mapped from the high latitude vector space to the low latitude vector space by the encoder, and the features of the low latitude vector space are interpreted by the decoder, and the spatial feature vectors of the same dimensional data are output.

Each document has a different topic. The formula is as follows (1). Suppose o is a document and m is a word in o. Topic models calculate the conditional probability distribution $p(m|o)$ as the combination of word-topic and topic-document.

$$p(m|o) = \sum_{i=1}^{L} q\left(m|f_i\right) d\left(f_i|o\right) \tag{1}$$

$$\psi(m) = \left[q\left(m|f_1\right), \dots, q\left(m|f_l\right)\right] \tag{2}$$

$$\eta(o) = \left[q\left(f_1|o\right), \dots, d\left(f_L|o\right)\right] \tag{3}$$

Where $\mathrm{f_i}$ is a latent topic and L is the pre-designed topic number, where ψ is shared among the corpus and η is a specific document. Based on formula (2), we can explain topic models from the view of a neural network, where $\psi(m)$ functions expressed as the query layer for words with the relu activation layer, and $\eta(o)$ as the layer for documents with the relu activation layer. The output of the R-Neural Topic Model is calculated as the scalar product of $\psi(m)$ and $\eta(p)$.

$$p(m|o) = \psi(w) \times \eta^T(o) \tag{4}$$

3.2. End to end topic memory network

Topic memory network maps the topic model generated by the first layer to the feature set for matching, and the two memory unit matrices A and B generated by the Sigmoid activation layer. The text content is encoded as A vector by the Input module, which will be written into the memory unit as the Input of the Update module, and then read and write the two memory modules A and B. According to X_q, the Output will weight the content of the storage unit and accumulate the correlation degree between the memory module and X_q to get the Output vector. As shown in Fig. 3, the topic memory neural network consists of four modules.

- Input module: The way in which the input content is converted to an internal feature representation. The input module adopts the method of encoding the position, and the word vector of each word is weighted and summed depending on the weight of different positions to get the sentence representation. Where X stands for $\mathrm{X} = \left\{\mathrm{X_1}, \mathrm{X_2}, \dots \mathrm{X_N}\right\}$ is stored in memory, loc is the position information vector, in order to increase the text timing information, the index sequence needs to be added in the A matrix. O represents the memory vector matrix of the output module.

$$M_i = \sum_j loc_j \bullet Ax_{ij} + O_A(i) \tag{5}$$

- Update module: In the step of the generalization phase, new content is entered to update the previous memory, which is compressed for future use of memory.
- Output module: The new expression is output in the presentation space according to the state of the current memory. The two memory modules A and B generated by the above input module. A is used for the calculation of the problem, and the correlation between the problem and A single memory matrix is obtained. B is used for the calculation of the correlation between the problem and A, and the output of the answer is obtained. After encoding X_q into vector through the input module, it is the same as A dimension. And its similarity with each A dot product is obtained to obtain the similarity of two vectors. After normalization through a sigmoid function, the weight of X_q and a single memory matrix is obtained. S indicates internal state, The weight formula is as follows:

$$p_i = \mathrm{Sigmoid}\left(S^T A\right) \tag{6}$$

Output vector P is the accumulation of information in storage memory unit and $\mathrm{X_q}$. Di represents a single memory in Output, where P can be expressed as:

$$P = \sum_i M_i d_i \tag{7}$$

- Response module: The output expression is converted to the required content format during the text response phase. Output generates the sum of memory units according to X_q, and Response module mainly generates the final answer based on these information. It combines the sum of two vectors X_o and X_q with the classifier to generate the probability of each word through a software function and select the result of the highest value. The cross entropy loss function is invoked as the objective function for training.

3.3. CLSTM architecture

The CLSTM model is shown in Fig. 4. LSTM performs better in feature extraction of text sequence than RNN model (Zhang et al., 2018b). In the text classification task based on CNN, there is a simple strategy to further extract local feature-maximum pooling from the output of CNN. The CLSTM model proposed in this paper takes the output vector of multi-layer CNN model as the input vector of LSTM, and establishes a LSTM model at the bottom of multi-layer CNN to further extract the features of the CNN model.

The convolutional layer filter is represented as $P \in M^{\zeta \times \upsilon}$, ζ is represented as the number of words in the vector window, υ is represented as the dimension size of the word embedding vector. M represents the matrix for each phase. a_γ is represented as a feature vector, which is ω-dimensional. γ is represented as an index value in a text sequence. $A \in M^{\omega \times L}$ is the input matrix of the CNN, ω is the dimension of the eigenvector, L is equal to the text sequence length. The convolutional layer $P = \left[P_0, \dots, P_{\omega-1}\right]$ will create one new value γ at times goes on, The formula is:

$$O_{P_\gamma} = \mathrm{Re}\,LU\left[\left(\sum_{i=0}^{\omega-1} a_{\gamma+i}^{L} P_i\right) + \sigma\right] \tag{8}$$

Where P and σ are the parameters of the filter layer, and σ is a bias.

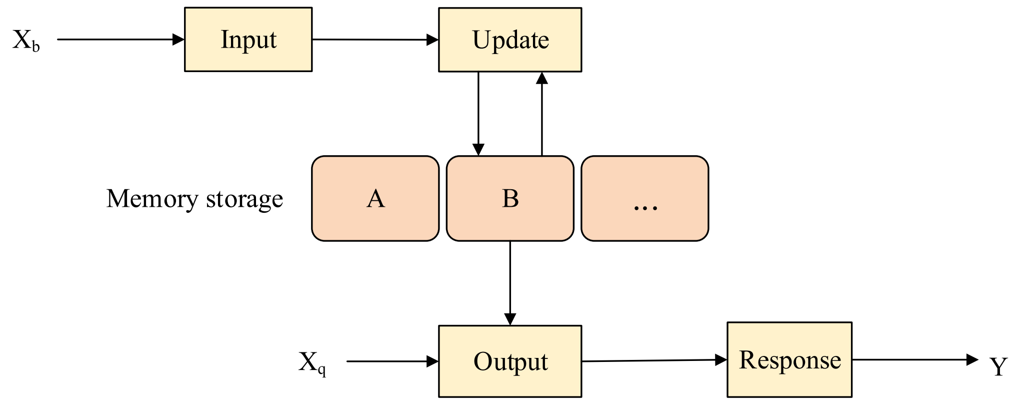

Fig. 3. Topic memory-network.

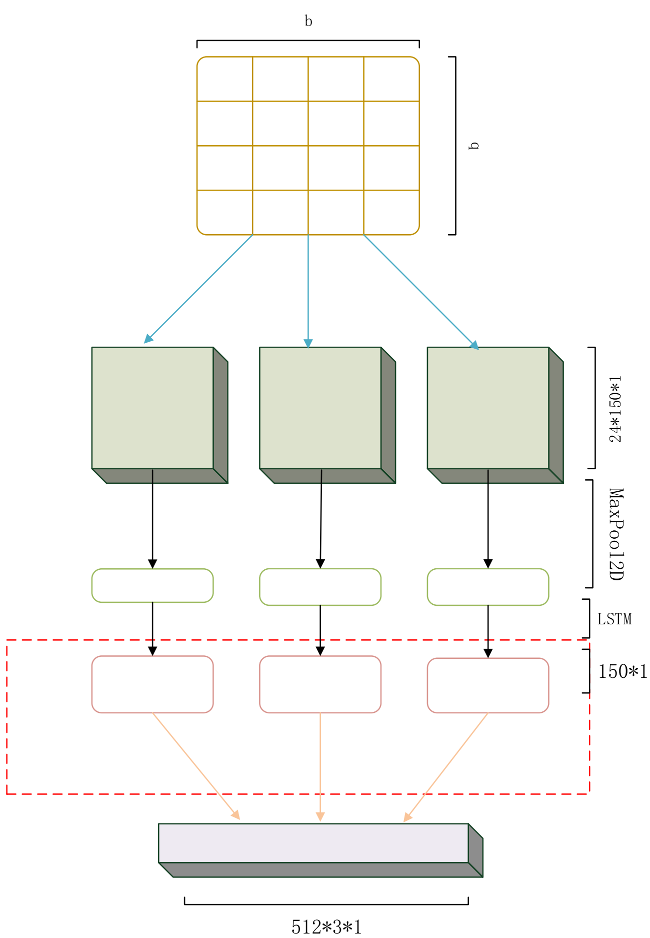

Fig. 4. Structure CLSTM.

The LSTM model is used to obtain the word sequence for each feature, the input is $A = [a_1, a_2, \dots, a_L]^T$, L is equal to the length of the sequence of text vectors and it is the number of iterations of the LSTM, The number of LSTM hidden layer nodes is equal to the length of the vector. The CLSTM model first comes to local perception through the convolutional layer, which is built on top of the pooling layer. The optimal value of the feature matrix is extracted through the maximum pooling layer operation, and the optimal value usually represents the most significant information in the feature set. The model presented in this paper uses a multi-layer convolutional layer with different parameters, and a number of different features are initialized by a multi-layer convolutional layer. The largest pooling layer is passed through these feature matrices and passed to the fully connected layer, and the softmax function is used to obtain the corresponding class probability distribution. The probability to classify α as category ζ is below, where ψ is a coefficient and β is represented as a function value.

$$p\left(\beta^{(i)} = \zeta | \alpha^{(i)}; \psi\right) = \frac{e^{\theta_\zeta^L \alpha^{(i)}}}{\sum_{v=1}^{V} g^{\psi_\zeta^L \alpha^{(i)}}} \tag{9}$$

We use relu as activation function. The formula is:

$$P(\alpha) = \max(0, \alpha) \tag{10}$$

4. Experiments setup

4.1. Dataset

In order to verify the accuracy of our experiment, the data set of Sohu content algorithm contest (Wang et al., 2019b; Sun et al., 2018) and AG's corpus of news articles were used in the experiment (Shen et al., 2019; Wang et al., 2019a). In the experiment, the GLOVE (Rezaeinia et al., 2019) method was used to get a vector representation of the word, And words with stop words less than three times are deleted. We use the Qiu et al. (2016) tool to segment the data set for Chinese news text. We randomly selected 5000, 15 000, 25 000, 35 000, and 45 000 data in large-scale training samples, and divided them into 80% as training samples and 20% as test samples for model analysis. The origin of the two data sets will be introduced. The data format mainly includes the text of the news and the tags of the text.

- Millions of real data are provided by the Sohu sponsor, which is currently the largest data set to identify marketing intention detection. News data comes from 250,000 web pages, which is a challenging data set. The training data is split into annotated data and unlabeled data. The annotated data includes 50,000 news texts and 350,000 news photos. News, text fragments and pictures are indicated as labels with marketing intention detection. The size of unlabeled data is 200,000 news items and 1 million news images. The scale of the test data set is 10,000 news and 70,000 new maps. The address of the dataset can be downloaded from here (https://biendata.com/competition/sohu2018/data/).
- AG is a collection of more than 1 million news articles. News articles have been gathered from more than 2000 news sources by ComeToMyHead in more than 1 year of activity. ComeToMyHead is an academic news search engine which has been running since July, 2004. Data sets are provided by academic associations for data mining and information retrieval. The data set was derived from the original corpus. The news was classified into marketing and non-marketing categories, and each category contained 30,000 training samples and 1900 test samples. The address of the datasets can be downloaded from here (https://github.com/mhjabreel/CharCNN/tree/master/data).

Table 1
Experimental parameters.

Parameter name	Catboost	XGBoost	Lightgbm	DNN	CNN	LSTM	CLSTM-TMN
topic_num	–	–	–	–	–	–	24
hidden_num	–	–	–	30	20	20	32
topic_emb_dim	–	–	–	60	65	60	60
max_seq_len	–	–	–	30	20	20	50
batch_size	–	–	–	80	50	40	–
max_depth	6	5	−1	–	–	–	–
reg_lambda	3	4	0	–	–	–	–
n_estimators	130	100	60	–	–	–	–

Table 2
Environment configuration.

Hardware and software 1	Configure
CPU	i5-8500 CPU @3.00 GHz X64
RAM	8.00 GB
Operating System	Windows 10
Environment	Anaconda 3.7 Keras 2.3.1

Table 3
Result matrix.

	Predict A	Predict B
Class A	True Positive(e)	False Negative(n)
Class B	False Positive(p)	True Negative(t)

4.2. Model settings

After continuous training of the model, the following parameters were finally determined. In the process of model training, the parameters are optimized, and the optimal parameters are obtained while achieving the highest accuracy. The model tends to be stable and has good results based on these parameters. The dimension of Embeddings in the experiment is 200, topic_num represents the number of subjects, hidden_num represents the size of the hidden layer, max_seq_len represents the length of the clipped text, n_estimators is represented as the number of submodels, topic_emb_dim represents the size of the subject memory, max_depth indicates the depth of the tree, reg_lambda is expressed as the weight of the regular item, specific experimental parameters are shown in Table 1.

4.3. Construction of environment

The experiment is based on the Python language. The Keras framework is used to build the model. The specific parameters of the experimental environment are as Table 2:

5. Result and discussion

Topic extraction methods and classifier effects are analyzed in detail in this chapter. In order to compare the effects of the topic feature extraction method and the text classification method in our experiment, the F1, ACC, and LOSS indicators were used to analyze the experiment. F1 is a method of measuring the reliability of a model, which is the weighted and adjusted average of the accuracy rate (P) and the recall rate (R). Accuracy is the ratio of the number of correctly categorized samples to the total number of samples, It is one of the commonly used indicators in the hybrid model. LOSS is used to evaluate the inconsistency between the predicted value of the model and the true value. The smaller its value, the better the robustness of the model. a represents the sample, b represents the category, and the calculation formula is as follows 3:

$$R = \frac{e}{e+n} \tag{11}$$

$$P = \frac{e}{e+p} \tag{12}$$

Table 4
Example topics generated by LSI, PLSA, NMF, NTM and RNTM.

LSI	PLSA	NMF	NTM	RNTM
daily	fresh	sauce	sauce	dish
fresh	urchins	urchins	fresh	soup base
dishes	enjoyment	feel	craft	sauce
children	dishes	material	industry	chop in
colorful	pancake	fans	fresh	fresh and tender
holiday	green	cuisine	material	slack season
two	rice	knead	cost	low cost
pancake	cost	cost	business	serve and enjoy

Table 5
The topic extraction effect is based on CLSTM-TMN. The higher the F1 value, the better the model. Popular methods for topic extraction include LSI (Kicsi et al., 2018; Song et al., 2019) method applicable to small text data, PLSA method (Zhu et al., 2018; Xie et al., 2019) based on co-occurrence data analysis and NMF (Miljkovic et al., 2019; Liu et al., 2019) method based on matrix decomposition.

Data sets	Model	F1	LOSS	ACC
SouHu	LSI	0.657	0.0369	0.718
AG	LSI	0.557	0.0344	0.674
SouHu	PLSA	0.622	0.0453	0.453
AG	PLSA	0.667	0.0235	0.544
SOHU	NMF	0.665	0.0354	0.693
AG	NMF	0.681	0.0276	0.715
SOHU	NTM	0.675	0.0304	0.733
AG	NTM	0.671	0.0266	0.725
SOHU	RNTM	**0.715**	**0.0256**	**0.775**
AG	RNTM	**0.733**	**0.0255**	**0.765**

$$F1 = \frac{2PR}{P+R} \tag{13}$$

$$ACC = \frac{(e+t)}{e+n+p+t} \tag{14}$$

$$L(b, P(b|a)) = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i \log p_i + \left(1-y_i\right)\log\left(1-p_i\right)\right) \tag{15}$$

5.1. Comparison of topic extraction methods

In order to evaluate the performance of RNTM, we adopted different models based on the same newsgroup for qualitative analysis, to show that our model can well accomplish the identification of marketing intention. The closer the keywords are to our marketing representative model, the more reliable it is. The data set was used by us to analyze all the topics generated by LSI, PLSA, NMF, NTM, and RNTM. Analyze our topic extraction method according to the generated topic words and classification effects.

- Through the analysis in Table 4, we find that the extracted topic words in LSI are composed of Numbers or short characters. For example, LSI mistakenly captures the word "two". PLSA performed better than LSI, and most of the typical topic words were meaningful, but we found that there was always some overlap between the two topics. We also found that some of the typical topics generated by NTM, such as "industry" and "business", they are better suited for marketing topics. Instead, PLSA and NMF represent topics that distort the true meaning of the topic under the assumption of the word bag. We find that NTM contains more unigram representing the topic, while RNTM chooses more bigram in its topic. For example, a topic generated by RNTM contains a topic feature consisting of two words, while an NTM topic contains individual words. This shows that RNTM helps generate marketing topics.
- To verify the performance of RNTM, F1, ACC, and LOSS measures were used to score each model. In this section, we will focus on analyzing the performance of RNTM on topic extraction tasks. Generally, the better the performance, the more reasonable the topic representation generated by the model. As shown in Table 5,

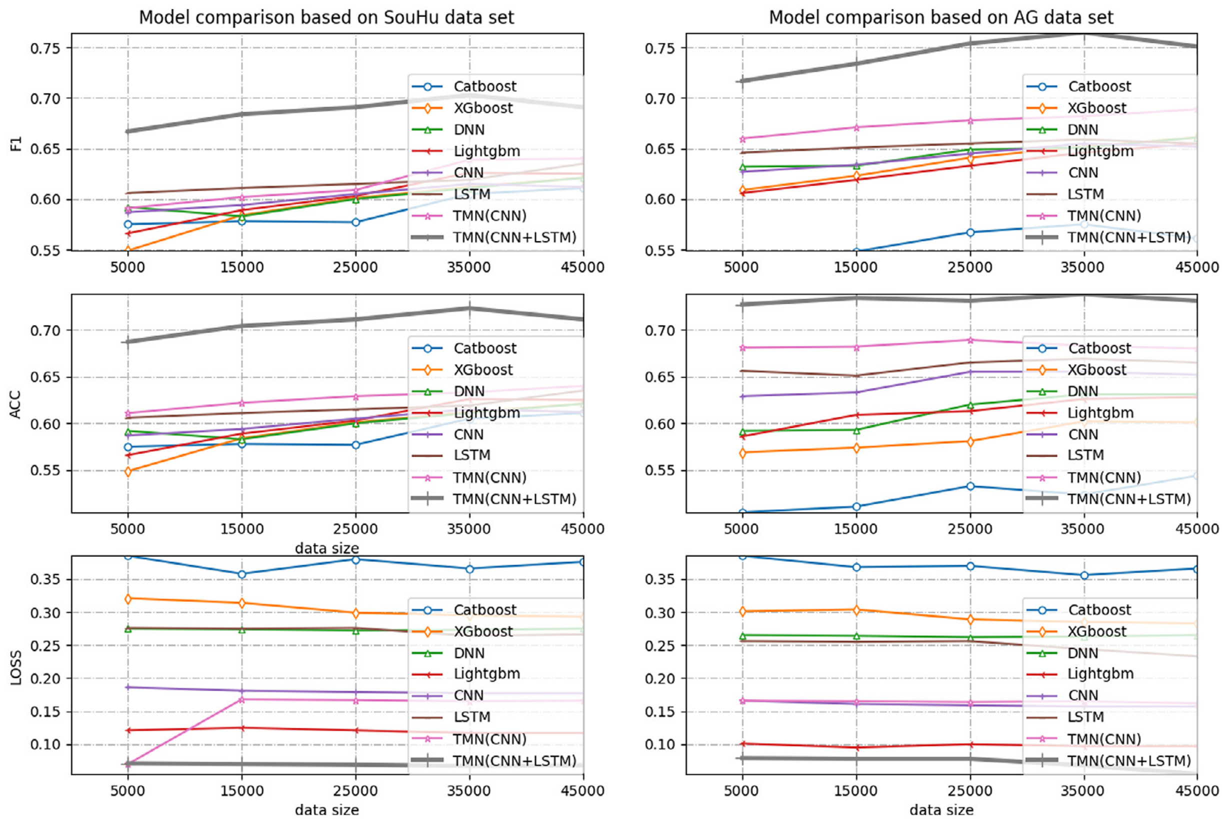

Fig. 5. The comparison result of the classifier is based on two data sets. The TMN model is respectively compared with the CNN (Le et al., 2018) method that captures local correlation, the Xgboost method (Mohammad, 2018) based on learning, the Lightgbm method (Chatterjee et al., 2019) based on histogram algorithm, the neural network Dnn (Paul et al., 2018) with multiple hidden layers, the LSTM model (Amplayo et al., 2018) based on timing sequence, and the Catboost (Sun et al., 2019) based on symmetric tree.

The parameters of each model are shown in Table 1. Experiments are compared and analyzed under data sets of different sizes. In general, the performance of the model improves with the increase of the number of topics. When the number of topics is set around 24, the performance of the model reaches the best. We can see that RNTM is more stable than PLSA and NMF, and we know that the only difference between NTM and RNTM is that RNTM has a modified linear unit. With the addition of the unit layer, RNTM can further adjust the topic distribution generated by NTM to improve classification performance. By comparing the three indicators, TMN can produce a coherent topic compared to other models. Compared to LSI and NMF, NTM produces better results through the results, which means that neural networks can successfully induce the generation of topics. The disadvantage of NTM is that it is prone to gradient disappearance in the process of back propagation, Topics are not representative of those extracted by NTM. Compared with other topic models, the advantage of RNTM is that it can complete the training of the deep-level model, and it is not easy for the problem of gradient disappearance to occur during the back propagation, so as to propose topics of common significance.TMN can help extract effective topics.

5.2. Classification comparison

- As shown in Fig. 5. When deep learning methods (CNN, LSTM, and DNN) are compared with traditional methods (Catboost, XGboost, Lightgbm), the results show that the neural network approach is superior to the traditional approach based on all data sets. It is proved that the method based on neural network can construct the semantic representation of text efficiently. Compared with the traditional machine learning method, deep learning can not only alleviate the data sparsity problem, but also obtain meaningful contextual feature information. The CLSTM model has advantages in indicators over a single CNN or LSTM.
- Obviously, Comparing model TMN with CLSTM-TMN, the CLSTM-TMN model can capture more text features than typical classifiers in short texts. Therefore, the performance of model CLSTM-TMN in the experiment is better than other models. The F1 of model CLSTM-TMN based on two test sets are respectively 71% and 73% higher than that of other models, which indicates that it is efficient for LSTM to save context information and historical information in short text. Because CNN ignores context dependencies in text during the process of convolution filtering of word vectors in text. LSTM can improve the dependency relationship between context parts of speech, and it can save the text information in the model, but LSTM's ability to extract text features from full text is insufficient. The LSTM layer, which is added between the CNN's pooling layer and the fully connected layer, enables the semantic information of the context to be fully extracted, and the effect is improved. The model proposed in this paper can well identify news for Marketing intention detection, which can be seen from the experimental results. Compared with the data volume of different scales, the CLSTM-based TMN model can extract the topic model more effectively than other models in text classification. By extracting time features from input sequences, it plays an important part in the text representation of learning topics. The performance of our CLSTM-TMN model is superior to the TMN model.

6. Conclusions and future work

In this work, an improved topic memory network model was proposed to identify marketing intent. Not only time information can be mined, but also spatial information can be mined. The F1 value is equal to 76%. The validity of this method is verified in this paper.

This article is based on CLSTM-TMN to implement marketing intent detection. More research ideas should be included in the future. The content of future research should be from the following two aspects: First, most of the data is now short text recognition, which can be analyzed against long texts to verify the experimental results Secondly, the training time of deep learning is long, so the model structure and parameters can be optimized to speed up the training time of the model without reducing the accuracy of the model.

CRediT authorship contribution statement

Yufeng Wang: Conceptualization, Methodology, Software, Writing - original draft, Writing - review & editing. **Kun Ma:** Methodology, Writing - original draft, Writing - review & editing, Supervision. **Laura Garcia-Hernandez:** Investigation, Validation. **Jing Chen:** Writing - original draft. **Zhihao Hou:** Conceptualization, Methodology, Software. **Ke Ji:** Investigation. **Zhenxiang Chen:** Investigation. **Ajith Abraham:** Supervision, Conceptualization.

References

Amplayo, R.K., Kim, J., Sung, S., Hwang, S.-w., 2018. Cold-start aware user and product attention for sentiment classification. arXiv preprint arXiv:1806.05507.

Athalye, A., Carlini, N., Wagner, D., 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. arXiv preprint arXiv:1802.00420.

Cao, Z., Li, S., Liu, Y., Li, W., Ji, H., 2015. A novel neural topic model and its supervised extension. In: Twenty-Ninth AAAI Conference on Artificial Intelligence.

Chatterjee, A., Narahari, K.N., Joshi, M., Agrawal, P., 2019. Semeval-2019 task 3: Emocontext contextual emotion detection in text. In: Proceedings of the 13th International Workshop on Semantic Evaluation. pp. 39–48.

Chen, S., 2018. K-nearest neighbor algorithm optimization in text categorization. In: IOP Conference Series: Earth and Environmental Science, Vol. 108. IOP Publishing, p. 052074.

Fang, D., Yang, H., Gao, B., Li, X., 2018. Discovering research topics from library electronic references using latent Dirichlet allocation. Libr. Hi Tech. 36 (3), 400–410.

Fang, Y., Zhang, C., Huang, C., Liu, L., Yang, Y., 2019. Phishing email detection using improved rcnn model with multilevel vectors and attention mechanism. IEEE Access 7, 56329–56340.

Geeganage, K., Tharanga, D., 2018. Concept embedded topic modeling technique. In: Companion Proceedings of the the Web Conference 2018. International World Wide Web Conferences Steering Committee, pp. 831–835.

Goudjil, M., Koudil, M., Bedda, M., Ghoggali, N., 2018. A novel active learning method using svm for text classification. Int. J. Autom. Comput. 15 (3), 290–298.

Hoffman, M., Bach, F.R., Blei, D.M., 2010. Online learning for latent dirichlet allocation. In: Advances in Neural Information Processing Systems. pp. 856–864.

Hofmann, T., 1999. Probabilistic latent semantic analysis. In: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann Publishers Inc., pp. 289–296.

Hofmann, T., 2017. Probabilistic latent semantic indexing. In: ACM SIGIR Forum, Vol. 51. ACM, pp. 211–218.

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., Mikolov, T., 2016. Fasttext. zip: Compressing text classification models. arXiv preprint arXiv:1612.03651.

Kang, M., Ahn, J., Lee, K., 2018. Opinion mining using ensemble text hidden markov models for text classification. Expert Syst. Appl. 94, 218–227.

Kicsi, A., Vidács, L., Csuvik, V., Horváth, F., Beszédes, A., Kocsis, F., 2018. Supporting product line adoption by combining syntactic and textual feature extraction. In: International Conference on Software Reuse. Springer, pp. 148–163.

Kim, Y., 2014. Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

Kipf, T.N., Welling, M., 2016. Variational graph auto-encoders. arXiv preprint arXiv:1611.07308.

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., Brown, D., 2019. Text classification algorithms: A survey. Information 10 (4), 150.

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

Landauer, T.K., McNamara, D.S., Dennis, S., Kintsch, W., 2013. Handbook of Latent Semantic Analysis. Psychology Press.

Le, H.T., Cerisara, C., Denis, A., 2018. Do convolutional networks need to be deep for text classification?. In: Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence.

Li, H., Huang, H., Cao, X., Qian, J., 2018. Falcon: A novel chinese short text classification method. J. Comput. Commun. 6, 216–226.

Liu, G., Guo, J., 2019. Bidirectional lstm with attention mechanism and convolutional layer for text classification. Neurocomputing 337, 325–338.

Liu, H., Li, Q., Yao, R., Zeng, D.D., 2019. Analyzing topics of juul discussions on social media using a semantics-assisted nmf model. In: 2019 IEEE International Conference on Intelligence and Security Informatics (ISI). IEEE, pp. 212–214.

Ma, K., Yu, Z., Ji, K., Yang, B., 2019. Stream-based live public opinion monitoring approach with adaptive probabilistic topic model. Soft Comput. 23 (16), 7451–7470.

Madera, Q., Castillo, O., García-Valdez, M., Mancilla, A., 2017. A method based on interactive evolutionary computation and fuzzy logic for increasing the effectiveness of advertising campaigns. Inform. Sci. 414, 175–186.

Madera, Q., Garcia, M., Castillo, O., 2016. Fuzzy logic for improving interactive evolutionary computation techniques for ad text optimization. In: Novel Developments in Uncertainty Representation and Processing. Springer, pp. 291–300.

Miao, Y., Grefenstette, E., Blunsom, P., 2017. Discovering discrete latent topics with neural variational inference. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, pp. 2410–2419.

Miljkovic, D., Kralj, J., Stepišnik, U., Pollak, S., 2019. Communities of related terms in a karst terminology co-occurrence network.

Mohammad, F., 2018. Is preprocessing of text really worth your time for online comment classification?. arXiv preprint arXiv:1806.02908.

Paul, A., Jha, D., Al-Bahrani, R., Liao, W.-k., Choudhary, A., Agrawal, A., 2018. Chemixnet: Mixed dnn architectures for predicting chemical properties using multiple molecular representations. arXiv preprint arXiv:1811.08283.

Qiu, X., Qian, P., Shi, Z., 2016. Overview of the nlpcc-iccpol 2016 shared task: Chinese word segmentation for micro-blog texts. In: Natural Language Understanding and Intelligent Applications. Springer, pp. 901–906.

Rezaeinia, S.M., Rahmani, R., Ghodsi, A., Veisi, H., 2019. Sentiment analysis based on improved pre-trained word embeddings. Expert Syst. Appl. 117, 139–147.

Sebastiani, F., 2002. Machine learning in automated text categorization. ACM Comput. Surv. (CSUR) 34 (1), 1–47.

Selvam, M.K., 2018. Classification of Malicious Web Code Using Deep Learning (Ph.D. thesis). National College of Ireland, Dublin.

Shen, J., Xiang, J., Zha, D., Fu, T., Duan, D., 2019. Intention understanding model inspired by cbc loops. In: 2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design (CSCWD). IEEE, pp. 105–110.

Song, Y., Chen, Y., Lang, B., Liu, H., Chen, S., 2019. Topic model based android malware detection. In: International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage. Springer, pp. 384–396.

Sun, X., Ma, X., Ni, Z., Bian, L., 2018. A new lstm network model combining textcnn. In: International Conference on Neural Information Processing. Springer, pp. 416–424.

Sun, B., Yang, L., Zhang, W., Lin, M., Dong, P., Young, C., Dong, J., 2019. Supertml: Two-dimensional word embedding for the precognition on structured tabular data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops.

Wang, R., Li, Z., Cao, J., Chen, T., Wang, L., 2019a. Convolutional recurrent neural networks for text classification. In: 2019 International Joint Conference on Neural Networks (IJCNN). IEEE, pp. 1–6.

Wang, Y., Liu, S., Li, S., Duan, J., Hou, Z., Yu, J., Ma, K., 2019b. Stacking-based ensemble learning of self-media data for marketing intention detection. Future Internet 11 (7), 155.

Wang, Y., Xu, W., 2018. Leveraging deep learning with lda-based text analytics to detect automobile insurance fraud. Decis. Support Syst. 105, 87–95.

Wu, P., Lu, Z., Zhou, Q., Lei, Z., Li, X., Qiu, M., Hung, P.C., 2019. Bigdata logs analysis based on seq2seq networks for cognitive internet of things. Future Gener. Comput. Syst. 90, 477–488.

Xi, X., Zhang, T., Du, D., Zhao, G., Gao, Q., Zhao, W., Zhang, S., 2018. Method and system for detecting anomalous user behaviors: An ensemble approach.. In: SEKE. 263–262.

Xie, X., Ge, S., Hu, F., Xie, M., Jiang, N., 2019. An improved algorithm for sentiment analysis based on maximum entropy. Soft Comput. 23 (2), 599–611.

Yao, L., Mao, C., Luo, Y., 2019a. Clinical text classification with rule-based features and knowledge-guided convolutional neural networks. BMC Med. Inform. Decis. Mak. 19 (3), 71.

Yao, L., Mao, C., Luo, Y., 2019b. Graph convolutional networks for text classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, pp. 7370–7377.

Young, T., Hazarika, D., Poria, S., Cambria, E., 2018. Recent trends in deep learning based natural language processing. IEEE Comput. Intell. Mag. 13 (3), 55–75.

Yuan, J., Gao, F., Ho, Q., Dai, W., Wei, J., Zheng, X., Xing, E.P., Liu, T.-Y., Ma, W.-Y., 2015. Lightlda: Big topic models on modest computer clusters. In: Proceedings of the 24th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, pp. 1351–1361.

Zaremba, W., Sutskever, I., Vinyals, O., 2014. Recurrent neural network regularization. arXiv preprint arXiv:1409.2329.

Zeng, J., Li, J., Song, Y., Gao, C., Lyu, M.R., King, I., 2018. Topic memory networks for short text classification. arXiv preprint arXiv:1809.03664.

Zhang, T., Huang, M., Zhao, L., 2018a. Learning structured representation for text classification via reinforcement learning. In: Thirty-Second AAAI Conference on Artificial Intelligence.

Zhang, R., Lee, H., Radev, D., 2016. Dependency sensitive convolutional neural networks for modeling sentences and documents. arXiv preprint arXiv:1611.02361.

Zhang, J., Li, Y., Tian, J., Li, T., 2018b. Lstm-cnn hybrid model for text classification. In: 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). IEEE, pp. 1675–1680.

Zhang, L., Nie, Y., Duan, S., 2019. Chinese spam filtering based on stacked denoising autoencoders. In: Journal of Physics: Conference Series, Vol. 1237. IOP Publishing, p. 032012.

Zhang, X.-L., Wang, D., 2016. A deep ensemble learning method for monaural speech separation. IEEE/ACM Trans. Audio Speech Lang. Process. (TASLP) 24 (5), 967–977.

Zhang, L., Wang, S., Liu, B., 2018c. Deep learning for sentiment analysis: A survey. Wiley Interdiscip. Rev.: Data Min. Knowl. Discov. 8 (4), e1253.

Zhao, S., Li, X., Liu, W., 2008. Bagging text classification algorithm based on classifier performance evaluation. Comput. Eng. 34 (1), 61–63.

Zhou, C., Sun, C., Liu, Z., Lau, F., 2015. A c-lstm neural network for text classification. arXiv preprint arXiv:1511.08630.

Zhu, Q., Zhong, Y., Wu, S., Zhang, L., Li, D., 2018. Scene classification based on the sparse homogeneous–heterogeneous topic feature model. IEEE Trans. Geosci. Remote Sens. 56 (5), 2689–2703.