Engineering Applications of Artificial Intelligence 98 (2021) 104118

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Attention-based learning of self-media data for marketing intention detection☆

Zhihao Hou [a], Kun Ma [a,b,*], Yufeng Wang [a], Jia Yu [a], Ke Ji [a,b], Zhenxiang Chen [a,b], Ajith Abraham [c]

[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] Machine Intelligence Research Labs, Scientific Network for Innovation and Research Excellence, Auburn, USA

ARTICLE INFO

Keywords:
Marketing intention detection
Attention model
Convolutional neural network
Feature extraction

ABSTRACT

In the context of natural language processing, accuracy of intention detection is the basis for subsequent research on human-machine speech interaction. However, the problem of ambiguity in word vectors reduces the accuracy of intent detection. Meantime, there is a disconnection between local features and global features as well, resulting in text feature extraction that cannot fully reflect semantic information. These issues are all barriers of intention detection. Therefore, this paper proposes an attention-based convolutional neural network for self-media data learning (called A-CNN) for marketing intention. We cascade the traditional CNN with the self-attention model in the Attention networks to form a new network structure called A-CNN, and put forward a fast feature extraction method based on skip-gram-based learning called FSLText, to represent the high-dimension word vectors in the A-CNN. On the premise of maintaining the advantages of the CNN, A-CNN can not only solve the problem of local and global features disconnection caused by the CNN pooling layer, but also avoid the increase of algorithm complexity. The Self-Attention mechanism in the Attention model can effectively optimize the weight of local features of the information in global features, and retain local features that are more useful for intention detection. A fast feature extraction method which is based on Skip-gram can retain the semantic and word order information of the text. The method is beneficial to the marketing intention detection. According to the experiment, our A-CNN, compared with traditional machine learning methods, can improve 12.32% accuracy. Contrast to the dual-channel CNN, the accuracy rate is improved by 9.68%, and compared with the ATT-CNN, it is improved by 9.97%. On the F1 score, the A-CNN can improve the F1 score by about 9.37% in comparison with the traditional machine learning methods, the accuracy rate is increased by 9.68% compared with the dual-channel CNN, and 9.68% in contrast with ATT-CNN. It illustrates that our A-CNN can effectively address semantic and feature selection for marketing intention detection.

1. Introduction

1.1. Background

In recent years, online media have changed the way people live and work with the rapid development of the Internet. New online media have the advantages of wide coverage, large popularity and rapid spread (Liang and Zhang, 2019). As a result, their influence on social public opinion is growing. Nevertheless, the rapid development of online media has brought about many social issues as well as its convenience due to inefficient supervision. At the meantime, lack of information transparency and the low cost of the threshold for spreading malicious news, have led to the rapid growth of online malicious marketing news (Wang et al., 2020). Malicious marketing news contains a lot of fake advertisements and spam marketing content, which pollutes the online news environment and reduces reading experience of netizens (Al-Rawi, 2019). In view of this background, this paper has proposed an attention-based learning of self-media data (called A-CNN) for the detection of marketing intention flooded in cyberspace. This method can optimize the ratio of local features to global features to address the issue of polysemy of words. Experiments have proven that this method provides an efficient solution to purify

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

* Corresponding author at: School of Information Science and Engineering, University of Jinan, Jinan 250022, China.
E-mail addresses: houzhihao@mail.ujn.edu.cn (Z. Hou), ise_mak@ujn.edu.cn (K. Ma), 464017311@qq.com (Y. Wang), 929652386@qq.com (J. Yu), 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.2020.104118
Received 15 June 2020; Received in revised form 30 September 2020; Accepted 26 November 2020
Available online 2 December 2020
0952-1976/

public opinion information, identify marketing news, and improve the reading experience of users.

1.2. Challenges

In current research of news text recognition, the widely used traditional Convolutional Neural Network (abbreviated as CNN) (Kim, 2014) has the disadvantage that the pooling layer will lose the relevance of the local and global features (Sabour et al., 2017). In current research of image processing, the Bag-of-words (abbreviated as BOW) model is used to replace the fully connected layer to be embedded in the CNN (Xue et al., 2016), which makes the transformation have stronger invariance and achieve better results. But in current research of natural language processing (abbreviated as NLP), the usage of the BOW model instead of the fully connected layer ignores the effect of word order on semantics. In 2019, graph convolutional neural networks (short for GCN) was proposed to classify the features in natural language processing (Yao et al., 2019). GCN is more robust in the case of reducing training data in text classification, but it also ignores the impact of word order on classification due to the usage of one-hot encoding as input. Therefore, it is necessary to optimize feature extraction and selection and find efficient classification methods for marketing intention detection.

Currently, traditional CNN has the disadvantage of translation invariance (Sabour et al., 2017). The mainstream feature extraction methods do not take into the account of the degree of semantic similarity between words and the high sparseness of vector space. It also affects the overall effect of the intention recognition and the training effect of the classification.

1.3. Contributions

The contributions of our work are attention-based convolutional neural network (abbreviated as A-CNN) and fast Skip-gram-based learning of word representations. The inspiration of this method comes from the improvement of current CNN and attention-based CNN. We employ the average pooling in the pooling layer together with the weighted summation in the attention mechanism so as to form the new Attention layer. Compared with the maximum pooling and the average pooling in the traditional pooling layer, it can better capture important information from the data.

- **Attention-based convolutional neural network (abbreviated as A-CNN)**. We have cascaded the self-attention mechanism of the Attention networks (Vaswani et al., 2017) in CNN to form a new A-CNN structure. In traditional CNN, the data retained by the maximum pooling and average pooling in the pooling layer may not be useful for intent recognition. Therefore, we have added the Attention mechanism to the pooling layer. By calculating the attention distribution of the data, the input information is weighted and averaged, and then sent to the fully connected layer together with the ordinary-averaged information. This is more effective than the simple maximum pooling and average pooling in terms of retaining useful information for classification. Self-attention mechanism can capture local and global features more flexibly. As a result, the ratio of local features to global features of the information can be significantly optimized in A-CNN for intent detection. Compared with another ATT-CNN (Zhao and Wu, 2016) that puts the attention model before the CNN convolutional layer, our A-CNN can not only solve the syntax and semantic problems which depend on feature extraction methods, but also solve the problem of feature loss in the pooling layer by cascading the self-attention mechanism in it.
- **Fast Skip-gram-based learning of word representations (abbreviated as FSLText)**. A feature extraction method based on skip-gram is proposed to represent high-dimension word vectors in our A-CNN, which is based on the Skip-gram model of word2vec (Zhang et al., 2018). For each word, it is divided into n-gram characters to represent. It not only takes into account the word order, but also solves the problem of out of vocabulary words. Therefore, we can still construct their word vectors for words outside the training vocabulary table. Considering local word order, our FSLText word vector allows the A-CNN to have a better recognition effect in the face of newly derived words than using ordinary word vectors.

1.4. Organization

The rest of this article is organized as follows. Section 2 discusses current feature extraction, model selection, and pattern cascading methods. Section 3 discusses the definition of the marketing intention problem and the framework and experimental evaluation criteria for the solution. Section 4 introduces our attention-based learning of self-media data for marketing intention detection. Firstly, the overall architecture of a convolutional neural network based on the Self-Attention model is introduced. Secondly, fast skip-gram-based learning of word representation is explained in detail. Experiments in Section 5 show that our method is effective. The last section outlines brief conclusions and future research directions.

2. Related work

Marketing intent detection belongs to text classification (Kowsari et al., 2019). In this section, we sort out the feature extraction methods of text classification in natural language processing, and give a detailed explanation of the selection of classification models as well as the commonly used model fusion methods. We point out the advantages and disadvantages of current methods, make some analysis and emphasize the differences between our method.

2.1. Feature extraction

In traditional natural language processing, the first step is feature extraction. One-hot coding is widely applied in conventional machine learning, which is easy to use and understand (Alpaydin, 2020). However, due to its high dimension and coding method, it is too sparse to cause disastrous dimension problems easily. In the traditional Vector Space Model, the text is represented as a multi-dimension vector composed of the frequency (or probability) of feature words, and then the similarity between the vectors is calculated (Salton et al., 1975). Compared with one-hot coding, the similarity between texts, such as TF-IDF (Ramos et al., 2003), is taken into account. But it assumes that each term is independent of each other, thus some context word order in the text might be lost. This problem led to Latent semantic analysis (LSA) model (Landauer et al., 1998). LSA, based on singular value decomposition (SVD), supposes that words have close meanings will occur in similar pieces of text. The model can reduce the number of rows while preserving the similar structure among columns, and find hidden semantic dimensions from the text. But the LSA assumes that the words in the text are Gaussian, which may not be suitable for all problems, and SVD requires a lot of computing power when new data or updates occur.

For word vector extraction in neural networks, the first language model is NNLM (Neural Network Language Model) (Bengio et al., 2003). It is the first time that the concept of word vectors has been proposed, that is, text is expressed in dense, low-dimension and continuous vectors. Nevertheless, it requires more training parameters and has a large computational overhead. In 2013 Word2vec word vector method appeared, namely CBOW and Skip-gram model. The CBOW

model predicts the central words according to the words surrounding the central word W (t), and the Skip-gram model predicts the surrounding words according to the central words W (t). The following are some of the problems. First of all, Word2vec does not consider the order of words. What is more, word2vec assumes that words and words are independent of each other. But in most cases, they affect each other indeed. Besides, the features it obtains are discrete and sparse. Subsequently, the Glove (Pennington et al., 2014) algorithm was proposed in 2014. It essentially reduces the dimension of the co-occurrence matrix. This algorithm constructs a co-occurrence matrix for each word and calculates the frequency of each word in each context. In practical applications, Glove distinguishes between the target word vector and the context vector, and finally sums the two sets to obtain the final word vector. It takes the word order of the text into account . But at the same time, Glove's loss function easily leads to adding a large constant vector to the word vector. Thus all the word vectors will be very close to each other, losing its original meaning.

In 2018, the dynamic word vectors ELMo (Peters et al., 2018) and Bidirectional Encoder Representation from Transformers (abbreviated as BERT) (Devlin et al., 2018) were successively proposed. ELMo uses a typical two-stage process (Peters et al., 2018). The first stage is to use language models for pre-training. The second stage is to, when doing downstream tasks, extract the Word Embeddings at each layer of the networks which is from the pre-trained network, and then add them to the downstream tasks as new features. BERT uses Transformer's Encode to train a bidirectional language model, followed by a specific classifier (Devlin et al., 2018). The common feature of the two is that they both use transfer learning methods, pre-trained language models and fine-tuning according to specific uses. With only a small amount of labeled data, the accuracy of text classification can be equivalent to thousands of times the amount of labeled data training level. Nevertheless, in practical applications, to choose a proper pre-trained model is a problem. And transfer learning is difficult to determine under what circumstances should pre-training stop. It is hard to determine the level and number of parameters of the pre-trained model as well.

In this paper, in order to simplify the difficulty of feature extraction and increase its speed, we put forward a fast Skip-gram-based word representation learning method to cope with the two problems. The method is based on the skip-syntax model in Word2vec. We use 300-dimension FSLText word vectors to describe each word in the text, and N-gram as words to train the word vectors. Derived word vector training increases the number of word vectors to a certain extent, making it perform better when facing new word texts.

2.2. Model selection

In terms of intention recognition, the machine learning methods which are often used by high-score teams in Kaggle competitions are mainly LightGBM (Ke et al., 2017), XGBoost (Chen and Guestrin, 2016), Random Forest (Couronné et al., 2018), Gradient Boosting (Xi et al., 2018) and so on. LightGBM is a boosting algorithm based on a tree model. The Histogram algorithm it uses reduces the training time. The leaf-wise growth strategy reduces more errors, achieves higher accuracy and effectively lower overfitting (Ke et al., 2017). XGBoost is currently the fastest and best open source boosting tree toolkit. It performs a second-order Taylor expansion of the loss function. It can also customize the loss function as well as increase accuracy. The column sampling and the processing of missing values make it effectively avoid overfitting and reduce the calculation times (Chen and Guestrin, 2016). Gradient Boosting is a widely used machine learning method that can flexibly process various types of data. It has no requirement for numerical features normalization. It is not sensitive to missing values, and can also learn different loss functions (Xi et al., 2018). Random Forest can process higher-dimension data and has a strong generalization ability. For unbalanced data sets, Random Forest can balance errors and still keep its accuracy when facing feature loss (Couronné et al., 2018).

The main models used in deep learning are Convolutional Neural Networks (abbreviated as CNN), Recurrent Neural Networks (abbreviated as RNN) (Salehinejad et al., 2017), Long Short-Term Memory (abbreviated as LSTM) (Hochreiter and Schmidhuber, 1997), Generative Adversarial Networks (abbreviated as GAN) (Li et al., 2017) and so on. Compared with RNN and LSTM, CNN has better parallelism. Deep learning often relies on large-scale samples (Kim, 2014), resulting in CNN currently having an advantage in computing power. While compared with GAN, GAN is not stable enough for processing discrete forms of data (Chen et al., 2018), such as text. Therefore, CNN still has high adaptability and wide application in NLP.

In recent years, CNN have been widely applied in NLP tasks. Their shared convolution kernels can process high-dimension data and can automatically select features (Kim, 2014). The Attention mechanism also has good experimental results in NLP tasks, due to its characteristics of parallel calculations, less model training time and hard to lose data (Vaswani et al., 2017). As a consequent, in order to simplify feature extraction process and avoid losing data, this paper chooses the CNN and the Attention model as the objects of the network structure cascade.

2.3. Model fusion

There are three common model fusion methods in machine learning: Bagging (Breiman, 1996), Boosting (Friedman, 2002) and Stacking (Wolpert, 1992). In the Bagging framework, each base model is trained based on different sub-training sets. The prediction values of all base models are synthesized to obtain the final prediction result (Breiman, 1996). The training process of Boosting is step-like. The training of the base models is sequential. Each base model will learn on the basis of the previous base model learning. Finally the prediction values of all base models are combined to produce the final prediction result (Friedman, 2002). Stacking is to train the base models with all the data firstly, and then each base model makes predictions for each training sample. The predicted value will be used as the feature value of the training sample. At last, a new training sample will be obtained. Thus, the sample will be trained to get the model and achieve the final result on the basis of the new one (Wolpert, 1992).

In neural networks, the network fusion methods are different in terms of the actual application requirements. For example, for the problem of video emotion recognition, the RNN-CNN method, which cascades RNN and CNN (Fan et al., 2016), is applied. RNN can use the appearance features extracted by Convolutional Neural Network (CNN) as the input of a single video frame, and after that, encode the motion. In the field of speech emotion recognition, the CNN-LSTM, cascading CNN and LSTM (Zhao et al., 2019), can learn features related to local and global emotions from the speech and spectrograms.

With regard to these traditional models, variants based on them have also been generated, such as BiLSTM, ABCNN, RCNN and so on. BiLSTM (Chen et al., 2017) is the abbreviation of Bi-directional Long Short-Term Memory. It combines forward LSTM and backward LSTM, and is often used to model contextual information in natural language processing tasks. The LSTM model can better capture longer-distance dependencies, while BiLSTM can better capture bidirectional semantic dependencies on its basis. ABCNN (Yin et al., 2016) is proposed to solve text similarity problems and matching problems. It puts Attention operations on different layers of traditional CNN networks. Compared with traditional CNN, it is not prone to semantic shift and can better retain the words useful for text matching. RCNN (Lai et al., 2015) can greatly reduce noise in comparison with traditional window-based neural networks, thereby capturing contextual information to the greatest extent. Moreover, this model can retain a wider range of word order when learning text representations, so as to avoid the impact of word order problems on text classification.

These are the fusion of the two networks to optimize the neural network, but there are many other ways to optimize the basic network.

On the one hand, it can be optimized for the basic network itself, such as the attention-based bidirectional long-short-term memory with convolution layer (AC-BiLSTM) (Liu and Guo, 2019). In AC-BiLSTM, the convolutional layer extracts the higher-level phrase representations from the word embedding vectors and BiLSTM is used to access both the preceding and succeeding context representations. Attention mechanism is employed to give different focus to the information outputted from the hidden layers of BiLSTM. Finally, the softmax classifier is used to classify the processed context information. AC-BiLSTM is able to capture both the local feature of phrases as well as global sentence semantics. You can also use the blur Fuzzy Gravitational Search Algorithm method (FGSA) (Poma et al., 2020b) to optimize the Convolutional Neural Network (CNN). It is inspired by the extension of the Gravitational Search Algorithm (GSA) using fuzzy logic. On the other hand, it can optimize the parameters of the basic network. For example, the dynamic parameter adjustment method in the Particle Swarm Optimization (PSO) (Sánchez et al., 2020), which is designed on the basis of the Particle Swarm Algorithm and the Modular Neural Network (MNN) of Fuzzy Logic (FL); and the optimization method of the filter size of the convolution neural network which uses the Fuzzy Gravitational Search Algorithm (FPGA) (Poma et al., 2020a).

In this paper, we address the problem of text intention recognition. We use the Attention mechanism to cascade into the conventional CNN, replace the pooling layer of the traditional convolutional neural network. We use the Attention mechanism to weight the local variables of the text to obtain the global ones. In addition, cascading the Attention mechanism into the CNN enables the CNN to maintain its basic network structure. Besides, for higher-dimension data, it is not easy to lose key information and can better handle the connection between local and global information. Meanwhile, the complexity of the Attention mechanism is small, and each step of the calculation does not depend on the previous calculation result. It can be processed in parallel with CNN. The cascade of the two will not increase the training time and complexity.

3. Problem definition

In this section, we present the overall implementation framework for the marketing intent detection regarding feature extraction, network structure, and network optimization. Besides, we specify the evaluation criteria used in our experiments.

3.1. Architecture of intention detection

The system framework of marketing intention detection is shown in Fig. 1. The main process consists of text pre-processing, feature extraction, classifier and output. After removing stop words and word segmentation, the next stage is to use FSLText to extract features. Finally, it is identified by A-CNN.

3.2. Metrics of intention detection

In the experiments, we use F1-score, accuracy (Acc) and Precision (P) as the measurement indicators. As is shown in Table 2, this is the corresponding classification result matrix.

- Precision: Calculates the ratio of all "correctly retrieved results (TP)" to all "actually retrieved (TP + FP)". The calculation formula is shown below.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

- F1: It is the weighted harmonic average of Precision (P) and Recall (R). F-measure is a suitable method to measure the reliability of the model. The calculation formulas are shown below.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

$$F1 = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

- Accuracy: For a given data set, the ratio of the number of correctly classified samples to the total number of samples. The formula is as follows.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

- AUC: AUC (Area Under Curve) is defined as the area enclosed by the coordinate axis under the ROC curve, and represents the probability that the positive example of the prediction is ranked before the negative example.
- The Akaike information criterion (AIC) is a fined estimator of in-sample fit to estimate the likelihood of a model to predict/estimate the future values for a given set of data. Friedman et al. (2001). In-sample prediction error is the expected error in predicting the resampled response to a training sample. Given a collection of models for the data, AIC estimates the quality of each model, relative to each of the other models. Thus, AIC provides a way for model selection. In the expression, K is the number of parameters and L is the likelihood function

$$AIC = 2k - 2\ln(L) = 2k + n\ln(RSS/n) \tag{5}$$

where RSS is the residual sum of squares.

4. Attention-based learning

In this section, we present our proposed A-CNN method to address marketing intention detection. The data preprocessing, feature extraction and model optimization are discussed in detail.

4.1. Pre-processing

The data cleaning process is divided into three steps. First, we use regular expressions to remove HTML tags and change the original web page information into text information. Then we use Jieba kit (Sun, 2012) to divide sentences into tokens. The processing process is shown in detail in Fig. 2. First, an efficient word graph scanning is based on the Trie-tree structure. Second, it generates a directed acyclic graph (DAG), which is composed of all possible word formation of Chinese characters in a sentence. Third, it uses dynamic programming to find the maximum probable path and the largest segmentation combination based on word frequency. For unregistered words, it uses the hidden Markov model (abbreviated as HMM model) to form words, which is based on the ability of Chinese characters, and the Viterbi algorithm to get the words sequence. Finally, it uses iterative search and deletes useless words in text information.

4.2. Feature extraction

In natural language processing, word vectors are used to reflect the features of text information. We propose FSLText to extract features from the pre-processed text. It is a kind of fast learning of word representation based on Skip-gram. For each word, FSLText divides it into n-gram characters to represent. This method not only considers the word order in the text but also performs well when the vocabulary is limited. Therefore, for words outside the training vocabulary set, we can still construct their word vectors. The FSLText model we proposed is used to train the corpus of divided words, and then achieve corresponding word vectors. Meanwhile, we serialize the text of the training set and the text of the validation set. And then put them into the word embedding layer of the neural network to form a semantic-based vector model. FSLText word vector representations can effectively address the issue of polysemy in text intention recognition. And it has a better classification performance when facing new words derived from existing words.

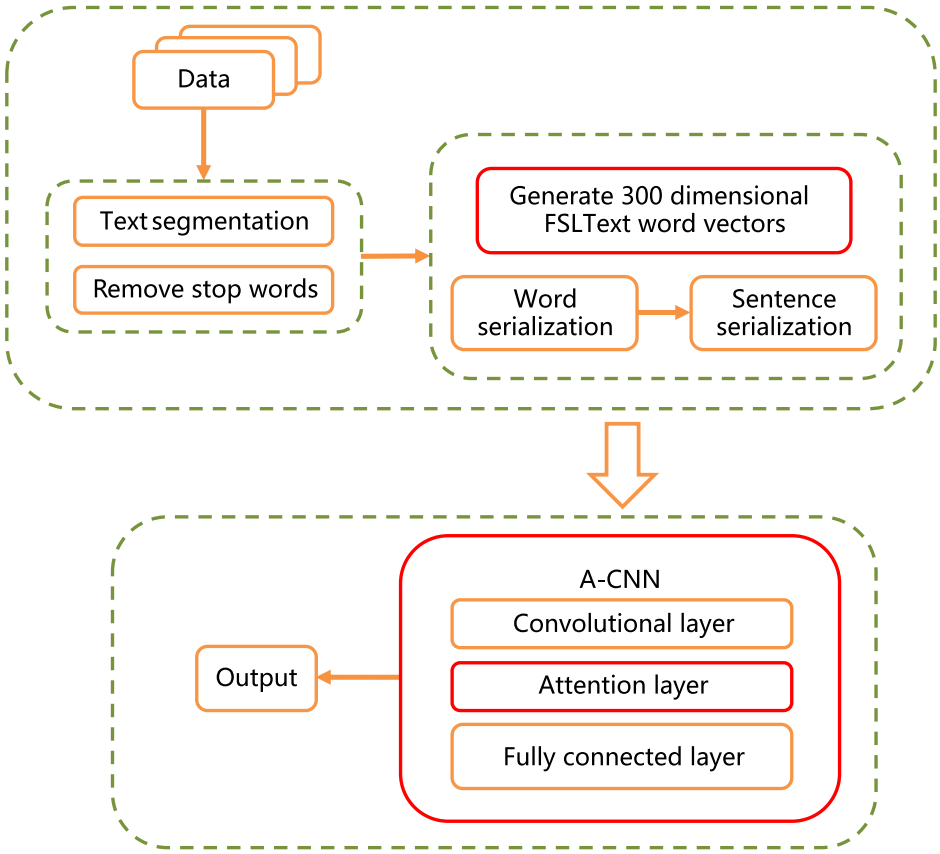

Fig. 1. General architecture of intention detection.

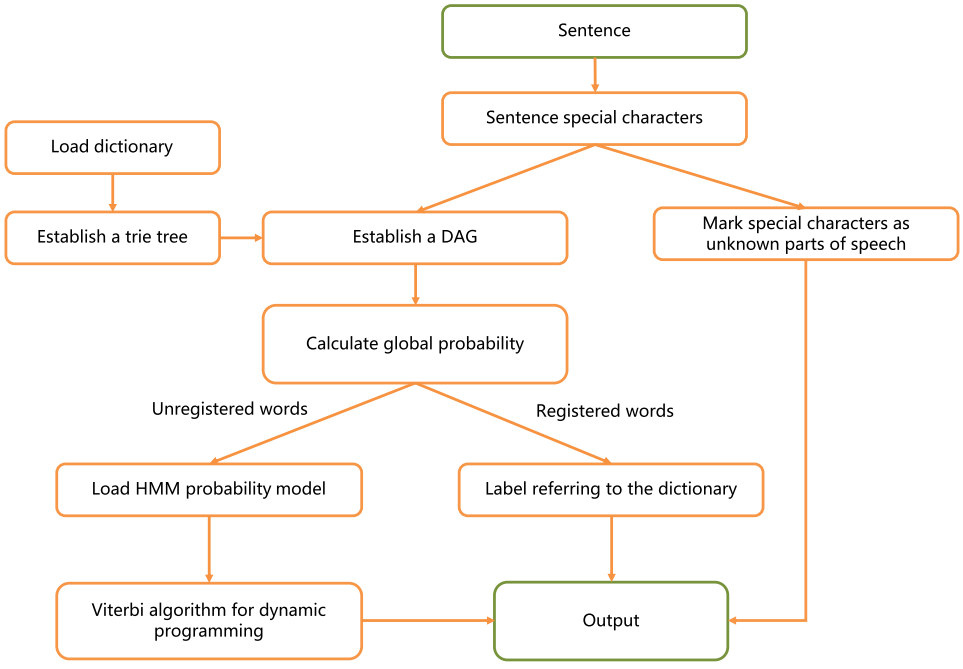

Fig. 2. Word segmentation process.

4.3. Model fusion

Because traditional CNN uses maximum pooling or average pooling to process the convolutional data, only the maximum or the average value of the current area can be obtained. In most cases, the data useful for classification is not necessarily the maximum or the average value. Traditional CNN leads to the loss of the local important information in this convolution area to a certain extent, while some useless information is retained instead.

In our ACNN model, we introduce the Attention mechanism to solve the word order problem in intention detection. We use the combination of global average pooling and Attention mechanism to replace the

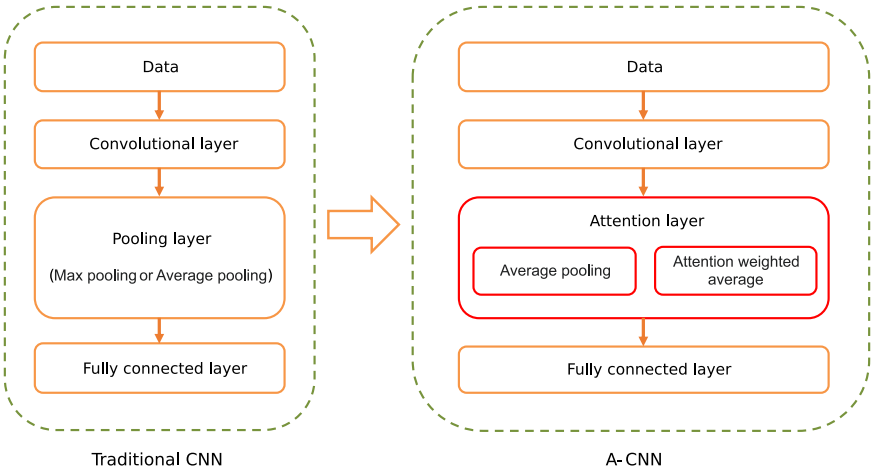

Fig. 3. Improvement of convolutional neural network.

original pooling layer. On one hand, the use of global average pooling can better reduce the data dimension and network parameters. On the other hand, the use of the Attention mechanism makes it easier for models to learn semantic features and retain valid information. The new features extracted from both can be used to improve the local and global connection missing problems caused by the pooling layer. The model structure is shown in Fig. 3. Using global average pooling can regularize the whole network structure to prevent overfitting. It can be used to extract useful information while reducing the data dimension. At the same time, it is combined with the Attention mechanism to learn the edge characteristics.

In the Attention layer, we integrate the k local feature vectors into one vector with the summation of weights, which is the output of the convolution layer. After cascading the Attention networks, A-CNN can better handle the relationship between local and global information. Global features can better express useful local features and weaken the useless part of the local ones. Moreover, global features can also avoid such situations that both useful and useless local features have the same weight in the global features and have the same contribution to intention recognition.

This Attention layer uses a Self-Attention mechanism. In Self-Attention, Query, Key, and Value are the input word sequence information. The self-attention mechanism can be divided into three stages. First, the similarity is calculated by a Query and a certain Key. The formula is dot product.

$$\text{Similarity (Query, Key}_i\text{)=Query} * \text{Key}_i \tag{6}$$

Second, a calculation method is introduced which is similar to SoftMax. It numerically converts the scores at the previous stage. On one hand, the original calculated scores are sorted into a probability distribution with the sum of all element weights. On the other hand, the weight of important elements can also be more prominent through the internal mechanism of SoftMax. The following formula is shown.

$$a_i = \text{Softmax}\left(\text{Sim}_i\right) = \frac{e^{\text{sim}_i}}{\sum_{j=1}^{L_x} e^{\text{sim}_j}} \tag{7}$$

The result a_i obtained at the second stage is the weight coefficient corresponding to $Value_i$, and the Attention value can be obtained by weighted summation, the formula is:

$$\text{Attention(Query, Source)} = \sum_{i=1}^{L_x} a_i * \text{Value}_i \tag{8}$$

After the above calculations, the Attention value for Query can be obtained. The specific structure of the Attention layer is shown in Fig. 4:

Finally, the results obtained from the average pooling layer and the attention layer are cascaded and input to the model full connection layer. The data features are mapped to 1 space through the full connection layer to achieve classification.

Table 1
The format of the data set.

File	Property	
News_info_train.txt	News ID	News text
News_label_train.txt	News ID	News label
News_info_validate.txt	News ID	News text
News_label_validate.txt	News ID	News label
News_info_test.txt	News ID	News text
News_label_test.txt	News ID	News label

5. Experiments

5.1. Experiment environment

All experiments in this paper are run under Win10 x64 operating system. The CPU is AMD Ryzen 9 3950X CPU@3.50 GHz. The memory size is 128G. The version of Python is 3.6.8. The deep learning framework is Keras 2.2.4, and the corresponding machine learning framework is Scikit-learn 0.21.3. The GPU is GeForce RTX 2080 Ti @11GB. All experiments are done using CPU or GPU to get results, and the experiment result is the average value of 10 times experiments.

5.2. Data set

The experiment data in this article comes from the 2018 Second Sohu Content Recognition Algorithm Competition. The text of data set is original HTML format Zhan (2018). The data set was officially announced by the competition in March 2018 and could be downloaded on the official website of the Second Sohu Algorithm Competition (Zhan, 2018). The training data set provided by the competition includes 48 480 news texts. The amount of training, validation, and testing data is 60%, 20%, 20% respectively. There are 24 240 positive samples and 24 240 negative samples. The data files used in this article and the attributes are shown in Table 1.

5.3. Model comparison

5.3.1. A-CNN vs. traditional machine learning

Feature selection is made to represent word vectors with BOW bag-of-bags model (Cao et al., 2010), represent word frequency features and LSI for dimension reduction (Altszyler et al., 2017) with TF-IDF (Zhu et al., 2019). We have made experiments to compare our model with tractional machine learning such as LightGBM, Xgboost, GradientBoosting, and Random Forest. Meanwhile, we have made experiments to compare with the Stacking model with the integration of all above methods. Stacking combines limited models

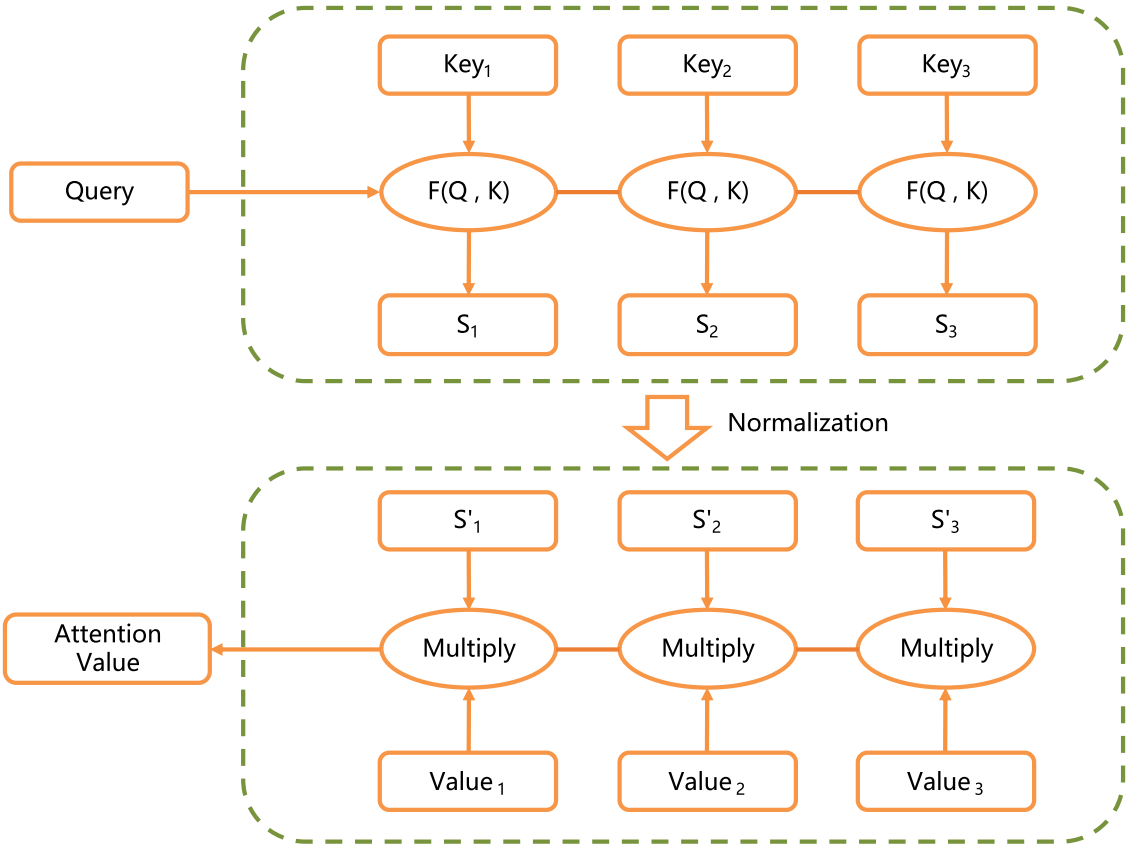

Fig. 4. Attention layer of A-CNN.

Table 2
Experiment results of A-CNN and traditional machine learning models.

Model	Accuracy	Precision	F1	AUC
GradientBoosting	0.7039	0.6799	0.6891	0.7036
RandomForest	0.6913	0.6731	0.6698	0.6899
Xgboost	0.7061	0.6860	0.6881	0.7052
Lightgbm	0.6983	0.6821	0.6760	0.6967
Stacking	0.7075	0.6956	0.6831	0.7054
A-CNN	**0.7878**	**0.7815**	**0.7537**	**0.7923**

to estimate non-parametric density to reduce generalization errors. Although our previous work (Wang et al., 2019) illustrated stacking strategy is effective for marketing intention detection, there is room for improvement.

The experiment process is as follows. First, we process the raw data, removing HTML tags and stop words. Second, each sentence is cut into words. Next, after data cleaning, we employ our proposed FSLText to extract data features. Finally, words and text are serialized to be put into the embedding layer of our A-CNN. In the traditional CNN, the data is sent to the pooling layer for average pooling or maximum pooling after it completes the convolution operation (Kim, 2014). However, our A-CNN performs convolution operations on the data as well. The difference is that A-CNN sends the convolutional data into the Attention layer. The Attention layer has two functions. On one hand, it performs an average pooling operation on the convolutional data. On the other hand, its self-attention model performs a weighted average operation on the convolutional data. The results of the above two operations are combined into the fully-connected layer.

The experiment results of A-CNN and traditional machine learning models are shown in Table 2. Compared with traditional machine learning models, A-CNN has improved the accuracy by 11.35% in comparison with the next highest Stacking method. In terms of F1 score, A-CNN has increased by 9.37% in comparison with the next-highest GradientBoosting. In terms of AUC index, A-CNN has increased by 12.32% in comparison with the next-highest Stacking method. Some factors lead to this result. The BOW (bag-of-words) model that is often used in traditional machine learning cannot better express the order of words in a sentence. The same words but different order may lead to completely different meanings of the text. Second, machine learning models with high classification effects are mainly based on the decision tree method. One disadvantage of decision trees is that they ignore the correlation of attributes in the data set. The word order loss problem and the decision tree's ignorance of the centralized attributes lead to the fact that its classification effect is not as good as A-CNN. A-CNN's FSLText word vector representation method converts bag-of-words into bag-of-features, which can use the word order in the context to help the neural network make judgments. A-CNN performs a weighted average on the convolutional data on the Attention layer, which can better deal with the association between local features and global features, and will not easily ignore centralized attributes. Therefore, the accuracy, F1 score, and AUC value of A-CNN are generally higher than the results obtained by traditional machine learning methods based on decision trees. This proves that A-CNN is more effective on the intent recognition problem than traditional machine learning methods which are based on decision trees.

5.3.2. A-CNN vs. deep learning

Next, we have conducted experiments to compare the A-CNN method we proposed with BiLSTM, ATT-BiLSTM Zhou et al. (2016), CNN, multi-channel CNN (Xu et al., 2017), ATT-CNN Zhao and Wu (2016), RCNN and Transformer. As is shown on Table 3, A-CNN's AIC index is significantly ahead of all comparative deep learning models' at the same level, which proves that the A-CNN model is low in complexity and it avoids the appearance of overfitting. At the same time, the performances of A-CNN under various evaluation indicators are in

Table 3
Experimental results of A-CNN and deep learning.

Model	Accuracy	Precision	Recall	F1	AUC	AIC
RCNN	0.6953	0.7333	0.6929	0.7125	0.6972	4 670 215.9863
CNN	0.7162	0.7803	0.7324	0.7506	0.7739	4 704 345.7348
Multi-channel CNN	0.7183	0.7168	0.7235	0.7199	0.7828	148 460 394.9302
BiLSTM	0.6968	0.7030	0.6201	0.6589	0.6928	4 707 286.2898
Att-BLSTM	0.7421	0.6779	0.7079	0.6926	**0.7939**	4 670 058.3336
Transformer	0.6884	0.6739	0.6596	0.6667	0.6869	4 707 549.8246
ATT-CNN	0.7164	0.7187	0.7109	0.7147	0.7898	102 482 617.0083
A-CNN	**0.7878**	**0.7815**	**0.8717**	**0.7537**	0.7923	**177 011.1797**

Table 4
Experimental results of A-CNN, BCNN and ABCNN.

Model	MAP	Precision	Recall	F1	AIC
ABCNN1 (1 conv layers)	0.5143	0.4723	0.5521	0.5091	263 888.3134
ABCNN2 (1 conv layers)	0.5038	0.4723	0.4939	0.4828	108 162.8780
ABCNN3 (1 conv layers)	0.5107	0.4723	0.5342	0.5014	263 890.1252
BCNN (1 conv layer)	0.5191	0.4723	0.4549	0.4634	**107 876.0220**
ABCNN1 (2 conv layers)	0.5234	0.4723	0.4660	0.4691	309 803.3832
ABCNN2 (2 conv layers)	0.5038	0.4723	0.5167	0.4935	128 335.0713
ABCNN3 (2 conv layers)	0.5277	0.4723	0.3675	0.4134	309 087.1499
BCNN (2 conv layers)	0.5052	0.4723	0.4138	0.4412	127 833.1113
A-CNN	**0.7878**	**0.7815**	**0.8717**	**0.7537**	177 011.1797

the leading position. The accuracy value is second only to CNN and ATT-BiLSTM, the F1 score is second only to CNN, and the AUC value is second only to ATT-CNN and multi-channel CNN. Because Precision and Recall are negatively correlated, it is impossible for the two values to be at a high level simultaneously. The Recall value of A-CNN is at a leading level, which means that it can more effectively detect news with marketing intent. Compared with the comparative deep learning models, the A-CNN we proposed has some advantages. First of all, the A-CNN model has low complexity and is faster when processing high-dimensional data, while the multi-channel CNN and ATT-CNN have higher model complexity and are prone to overfitting. Second, the addition of the Attention layer in A-CNN solves the problem of data loss in the pooling layer without losing important information in long texts. At the same time, A-CNN also inherits the shared convolution kernel method in CNN and uses parallel operations to reduce its training time. The experiment results prove that our A-CNN can not only maintain a sound experiment result, but also has a training speed far exceeding the same-level deep learning models, and A-CNN will not increase the model complexity on the purpose of improving experiment results.

5.3.3. A-CNN vs. ABCNN

In addition, there is also another similar Attention-based CNN called ABCNN (Yin et al., 2016). The ABCNN has 3 basic architectures based on BCNN: ABCNN1, ABCNN2 and ABCNN3. ABCNN1 employs the attention operation on the vector representation of the input sentence on the data before the convolutional layer, thereby influencing the convolutional network. ABCNN2 employs an attention operation on the convolutional data before the pooling layer, and performs a weighted average in the pooling layer. ABCNN3 is a combination of this two. ABCNN2 is similar to our A-CNN idea. However, our A-CNN performs the attention operation on the convolutional data, and meanwhile it also performs average pooling on the data of the pooling layer. After these operations, the data is simultaneously sent to the fully-connected layer for intent detection and classification. Therefore, our A-CNN combines the weighted average idea in the Attention mechanism and the average pooling idea of the traditional CNN pooling layer. A-CNN is not easy to lose important information. Meanwhile, it also combines the word order of the context to a greater extent. It has a better effect on the intent detection. We refactor the ABCNN to address text classification issue, and make more experiments with our A-CNN. The results are shown in Table 4.

As shown in Table 4, the Precision, Recall and F1 scores of A-CNN are significantly ahead of the ABCNN model, and there is no obvious difference between the AIC index values of A-CNN and ABCNN. Meanwhile, by comparing the experiment results of the sub-models of ABCNN, it can be found that the Precision, Recall and F1 scores of multiple sub-models are basically at the same level, and there is no apparent difference. This shows that, ABCNN, which is used to deal with text similarity and text matching problems, cannot be easily transferred to the text classification problem. It indirectly proves that although our A-CNN and ABCNN2 are similar in concept, there is a clear difference in model implementation. Compared with the model structure of ABCNN2, A-CNN does not just use the weighted summation of the Attention model. It also retains the average summation of the traditional pooling layer, which directly proves the difference between the two. In terms of the problems to be solved, ABCNN is more suitable for solving the text matching problem, while the model structure of A-CNN determines that it is more suitable for solving the problem of marketing intent recognition.

6. Conclusions and future work

The vigorous development of online media and the low-cost threshold of malicious marketing news have led to the fact that a large number of malicious marketing news has mixed among various online news. Our proposed A-CNN can solve the problem of disconnection between local features and global features of the text. Besides, it employs the FSL word vector to avoid the impact brought by word ambiguity on intent detection. The accuracy and F1 score of the experiment results show the effectiveness of our A-CNN network. Our method can be used to detect marketing news in cyberspace, purify cyberspace, and improve users' reading experience.

This article integrates Attention model with CNN to solve the problem of marketing intent detection. Future directions are concluded as follows. First, we should pay more attention to different combinations of other neural networks to improve the network structure. Second, the network training time of deep learning should be optimized for the structure or parameters of this model.

CRediT authorship contribution statement

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

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Al-Rawi, Ahmed, 2019. Viral news on social media. Digit. Journal. 7 (1), 63–79.

Alpaydin, Ethem, 2020. Introduction to Machine Learning. MIT press.

Altszyler, Edgar, Sigman, Mariano, Slezak, Diego Fernández, 2017. Corpus specificity in lsa and word2vec: the role of out-of-domain documents. arXiv preprint arXiv:1712.10054.

Bengio, Yoshua, Ducharme, Réjean, Vincent, Pascal, Jauvin, Christian, 2003. A neural probabilistic language model. J. Mach. Learn. Res. 3 (Feb), 1137–1155.

Breiman, Leo, 1996. Bagging predictors. Mach. Learn. 24 (2), 123–140.

Cao, Yang, Wang, Changhu, Li, Zhiwei, Zhang, Liqing, Zhang, Lei, 2010. Spatial-bag-of-features. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, pp. 3352–3359.

Chen, Liqun, Dai, Shuyang, Tao, Chenyang, Zhang, Haichao, Gan, Zhe, Shen, Dinghan, Zhang, Yizhe, Wang, Guoyin, Zhang, Ruiyi, Carin, Lawrence, 2018. Adversarial text generation via feature-mover's distance. In: Advances in Neural Information Processing Systems. pp. 4666–4677.

Chen, Tianqi, Guestrin, Carlos, 2016. Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining. pp. 785–794.

Chen, Tao, Xu, Ruifeng, He, Yulan, Wang, Xuan, 2017. Improving sentiment analysis via sentence type classification using bilstm-crf and cnn. Expert Syst. Appl. 72, 221–230.

Couronné, Raphael, Probst, Philipp, Boulesteix, Anne-Laure, 2018. Random forest versus logistic regression: a large-scale benchmark experiment. BMC Bioinform. 19 (1), 270.

Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton, Toutanova, Kristina, 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Fan, Yin, Lu, Xiangju, Li, Dian, Liu, Yuanliu, 2016. Video-based emotion recognition using cnn-rnn and c3d hybrid networks. In: Proceedings of the 18th ACM International Conference on Multimodal Interaction. pp. 445–450.

Friedman, Jerome H., 2002. Stochastic gradient boosting. Comput. Stat. Data Anal. 38 (4), 367–378.

Friedman, Jerome, Hastie, Trevor, Tibshirani, Robert, 2001. The Elements of Statistical Learning, Vol. 1. Springer series in statistics New York.

Hochreiter, Sepp, Schmidhuber, Jürgen, 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.

Ke, Guolin, Meng, Qi, Finley, Thomas, Wang, Taifeng, Chen, Wei, Ma, Weidong, Ye, Qiwei, Liu, Tie-Yan, 2017. Lightgbm: A highly efficient gradient boosting decision tree. In: Advances in Neural Information Processing Systems. pp. 3146–3154.

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

Kowsari, Kamran, Jafari Meimandi, Kiana, Heidarysafa, Mojtaba, Mendu, Sanjana, Barnes, Laura, Brown, Donald, 2019. Text classification algorithms: A survey. Information 10 (4), 150.

Lai, Siwei, Xu, Liheng, Liu, Kang, Zhao, Jun, 2015. Recurrent convolutional neural networks for text classification. In: Twenty-Ninth AAAI Conference on Artificial Intelligence.

Landauer, Thomas K., Foltz, Peter W., Laham, Darrell, 1998. An introduction to latent semantic analysis. Discourse Process. 25 (2–3), 259–284.

Li, Jiwei, Monroe, Will, Shi, Tianlin, Jean, Sébastien, Ritter, Alan, Jurafsky, Dan, 2017. Adversarial learning for neural dialogue generation. arXiv preprint arXiv:1701.06547.

Liang, Yan, Zhang, Wenyun, 2019. The impact of new media environment on public opinion dissemination and the coping strategies.

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

Pennington, Jeffrey, Socher, Richard, Manning, Christopher D., 2014. Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543.

Peters, Matthew E, Neumann, Mark, Iyyer, Mohit, Gardner, Matt, Clark, Christopher, Lee, Kenton, Zettlemoyer, Luke, 2018. Deep contextualized word representations. arXiv preprint arXiv:1802.05365.

Poma, Yutzil, Melin, Patricia, González, Claudia I, Martínez, Gabriela E, 2020a. Filter size optimization on a convolutional neural network using fgsa. In: Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications. Springer, pp. 391–403.

Poma, Yutzil, Melin, Patricia, González, Claudia I, Martinez, Gabriela E, 2020b. Optimal recognition model based on convolutional neural networks and fuzzy gravitational search algorithm method. In: Hybrid Intelligent Systems in Control, Pattern Recognition and Medicine. Springer, pp. 71–81.

Ramos, Juan, et al., 2003. Using tf-idf to determine word relevance in document queries. In: Proceedings of the First Instructional Conference on Machine Learning, Vol. 242. Piscataway, NJ. pp. 133–142.

Sabour, Sara, Frosst, Nicholas, Hinton, Geoffrey E., 2017. Dynamic routing between capsules. In: Advances in Neural Information Processing Systems. pp. 3856–3866.

Salehinejad, Hojjat, Sankar, Sharan, Barfett, Joseph, Colak, Errol, Valaee, Shahrokh, 2017. Recent advances in recurrent neural networks. arXiv preprint arXiv:1801.01078.

Salton, Gerard, Wong, Anita, Yang, Chung-Shu, 1975. A vector space model for automatic indexing. Commun. ACM 18 (11), 613–620.

Sánchez, Daniela, Melin, Patricia, Castillo, Oscar, 2020. Comparison of particle swarm optimization variants with fuzzy dynamic parameter adaptation for modular granular neural networks for human recognition.. J. Intell. Fuzzy Syst. 38 (3), 3229–3252.

Sun, Junyi, 2012. Jieba. In: Chinese Word Segmentation Tool.

Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N, Kaiser, Łukasz, Polosukhin, Illia, 2017. Attention is all you need. In: Advances in Neural Information Processing Systems. pp. 5998–6008.

Wang, Yufeng, Liu, Shuangrong, Li, Songqian, Duan, Jidong, Hou, Zhihao, Yu, Jia, Ma, Kun, 2019. Stacking-based ensemble learning of self-media data for marketing intention detection. Future Internet 11 (7), 155.

Wang, Yufeng, Ma, Kun, Garcia-Hernandez, Laura, Chen, Jing, Hou, Zhihao, Ji, Ke, Chen, Zhenxiang, Abraham, Ajith, 2020. A clstm-tmn for marketing intention detection. Eng. Appl. Artif. Intell. 91, 103595.

Wolpert, David H., 1992. Stacked generalization. Neural Netw. 5 (2), 241–259.

Xi, Yun, Zhuang, Xutian, Wang, Xinming, Nie, Ruihua, Zhao, Gansen, 2018. A research and application based on gradient boosting decision tree. In: International Conference on Web Information Systems and Applications. Springer, pp. 15–26.

Xu, Jun, Zhang, Lei, Zhang, David, Feng, Xiangchu, 2017. Multi-channel weighted nuclear norm minimization for real color image denoising. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1096–1104.

Xue, Kunnan, Xue, Yueju, Mao, Liang, Liu, Hongshan, 2016. Bag of convolutional words networks for visual recognition. Comput. Eng. Appl. 2016 (21), 31.

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

Yin, Wenpeng, Schütze, Hinrich, Xiang, Bing, Zhou, Bowen, 2016. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. Trans. Assoc. Comput. Linguist. 4, 259–272.

Zhan, Zecheng, 2018. Solution of luckyrabbit team in 2018 second sohu content recognition algorithm competition. https://github.com/zhanzecheng/SOHU_competition.

Zhang, Chenchen, Wang, Xingjuan, Yu, Shuiyuan, Wang, Yongbin, 2018. Research on keyword extraction of Word2vec model in Chinese corpus. In: 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS). IEEE, pp. 339–343.

Zhao, Jianfeng, Mao, Xia, Chen, Lijiang, 2019. Speech emotion recognition using deep 1d & 2d cnn lstm networks. Biomed. Signal Process. Control 47, 312–323.

Zhao, Zhiwei, Wu, Youzheng, 2016. Attention-based convolutional neural networks for sentence classification. In: INTERSPEECH. pp. 705–709.

Zhou, Peng, Shi, Wei, Tian, Jun, Qi, Zhenyu, Li, Bingchen, Hao, Hongwei, Xu, Bo, 2016. Attention-based bidirectional long short-term memory networks for relation classification. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 207–212.

Zhu, Zhiliang, Liang, Jie, Li, Deyang, Yu, Hai, Liu, Guoqi, 2019. Hot topic detection based on a refined tf-idf algorithm. IEEE Access 7, 26996–27007.