

## Review of Text Classification Methods on Deep Learning

Hongping Wu<sup>1</sup>, Yuling Liu<sup>1, \*</sup> and Jingwen Wang<sup>2</sup>

**Abstract:** Text classification has always been an increasingly crucial topic in natural language processing. Traditional text classification methods based on machine learning have many disadvantages such as dimension explosion, data sparsity, limited generalization ability and so on. Based on deep learning text classification, this paper presents an extensive study on the text classification models including Convolutional Neural Network-Based (CNN-Based), Recurrent Neural Network-Based (RNN-based), Attention Mechanisms-Based and so on. Many studies have proved that text classification methods based on deep learning outperform the traditional methods when processing large-scale and complex datasets. The main reasons are text classification methods based on deep learning can avoid cumbersome feature extraction process and have higher prediction accuracy for a large set of unstructured data. In this paper, we also summarize the shortcomings of traditional text classification methods and introduce the text classification process based on deep learning including text preprocessing, distributed representation of text, text classification model construction based on deep learning and performance evaluation.

**Keywords:** Text classification, deep learning, distributed representation, CNN, RNN, attention mechanism.

### 1 Introduction

With the rapid development of the internet and big data, the internet information increases explosively. As an efficient information retrieval and mining technology, text classification has achieved extensive attention by natural language processing researchers, which also plays an important role in the management of internet information. Text classification can effectively extract valuable information and automatically classify them from massive information data by using natural language processing, text mining, machine learning techniques and deep learning. At present, common text classification applications include sentiment analysis [Tang, Qin, Wei et al. (2015)], news text classification [Li, Shang and Yan (2016)], and topic classification.

Text classification mainly includes binary classification, multi-classification and multi-label classification which indicates that a text may belong to one or more categories simultaneously. The main task of text classification is to divide a given dataset of text into one or more categories according to their content. If we want to build a text

---

<sup>1</sup> College of Computer Science and Electronic Engineering, Hunan University, Changsha, 410082, China.

<sup>2</sup> Department of Computer Science, Elizabethtown College, PA, 17022, USA.

\*Corresponding Author: Yuling Liu. Email: yuling\_liu@126.com.

Received: 15 February 2020; Accepted: 25 February 2020.

classification system, we first need to divide the datasets into training set and test set, then we use the labeled training set to train the classification model, and finally we use this model to predict the category results of the test set.

The major challenges of building a well-performed text classification system are text representation and classification model. Existing machine learning methods can rarely process text data directly, while text representation can make text data mathematically computable by converting the text data into the numerical data. Text presentation is a mapping process, which maps a word in a text space to numerical vector space by a certain method. Traditional text presentations are discrete representations, such as one-hot encoding, Bag-of-Words [Kesorn and Poslad (2012)], TF-IDF [Soucy and Mineau (2005); Wu, Luk, Wong et al. (2008)], N-Gram and so on. General natural language processing problems can be solved by discrete representation, but there are many problems for scenes with high accuracy, such as high dimension, sparse matrix, no semantic retention and so on. In order to improve the accuracy of the model, the researchers proposed the distributed representation of text, in which the main idea is to map each word to a shorter word vector through training. Common distributed representations include collinear matrices and word embedding. In the neural network language model, word embedding can illustrate the semantic relationship between words.

Based on the assumption that classification tasks obey a probability distribution, the traditional text classification models use Bayes theory to obtain the classifier, but if the hypothesis is not true, it will affect the classification accuracy. There are several traditional text classification models [Miao, Zhang, Jin et al. (2018)], such as K-Nearest Neighbor classification algorithm [Xu and Liu (2008)], Support Vector Machine algorithm [Wei, Guo, Yu et al. (2013)], Bayesian classification algorithm [Gong and Yu (2010)], and Decision Tree [Johnson, Oles, Zhang et al. (2002)]. But traditional text classification methods based on machine learning have many defects such as dimension explosion, data sparsity and limited generalization ability.

The concept of deep learning has been put forward for the first time since 2006, which has made great breakthroughs in image recognition, speech recognition and so on. In recent years, deep learning has also shown the best results in text classification task. Compared with the traditional machine learning algorithm, deep learning extracts more abstract features from the input features by deepening the number of levels of the model, which makes the final classification information of the model more reliable. The deeper the level of the model, the more reliable the abstract features are extracted. As a result, the number of parameters of the model increases geometrically as well as the training time of the model. However, with the advent of the era of big data, enough training samples can overcome this risk.

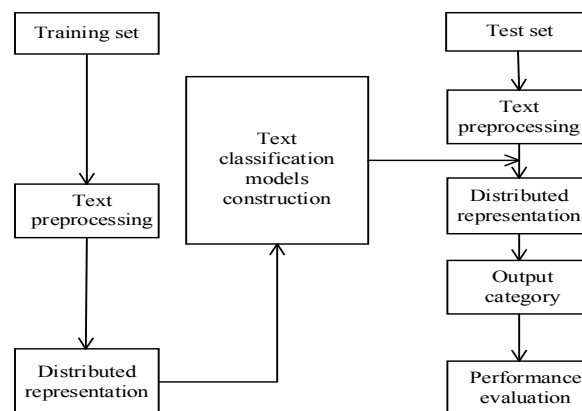
The text classification models based on deep learning greatly improve the problems caused by traditional machine learning methods, which avoid cumbersome feature extraction process and have higher prediction accuracy. Many text classification models based on deep learning have been explored. For example, Kim [Kim (2014)] first proposes the application of the Convolutional Neural Network (CNN) in text classification task and obtains excellent classification results, which also inspires us to use deep learning methods in some complex text structures. Later, Recurrent Neural

Network (RNN) is also becoming more and more popular in text classification. Liu et al. [Liu, Qiu and Huang (2016)] introduce three text classification methods based on multi-task learning of RNN. Based on the text RNN model, Attention Mechanism [Yang, Yang, Dyer et al. (2016)] is also added into the RNN model, which can solve the problem of long-term dependence of text, directly present the importance of each word. And some researchers combine the advantages of CNN and RNN by using them to extract global long-term dependencies and local semantic features.

This paper not only introduces the text classification process based on deep learning, but also focuses on the classical classification models based on deep learning in recent years. Section 1 introduces the research background and signification on text classification, and analyzes the advantages and disadvantages of text representation and classification model based on traditional methods and deep learning methods. Section 2 presents the text classification process based on deep learning, which including text preprocessing, distributed representation of text, text classification models based on deep learning and performance evaluation. Section 3 analyzes and summarizes the various text classification models based on deep learning, which including CNN-based, RNN-based, Attention Mechanism-based. Section 4 draws the conclusion.

## 2 Text classification process based on deep learning

The main challenges of text classification include extracting the text features and training the classification models. In text classification methods based on machine learning, features extraction and classification model are two completely separate parts, which are studied separately in most cases. Tradition methods need artificial feature extraction, which have complex feature extraction process and low accuracy. Compared with the tradition methods, the feature extraction of the text classification based on deep learning are obtained by multi-layer complex feature extraction of the artificial neural network, which can achieve higher accuracy, faster training speed and stronger interpretability. The text classification process based on deep learning includes text preprocessing, distributed representation of text, text classification model construction based on deep learning and performance evaluation. Fig. 1 shows the text classification process.



**Figure 1:** Text classification process based on deep learning

### ***2.1 Text preprocessing***

The primary task of text classification is text preprocessing, which converts text into clean word sequences. In general, text preprocessing mainly includes word segmentation, word stemming, stop words removing, and low-frequency words removing. When preprocessing Chinese texts and English texts, there are several special points, as follows:

(1) Simple English texts have no word segmentation process because it can be done through spaces and punctuation. However, word segmentation of Chinese texts is essential, which needs word segmentation algorithm to complete word segmentation, such as forward maximum matching [Li and Chen (2014)], reverse maximum matching [Zhang, Li and Meng (2006)], bidirectional maximum matching [Gai, Gao, Duan et al. (2014)]. At present, open source Chinese word segmentation tools have also emerged, such as, Paoding Analyzer, IK Analyzer and Jieba and so on.

(2) In both Chinese and English texts, we need to eliminate stop words. The English texts stop words include “a”, “the”, some punctuation marks and so on. The removal of the stop word is to filter the noise of the segmentation result, which makes the text classification more accurate.

(3) For English texts, English words have singular, plural and various tenses, so we want every word to be converted to its original form. For example, “tree” and “trees”. In practical applications, NLTK is generally used for word stemming and word lemmatization.

(4) Most English texts coding is utf-8, but Chinese texts coding must be Unicode.

### ***2.2 Distributed representation of text***

In the traditional text classification methods based on machine learning, one-hot is a common method for text representation. This method represents each word in the text as a vector, where its dimension is the number of vocabularies in the preprocessed text. However, this method has obvious limitations. On the one hand, if the whole data is large and the vocabulary contains a large number of words, the text vector dimension will be too high, which seriously affects the calculation efficiency. On the other hand, one-hot ignores the semantic information of the context, which causes serious information loss.

In order to overcome the above defects, Hinton [Hinton (1986)] proposes the concept of Word Embedding. Word Embedding is a distributed representation [Davis and Fonseca (2007)]. The main idea of this method is to map words from high-dimensional space to low-dimensional space, which solves the problem of vector sparsity. And after mapping to low-dimensional space, the position relation between the corresponding word vectors of different words reflects their contextual semantic information [Bian, Gao and Liu (2014); Hu, Tang, Chen et al. (2016)]. To train word embedding faster and more efficiently, Mikolov proposes two neural network language models, CBOW and Skip-Gram [Mikolov, Sutskever, Chen et al. (2013); Goldberg and Levy (2014)]. CBOW is used to predict the current word according to the context, whereas Skip-Gram is used to predict the context according to the current word. Currently there are many training tools for word embedding, and the most representative are Google Word2Vec [Tomas (2017)], GloVe [Jeffrey, Richard and

Christopher (2017)] of Stanford University and Python Library Genism [Rare (2017)].

### 2.3 Text classification models based on deep learning

In recent years, deep learning has achieved success in computer vision [Krizhevsky, Sutskever and Hinton (2012)], speech recognition [Senior, Vanhoucke, Nguyen et al. (2012)] and natural language processing [Socher, Perelygin, Wu et al. (2013)]. Scholars have shifted the focus of text classification from the traditional machine learning to the artificial neural networks. Artificial neural network can extract abstract hierarchical features from complex original data, and it has strong nonlinear mapping ability. One of the advantages of using a neural network to process text classification is that there is no need to spend a lot of time on feature extraction and selection, and the distributed representation of words as features input into the network. Then the neural network can automatically extract valuable information for text classification task. At present, there are many text classification models based on deep learning, including CNN-based, RNN-based, and the relevant Attention Mechanism. The variants of these three models based on them will be discussed in detail in the next section.

### 2.4 Performance evaluation

Performance evaluation is the last stage of training a text classification model. Precision and recall are often used to evaluate the performance of a text classification model. The precision is calculated by Eq. (1) and the recall is calculated by Eq. (2), where  $TP$  is the number of true positive classes,  $FP$  is the number of false positive classes and  $FN$  is the number of false negative classes.

$$precision = \frac{TP}{TP + FP} \quad (1)$$

$$recall = \frac{TP}{TP + FN} \quad (2)$$

We expect the accuracy of classification results and recall to be high at the same time, but they sometimes produce conflicts. In order to observe the two performances at the same time, scholars put forward the value of  $F1$  based on the precision and recall. The  $F1$  is calculated by Eq. (3).

$$F1 = \frac{precision * recall * 2}{precision + recall} \quad (3)$$

## 3 Deep learning models using for text classification

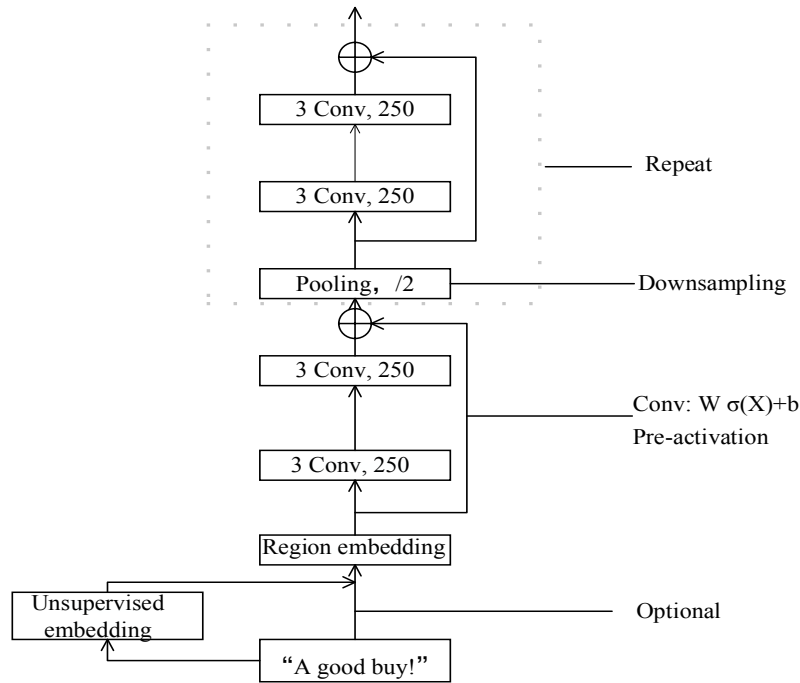
This section focuses on the text classification methods based on deep learning, which include CNN-based, RNN-based and Attention Mechanism-based.

### 3.1 CNN-based

Convolutional neural network is a kind of multi-layer complex neural network structure, which has been widely used in our life and changed our life to some extent. For example,

in the field of image recognition, Wang et al. [Wang, Qin, Xiang et al. (2019)] propose CAPTCHA recognition methods based on deep CNN. Pan et al. [Pan, Qin, Chen et al. (2019)] propose a food recognition algorithm based on CNN. Moreover, Pan et al. [Pan, Qin, Xiang et al. (2019)] also combine CNN with agricultural products and propose a disease monitoring system for agricultural products. In the field of text classification, Kim proposes an effective text classification method by combining CNN with natural language. He uses a CNN with a convolutional layer for text classification and compares different methods such as random initialization, preprocess word embedding, static input matrix and dynamic input matrix. Finally, he concludes that the static input matrix has the best classification effect. Kalchbrenner et al. [Kalchbrenner, Grefenstette and Blunsom (2014)] propose a similar model called the Dynamic Convolutional Neural Network (DCNN). Unlike the CNN method proposed by Kim, DCNN contains five convolutional layers and multiple temporary k-max pool layers. The k-max pooling extracts the k top values from a series of convolutional filters and make sure the output length is fixed. Moreover, Johnson et al. [Johnson and Zhang (2014)] also propose a similar model. Their model uses up to six convolutional layers and three fully connected layers. Because the combination of CNN and RNN in the field of computer vision has achieved good success, Xiao et al. [Xiao and Cho (2016)] also combine RNN and CNN in sentence classification. Their model uses a convolutional network with up to five layers to learn high-level features, and these high-level features are also used as input to the LSTM.

In the previous text classification works, CNNs use a rather superficial architecture, where their convolutional layers depth is up to six. Since shallow CNN can only extract local features with limit window size, Conneau et al. [Conneau, Schwenk, Barrault et al. (2016)] propose a very deep CNN to extract hierarchical local features in text classification. Their convolutional layer depth is up to 29. And the architecture achieves stable performance on eight freely available large-scale datasets. This is the first evidence that depth is good for convolutional neural network. Similarly, Johnson et al. [Johnson and Zhang (2017)] propose a Deep Pyramid Convolutional Neural Network (DPCNN), which carefully study the depth of word-level CNN. This novel DPCNN structure can effectively extract the features of long-range associations and obtain more global information. Fig. 2 shows the structure of this DPCNN model. Firstly, this model inputs a sentence "A good buy!" into the text region embedding layer, which uses word embedding to generate vector representations for each word in the sentence. It is followed by stacking of two convolution blocks and a shortcut. They fix the number of feature maps to 250 and the kernel size to 3. And the shortcut connections with pre-activation  $W\sigma(x)+b$  and identity mapping for enable training of deep networks. Downsampling can effectively represent more global information in the text. In this model, the step size of the downsampling is 2. This method uses the unsupervised embedding to train the text region embedding for improving the accuracy and reducing training time.



**Figure 2:** The architecture of DPCNN model

However, most of the CNN-based methods use a fixed window size, thus variable n-gram features cannot be extracted. Wang et al. [Wang, Huang and Deng (2018)] propose a densely connected CNN with multi-scale feature to extract variable n-gram features for text classification. The reason why the dense connections create shortcut paths between upstream and downstream convolutional blocks is to combine features of a smaller scale into features of a large scale, which produce variable n-gram features [Ma, Yu, Tian et al. (2019)]. Although CNN-based approaches have played a great advantage in extracting of variable n-gram features, they focus on the local continuous word sequence and ignore the global word co-occurrence information in the corpus. Moreover, the local semantic features extracted by CNN also expose a disadvantage, which is redundancies. Yao et al. [Yao, Mao and Luo (2019)] propose a novel Graph Convolutional Network (GCN) for text classification. GCN can capture document and word relationships, and global word co-occurrence information. Firstly, we need to generate a graph representation from unstructured data. The nodes consist of documents and words in the corpus. The edges among nodes are composed of the document-word edges and word-word edges, where the word-word edges are word co-occurrence in the whole corpus. And the weight of the edge is calculated by the Eq. (4).

$$A_{i,j} = \begin{cases} PMI(i,j) & i, j \text{ are words, } PMI(i,j) > 0 \\ TF-IDF_{i,j} & i \text{ is document, } j \text{ is word} \\ 1 & i = j \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

After building the text graph, the authors input the graph into a simple two layers GCN for feature extraction. The first layer activation function uses ReLU and the second layer uses Softmax Function. The propagation mode between layers is calculated by the Eq. (5).

$$Z = \text{soft max}(\tilde{A} \text{ ReLU}(\tilde{A} X W_0) W_1) \quad (5)$$

where  $A$  is an adjacency matrix of text graph, and the  $\tilde{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$  is the normalized symmetric adjacency matrix. The  $W_0$  and  $W_1$  are weight matrix. Compared with the previous CNN-based models, the text GCN model achieves the better sort effect on multiple text classification benchmark data sets, which shows better robustness. Although text GCN can produce better text classification results, it cannot quickly generate embedding. In the future, we can use attention mechanism in GCN to improve classification performance and develop unsupervised text GCN for large-scale unlabeled text data representation learning.

### 3.2 RNN-based

At present, the Recurrent Neural Networks has been widely used in machine translation, speech recognition, image description generation and other sequence data processing tasks. The bidirectional recurrent structure is introduced into RNN, which solves the problem of interrelation between input information. RNN has great advantages when modeling sequentially in text sequences and long-term dependencies [Chung, Gulcehre, Cho et al. (2014)]. The main application model of text classification is bidirectional recursive neural network (BRNN), which is proposed by Socher et al. [Socher, Pennington, Huang et al. (2011)]. The bidirectional recursive structure assumes that the current output is related to the previous information and the following information, which can capture global long-term dependencies. Therefore, RNN has multiple variant models on text classification. The Long Short-Term Memory Network (LSTM) is a variant of RNN that can solve long-term dependencies problems. LSTM updates the hidden state of each layer by removing or adding information to the cell state through the “gate” structure. Tang et al. [Tang, Qin and Liu (2015)] propose the gated recurrent network models to learn the semantics of sentences and their context relations. Firstly, the model learns text representation by CNN or LSTM. Then by using a gated recurrent neural network structure, the semantics of sentences and their relations are encoded into a document representation.

Lai et al. [Lai, Xu, Liu et al. (2015)] design a more complex network structure. They propose a Recurrent Convolution Neural Network (RCNN), which combine RNN with CNN and use bidirectional LSTM to obtain the context representation of each word. Firstly, all left-side and right-side contexts semantics are captured. The left-side context  $c_l(w_i)$  is the



left-side context of word  $w_i$ , which calculated by Eq. (6). The  $W^{(sl)}$  is a matrix that combined the semantic of the current word with the next word's left word, and the  $f$  function is a non-linear activation function. In the same way,  $c_r(w_i)$  is the right-side context of word  $w_i$ , which is calculated by Eq. (7). Then, this model used Eq. (8) to define  $w_i$ , which is the representation of word and it's connected by the  $c_l(w_i)$ ,  $e(w_i)$  and  $c_r(w_i)$ .

$$c_l(w_i) = f(W^{(l)}c_l(w_{i-1}) + W^{(sl)}e(w_{i-1})) \quad (6)$$

$$c_r(w_i) = f(W^{(r)}c_r(w_{i+1}) + W^{(sr)}e(w_{i+1})) \quad (7)$$

$$x_i = [c_l(w_i); e(w_i); c_r(w_i)] \quad (8)$$

Finally, this model uses max-pooling to extract the maximum features of the vector in order to obtain the information of the whole text. In this method, a new model is constructed flexibly by combining RNN and CNN, and the advantages of the two models are combined to improve the final text classification performance.

### 3.3 Attention mechanism-based

CNN and RNN can achieve good results in text classification tasks, but their shortcomings are not intuitive enough and their interpretability are not good. Recently, attention mechanisms is also proposed based on the aforementioned architectures. Attention mechanisms are common model of long-term memory mechanisms in the field of natural language processing. The biggest difference from CNN and RNN is that the attention mechanism-based can visually present the contribution of each word to the results. Du et al. [Du, Gui, Xu et al. (2017)] propose a novel attention mode, which combine RNN with CNN-based attention model. Firstly, this approach uses the convolution operation to obtain attention signals, that each attention signal represents the local semantic information of a word context. And then uses RNN to create text with attention signals. The higher the attention weight of a word, the more valuable the information it contains, and the more important it is in the process of text construction. Zhou et al. [Zhou, Shi, Tian et al. (2016)] also propose an Attention-based Bidirectional Long Short-Term Memory Networks (Att-BLSTM). One of the biggest advantages of this model is the combination of neural network attention mechanism and BLSTM to capture the most important semantic information in a sentence. Ma et al. [Ma, Yu, Tian et al. (2019)] propose the Global-Local Mutual Attention (GLMA) model. The model has two advantages, which can capture local semantic features and solve global long-term dependencies effectively. The mutual attention mechanism contains a local-guided global attention and a global-guided local attention. The local-guided global attention keeps the useful information of global long-term dependencies and the global-guided local attention extracts the most useful and informative local semantic features.

Yang et al. [Yang, Yang, Dyer et al. (2016)] also proposed Hierarchical Attention Network (HAN) models based on RNN, which can solve the problem of long-term dependence of text. This model adds an attention mechanism at the sentence level and a document level, which stands for different weights for highly important content. It can

alleviate the gradient disappearance problem when RNN captures the sequence information of the document. However, HANs are much slower to train because they utilize RNN. Gao et al. [Gao, Ramanathan and Tourassi (2018)] propose a Hierarchical Convolutional Attention Network (HCAN), an architecture based self-attention that can capture semantic relationships over long sequences like RNN, which also can achieve both fast like CNN and accurate performance in text classification task. Their experiment also shows that self-attention-based models may replace RNN-based models to reduce training time without sacrificing accuracy.

Taking the classification of news texts as an example, Cai et al. [Cai, Li, Wang et al. (2018)] use Sohu news data as the datasets, which are from 18 channels including domestic, international, sports, social and entertainment during June 2012 and July 2012. Firstly, the datasets are preprocessed, then build multiple models according to the training set. Finally predict the test set's output category and carry out an intensive calculation. The test results are shown in Tab. 1 [Cai, Li, Wang et al. (2018)].

**Table 1:** The classification results of the datasets

| Model | Precision | Training time     |
|-------|-----------|-------------------|
| CNN   | 0.8534    | 77 s/epoh×2 epoh  |
| RNN   | 0.8273    | 164 s/epoh×2 epoh |
| HAN   | 0.8456    | 179 s/epoh×2 epoh |

The text classification models based on deep learning in the above table are common methods. Compared with the previous research results, their performance and efficiency are improved. Although there are many model variants, a good and suitable model depends not only on the type of project task but also on the type and the size of the datasets.

#### 4 Conclusion

This paper mainly presents the text classification methods based on deep learning and several classic text classification network models. The analysis results show that training a good text classification model depends on not only the deep learning network models, but also the training data. Moreover, the network structure based on the attention mechanism can intuitively explain what is valuable information in the process of text classification, which is conducive to the improvement of system performance. With the rapid development of deep learning, the text classification methods based on deep learning will face more severe challenges. In future work, it is necessary to pay attention to universality, accuracy, training speed, prediction speed, interpretability and the difficulties of adjusting parameters.

**Funding Statement:** This work supported in part by the National Natural Science Foundation of China under Grant 61872134, in part by the Natural Science Foundation of Hunan Province under Grant 2018JJ2062, in part by Science and Technology Development Center of the Ministry of Education under Grant 2019J01020, and in part

by the 2011 Collaborative Innovative Center for Development and Utilization of Finance and Economics Big Data Property, Universities of Hunan Province.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- Bian, J.; Gao, B.; Liu, T. Y.** (2014): Knowledge-powered deep learning for word embedding. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 132-148.
- Cai, J.; Li, J.; Li, W.; Wang, J.** (2018): Deep learning model used in text classification. *International Computer Conference on Wavelet Active Media Technology and Information Processing*, pp. 123-126.
- Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y.** (2014): Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv, pp. 1412-3555.
- Conneau, A.; Schwenk, H.; Barrault, L.; Lecun, Y.** (2016): Very deep convolutional networks for text classification. arXiv preprint arXiv, pp. 1606-01781.
- Davis, C. A.; Fonseca, F. T.** (2007): Assessing the certainty of locations produced by an address geocoding system. *Geoinformatica*, vol. 11, no. 1, pp. 103-129.
- Du, J.; Gui, L.; Xu, R.; He, Y.** (2017): A convolutional attention model for text classification. *National CCF Conference on Natural Language Processing and Chinese Computing*, pp. 183-195.
- Gai, R. L.; Gao, F.; Duan, L. M.; Sun, X. H.; Li, H. Z.** (2014): Bidirectional maximal matching word segmentation algorithm with rules. *Advanced Materials Research*, vol. 926, pp. 3368-3372.
- Gao, S.; Ramanathan, A.; Tourassi, G.** (2018): Hierarchical convolutional attention networks for text classification. *Oak Ridge National Lab, Oak Ridge & TN*.
- Goldberg, Y.; Levy, O.** (2014): Word2vec explained: deriving Mikolov et al.'s negative-sampling word-embedding method. arXiv preprint arXiv, pp. 1402-3722.
- Gong, Z.; Yu, T.** (2010): Chinese web text classification system model based on naive Bayes. *International Conference on E-Product E-Service & E-Entertainment*, pp. 1-4.
- Hinton, G. E.** (1986): Learning distributed representations of concepts. *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, vol. 1, pp. 1-12.
- Hu, B.; Tang, B.; Chen, Q.; Kang, L.** (2016): A novel word embedding learning model using the dissociation between nouns and verbs. *Neurocomputing*, vol. 171, pp. 1108-1117.
- Jeffery, P, Richard, S, Christopher, M.** (2017): GloVe: global vectors for word representation. <https://nlp.stanford.edu/projects/glove/>.
- Johnson, R.; Zhang, T.** (2014): Effective use of word order for text categorization with convolutional neural networks. arXiv preprint arXiv, pp. 1412-1058.
- Johnson, D. E.; Oles, F. J.; Zhang, T.; Goetz, T.** (2002): A decision-tree-based symbolic rule induction system for text categorization. *IBM Systems Journal*, vol. 41, no. 3, pp. 428-437.

- Johnson, R.; Zhang, T.** (2017): Deep pyramid convolutional neural networks for text categorization. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 562-570.
- Kalchbrenner, N.; Grefenstette, E.; Blunsom, P.** (2014): A convolutional neural network for modeling sentences. arXiv preprint arXiv, pp. 1404-2188.
- Kesorn, K.; Poslad, S.** (2012): An enhanced bag-of-visual word vector space model to represent visual content in athletics images. *IEEE Transactions on Multimedia*, vol. 14, no. 1, pp. 211-222.
- Kim, Y.** (2014): Convolutional neural networks for sentence classification. arXiv preprint, pp. 1408-5882.
- Krizhevsky, A.; Sutskever, I.; Hinton, G. E.** (2012): Imagenet classification with deep convolutional neural networks. *Advances In Neural Information Processing Systems*, pp. 1097-1105.
- Lai, S.; Xu, L.; Liu, K.; Zhao, J.** (2015): Recurrent convolutional neural networks for text classification. *Twenty-ninth AAAI Conference on Artificial Intelligence*.
- Li, H.; Chen, P. H.** (2014): Improved backtracking-forward algorithm for maximum matching Chinese word segmentation. *Applied Mechanics and Materials*, vol. 536, pp. 403-406.
- Li, Z.; Shang, W.; Yan, M.** (2016): News text classification model based on topic model. *IEEE/ACIS International Conference on Computer & Information Science*, pp. 1-5.
- Liu, P.; Qiu, X.; Huang, X.** (2016): Recurrent neural network for text classification with multi-task learning. arXiv preprint arXiv, pp. 1605-05101.
- Ma, Q.; Yu, L.; Tian, S.; Chen, E.; Ng, W. W.** (2019): Global-local mutual attention model for text classification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 2127-2139.
- Miao, F.; Zhang, P.; Jin, L.; Wu, H.** (2018): Chinese news text classification based on machine learning algorithm. *Proceedings of International Conference on Intelligent Human-Machine Systems and Cybernetics*, vol. 2, pp. 48-51.
- Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; Dean, J.** (2013): Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, pp. 3111-3119.
- Pan, L.; Qin, J.; Chen, H.; Xiang, X.; Li, C. et al.** (2019): Image augmentation-based food recognition with convolutional neural networks. *Computers Materials & Continua*, vol. 59, no. 1, pp. 297-313.
- Pan, W.; Qin, J.; Xiang, X.; Wu, Y.; Tan, Y. et al.** (2019): A smart mobile diagnosis system for citrus diseases based on densely connected convolutional networks. *IEEE Access*, vol. 7, pp. 87534-87542.
- Rare, T.** (2017): Gensim: topic modeling for humans.  
<https://radimrehurek.com/gensim/index.html>.
- Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T.** (2012): Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine*.
- Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D. et al.** (2013): Recursive

deep models for semantic compositionality over a sentiment treebank. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631-1642.

**Socher, R.; Pennington, J.; Huang, E. H.; Ng, A. Y.; Manning, C. D.** (2011): Semi-supervised recursive autoencoders for predicting sentiment distributions. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 151-161.

**Soucy, P.; Mineau, G. W.** (2005): Beyond TFIDF weighting for text categorization in the vector space model. *IJCAI*, vol. 5, pp. 1130-1135.

**Tang, D.; Qin, B.; Wei, F.; Dong, L.; Liu, T. et al.** (2015): A joint segmentation and classification framework for sentence level sentiment classification. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 23, no. 11, pp. 1750-1761.

**Tang, D.; Qin, B.; Liu, T.** (2015): Document modeling with gated recurrent neural network for sentiment classification. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1422-1432.

**Tomas, M.** (2017): Word2Vec. <https://code.google.com/archive/p/word2vec/>.

**Wang, J.; Qin, J. H.; Xiang, X. Y.; Tan, Y.; Pan, N.** (2019): CAPTCHA recognition based on deep convolutional neural network. *Mathematical Biosciences and Engineering* vol. 16, no. 5, pp. 5851-5861.

**Wang, S.; Huang, M.; Deng, Z.** (2018): Densely connected CNN with multi-scale feature attention for text classification. *IJCAI*, pp. 4468-4474.

**Wei, S.; Guo, J.; Yu, Z.; Chen, P.; Xian, Y.** (2013): The instructional design of Chinese text classification based on SVM. *Chinese Control and Decision Conference*, pp. 5114-5117.

**Wu, H. C.; Luk, R. W. P.; Wong, K. F.; Kwok, K. L.** (2008): Interpreting TF-IDF term weights as making relevance decisions. *ACM Transactions on Information Systems*, vol. 26, no. 3, pp. 1-37.

**Xiao, Y.; Cho, K.** (2016): Efficient character-level document classification by combining convolution and recurrent layers. arXiv preprint arXiv, pp. 1602-00367.

**Xu, Q. N.; Liu, Z.** (2008): Automatic Chinese text classification based on NSVMDT-KNN. *IEEE International Conference on Fuzzy Systems & Knowledge Discovery*, vol. 2, pp. 410-414.

**Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A. et al.** (2016): Hierarchical attention networks for document classification. *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1480-1489.

**Yao, L.; Mao, C.; Luo, Y.** (2019): Graph convolutional networks for text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 7370-7377.

**Zhang, L.; Li, Y.; Meng, J.** (2006): Design of Chinese word segmentation system based on improved Chinese converse dictionary and reverse maximum matching algorithm. *International Conference on Web Information Systems Engineering*, pp. 171-181.

**Zhou, P.; Shi, W.; Tian, J.; Qi, Z.; Li, B. et al.** (2016): Attention-based bidirectional long short-term memory networks for relation classification. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, vol. 2, pp. 207-212.