Advances in Computer, Signals and Systems (2023)
Clausius Scientific Press, Canada

DOI: 10.23977/acss.2023.070213
ISSN 2371-8838 Vol. 7 Num. 2

Research on Text Classification Method Based on NLP

Mengnan Wang

The University of Queensland, Brisbane St Lucia, 4072, QLD

Keywords: Natural Language Processing, Text Classification, Deep Learning

Abstract: Natural language processing is a science that integrates computer knowledge, mathematical knowledge and linguistic knowledge, while text classification and recognition is considered an important research area and direction of natural language processing. In the context of the big data era, how to effectively classify text information in the face of a sea of text-based data is the focus of current research. This paper describes the theoretical knowledge of text classification concepts, text representation methods and text classifiers. Firstly, the basic concepts of text classification and the classification process are introduced. Then the model structures of convolutional and recurrent neural networks and their variants are introduced, followed by the structure and implementation principles of two classical word embedding models, Word2vec and BERT.

1. Introduction

Since the birth of the Internet, there has been an explosion of information on the web, which is most often carried in the form of text, which is screened, classified, clustered and sorted before it can be of value. The use of computers to process text is called natural language processing. The development of natural language processing technology has broadly gone through three stages, regularised processing, shallow semantic processing based on statistical machine learning, and deep semantic processing based on deep learning. Text classification has been a research hotspot in NLP and plays an important role in helping both humans and computers to analyse, process and make decisions about natural language. This approach has the advantage of saving manpower, facilitating information filtering and pushing, and customising personalised services, and is commonly used for tasks such as opinion analysis [1], spam recognition [2], and sentiment analysis [3].

In recent years, domestic research on text classification has developed more rapidly, and the direction of research has shifted more towards combining with deep learning. liu et al. [4] proposed three LSTM-based multi-task learning architectures in 2016, which can explore the information sharing mechanism between different tasks in a text sequence model, and the method performed very well in several experiments. Zhao et al. in 2018 [5] applied capsule networks to the problem of text classification, constructing both Capsule-A and Capsule-B architectures, and showed that a more comprehensive textual information could be learned using parallel convolutional filter windows. Later, Lin et al. [6] considered combining RNN and CNN to propose RCNN. Feng et al. in 2019 [7] applied capsule networks to Chinese text classification through the traditional word vector representation method, verified the superiority of capsule networks in processing long and short texts, and proved that capsule network models converge faster than CNN models. Zhao Qi et al. in 2020 [8] respectively used Both recurrent networks and capsule networks were used to extract global and local information to construct text feature matrices, and achieved good results in the task of processing text similarity. Lei et al. [9] added threshold attention to capturing important features in capsule networks, and the results were better than the original model.

Throughout the current state of research at home and abroad, text classification techniques have been developing at a rapid pace, from discrete to distributed representations, and from machine learning methods to deep learning methods. With the advent of the era of big data, the complexity of the classification task has become higher in the face of today's larger and more diverse forms of text data. The effectiveness of traditional classification models cannot meet the task requirements, and the establishment of a more accurate, stable and efficient model is the focus of exploration and attention.

2. Overview of text classification

2.1 Text classification definition

Text classification is the process of automatically classifying a document into one or more categories by learning the intrinsic features of the document and building a model of the relationship between the document and the category based on a pre-labelled dataset. In other words, given a pre-labeled text dataset $D = \{(D_1, y_1), (D_2, y_2), \cdots, (D_m, y_m)\}$, where Dm represents the mth text in the text dataset, $y_m \in \{0,1\}^k$ represents the text category label, and k represents the number of categories, text classification takes part of the data in the text dataset D as the training set, learns the potential relationship between text and categories through an algorithmic model, and establishes a mapping function f to realize the mapping from text to categories $f(D_m) \to y_m$.

2.2 Text classification process

Text classification and recognition is generally based on theoretical knowledge of machine learning, using a dataset with a pre-defined category in the training set, and mining the connection between the features of the dataset and its corresponding category labels to build a model for classification. The main processes and steps of automatic text classification and recognition are shown in Figure 1.

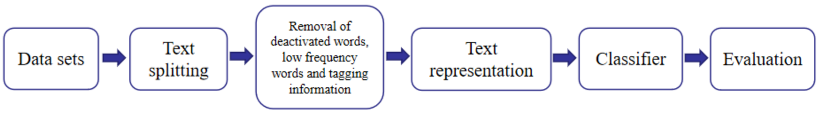
Data sets
Text splitting
Removal of deactivated words, low frequency words and tagging information
Text representation
Classifier
Evaluation

Figure 1: Flow chart of text classification

The text classification task consists of three key steps: text data format unification, vectorised representation, and feature extraction [10]: text pre-processing, text vectorisation table and text feature representation. The pre-processing of text is one of the most fundamental NLP tasks. Cluttered text data often has many irregular characters as well as web information, garbled codes, meaningless gibberish, etc. brought in from web crawling, which have a great impact on the performance of the classifier. Therefore, to improve the classification accuracy, we need to improve the quality of the data set in the pre-processing stage to reduce the time and space complexity for the classifier. The general pre-processing mainly goes through key steps such as text de-weighting, removing non-canonical symbols, word separation and deactivating words. There are three main word separation methods available: mechanical word separation, comprehension-based methods and string/dictionary matching-based methods [11].

2.2.1 Text separation

Chinese word separation is an important step in the text pre-processing process. For English text data, there are spaces between words, but Chinese words are connected, so the first task the computer has to deal with when processing Chinese text is to separate the Chinese text data into individual words. When slicing Chinese text, it is necessary to identify punctuation marks or specific words in a sentence, and then insert separators in the place of these punctuation marks or specific words to separate a sentence, the most common separator being spaces [12].

2.2.2 Text representation

Text representation is a method of processing text data in a way that allows the computer to correctly identify the content of the data, while preserving the original semantics of the text as much as possible. At present, the main text representation methods often used are One-Hot coding, Boolean model, Vector Space Model (VSM) model, Word Embedding representation, etc.

2.2.3 Classifier

The most efficient and feasible classification model is selected based on the category labels and corpus characteristics of the dataset, and the model parameters are adjusted using the training set to obtain the final classifier.

2.2.4 Text feature dimensionality reduction

At this stage, common automatic Chinese text classification systems usually treat words as feature terms, and the words treated as feature terms are called feature words. If all the words in a document are treated as feature terms, the feature vector will become a high-dimensional vector and the classification model may not be able to handle the high-dimensional data features, and the presence of interference features will affect the prediction results. Therefore, it is necessary to reduce the dimensionality of the features before inputting the data to make the computation easier and to improve the performance of the classification model. The text feature dimensionality reduction process is mainly divided into feature selection and feature weighting.

3. Deep learning models

The popularity of the Internet has led to an explosive growth of data, and in the face of the massive amount of data, machine learning models need to consume a lot of resources, and often fail to achieve satisfactory results, and are gradually eliminated. Since the deep neural network training method was proposed by Hinton [13] and others in 2006, deep learning technology has achieved rapid development, with unprecedented results in the field of image processing and natural language processing, and text classification methods based on deep learning have started to become the focus of scholars' research. Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) are the two classical models in deep learning and are often used as classifiers for text classification tasks.

3.1 CNN-based text classification models

Convolutional neural networks were first widely used in image processing and are good at capturing local features, and are also extremely competitive in the field of natural language processing. In traditional neural networks such as fully connected layer networks, the output of the upper layer is often connected to all the inputs of the lower layer, eventually forming a dense interaction structure, which is very likely to cause parameter explosion and lead to extremely slow convergence of the model, whereas in a convolutional neural network, each output node in the convolutional layer is only partially connected to the previous layer, forming a local sensory field. In contrast, in a convolutional neural network, each output node in the convolutional layer is only connected to some of the nodes in the previous layer, forming a local perceptual field, which reduces the number of model parameters and captures local features; secondly, the parameters are shared, i.e., different layers of the same model share the weight parameters, and no longer need to update the weights for each location, which greatly accelerates the model optimization process.

In the field of text classification, the sparse interaction property allows convolutional neural networks to automatically extract and combine N-gram level features on text to obtain multi-level local semantic information. When performing text classification, CNNs often consist of four layers as follows.

(1) Input layer: N × K word vector matrix, where N is the total number of words and K is the word vector dimension.

(2) Convolutional layer: convolution of the input matrix by multiple convolutional kernels of different sizes to obtain feature information of different granularity, forming the output of the convolutional layer.

(3) Pooling layer: the output of the convolutional layer is pooled to obtain a fixed-length representation of the text, removing unimportant features while further acquiring important information, effectively reducing the complexity of the model and speeding up the convergence process [14]; common pooling operations include Max Pooling, Average Pooling, Minimum Pooling, etc. Minimum pooling), etc.

(4) Output layer: Combine with fully connected layer and use Softmax function to complete the classification. Similar to the traditional CNN structure, the Text CNN also consists of four basic layers, namely the input layer, the convolutional layer, the pooling layer and the output layer, and its structure is shown in Figure 2. The input layer consists of two main channels, both using Word2vec pre-trained word vectors as the word embedding layer, but the training method is different. One of the channels directly initialises the non-occurring words at random, while the other channel continues the training of the word embedding layer. The Text CNN uses maximum pooling to stitch the pooled vectors together. Finally, a fully-connected layer and a softmax layer are combined to perform classification. As a pioneer, Text CNN uses the powerful local feature extraction ability of CNN to obtain the relationship of adjacent words in text, and further enriches the semantic features of text by means of multiple convolutional kernels, triggering a boom in the application of CNN to text classification tasks.

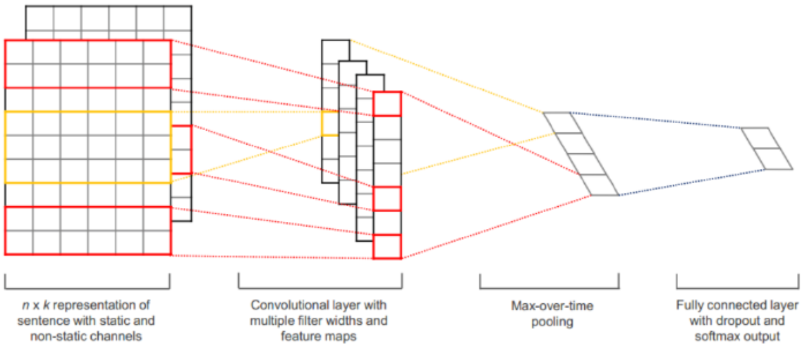
n x *k* representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Figure 2: Text CNN model structure

3.2 RNN-based text classification models

Recurrent Neural Network (RNN) has been developed due to the disadvantages of traditional feedforward neural networks, which cannot effectively use historical information. The unique chain structure of a recurrent neural network gives it the ability to process long sequences, taking into account both the current input and the output of the previous hidden layer during computation. Its standard structure is shown in Figure 3.

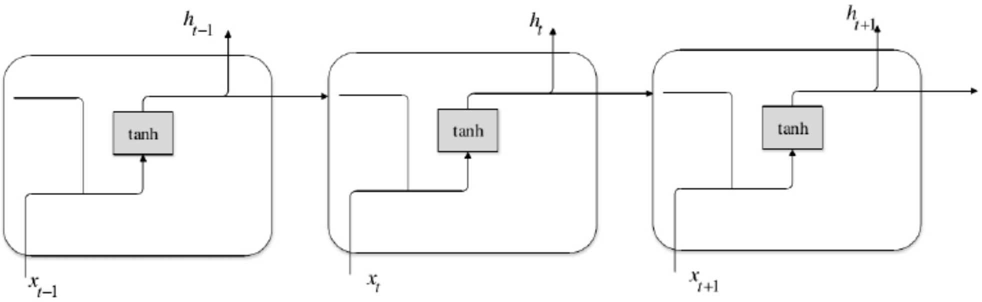

Figure 3: RNN model structure

In the above diagram xt and ht denote the input and output at time t respectively. From the RNN structure, it is clear that the output ht at time t is not only dependent on the input xt at time t, but also influenced by the output ht-1 at time t - 1. This structure gives the RNN a memory function, and the historical information also has an impact on the current information.

The memory feature of recurrent neural networks achieves great advantages when processing sequential data, but when the length of the sequence is too long, earlier input information has less and less influence on the output at later moments, leading to the problem of long-distance dependence. In addition, when the backpropagation algorithm updates the parameters, the error coefficients are also passed along the network hierarchy, and as the number of layers passed deepens, the gradient values increase or decrease exponentially, eventually leading to gradient explosion or gradient disappearance, limiting the learning capability of the RNN. To address the above problems, scholars have proposed two variants based on RNNs: long short-term memory network (LSTM) [15] and gated recurrent unit (GRU) [16], which can alleviate the gradient explosion and long-distance dependence problems of RNNs.

4. Deep learning text representation methods

Textual representation is a method used to digitally represent textual data. Traditional text representation methods suffer from high computational effort, lack of semantic information and serious waste of resources, and have been overtaken by deep learning-based word embedding models. This paper focuses on the commonly used Word2vec word embedding model and the pre-training model BERT.

4.1 Word2vec

Word2vec is one of the most commonly used word embedding models, introduced by Google, and consists of two main structures, CBOW and skip-gram, which are shown in Figure 4 below.

As can be seen from the Figure4, both CBOW and skip-gram are actually constituted as shallow neural networks, containing an input layer, a projection layer (implicit layer) and an output layer, where CBOW predicts the current word in context and skip-gram predicts the context in terms of the current word. In the figure wt is the current word, wt-1, wt-2, etc. are all words contained in the context, and the sliding window size is 2.

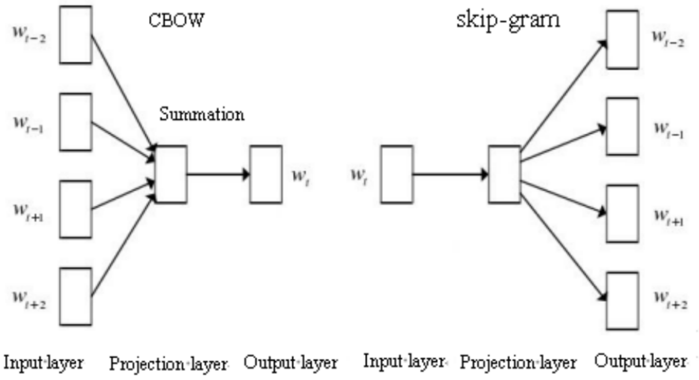

Figure 4: CBOW and skip-gram model structure

The input of both structures is represented by one-hot encoding, and the output is the current word or context generation probability. The training objective of the model is to maximise the overall generation probability of all words in the corpus. Either one of them can be used as the word embedding representation. To improve the training efficiency of the model, Word2vec provides two optimisation methods. The first is hierarchical softmax, which uses Huffman trees to encode the lexicon of the output layer, effectively reducing the time complexity of the algorithm; the second is negative sampling, which introduces negative samples during the training process of the model, updating only some of the node weights in each training, greatly reducing the computational effort of the model. Overall, Word2vec not only overcomes the drawbacks of traditional discrete representations, but also accelerates neural network training and opens up a new era of word embedding technology.

4.2 BERT

In response to the inability of static word embedding models such as Word2vec to solve the problem of multiple meanings of words, Google introduced the pre-trained language model BERT in 2018, sweeping 11 NLP tasks with the structure shown in Figure 5.

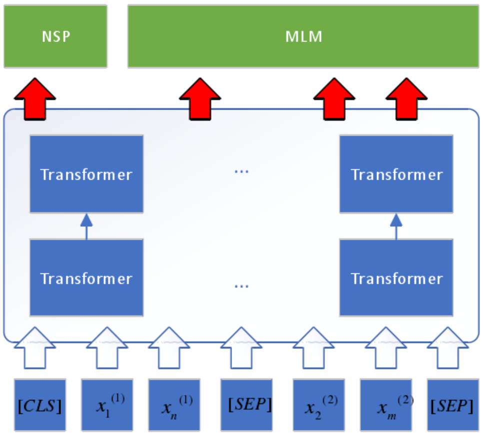

Figure 5: BERT model framework

The underlying architecture of BERT consists of a multi-layer Transformer, which has a special bi-directional structure that captures the contextual information of the words simultaneously and is highly capable of feature extraction. The input to the model usually consists of two pieces of text stitched together, which are modelled by the multi-layer Transformer and learned through two major pre-training tasks. The first is the Masked Language Model (MLM), which learns from the input text in a multilayer Transformer. (MLM), in which some of the input is masked at random with a certain probability in the input text sequence, and the model only predicts those masked inputs during training. Generally, a random percentage15 of words are selected for masking, and the masked words are replaced by special tags. The masked language model can enhance the scalability and error correction ability of the model, and ensure the richness and comprehensiveness of each input representation of the model. Secondly, the Next Sentence Prediction (NSP) task, which can be simply described as: given two sentences in the corpus, determining whether the second sentence should follow the first sentence in the corpus, ensures the model's ability to understand the context without overfitting. In summary, the BERT model incorporates ideas from classical models including Word2vec, GPT and Transformer, and after pre-training on a large scale corpus, it often results in highly applicable word vectors that can be fine-tuned to achieve good results when applied to specific downstream tasks, saving time and resources in retraining the word vectors. The BERT model is the most comprehensive of its kind and has been used as the basis for many subsequent improvements.

5. Conclusion

With the continuous development of Internet technology, the twenty-first century has become the era of big data, and the largest resource base comes from the Internet. How to quickly obtain the required data from the hundreds of millions of massive data has started to become a hot problem that needs to be solved, so text classification methods have started to become the focus of research and discussion among scholars. As a fundamental task of natural language processing, text classification methods have been continuously proposed and improved. This paper describes the theories and specific methods related to text classification, especially in recent years, new ideas such as word embedding models, pre-training models and attention mechanisms have been proposed, and text classification has made great progress. However, it still requires continuous in-depth research and exploration. In the field of natural language processing, there is still a very wide scope for research work such as the construction of deep learning models around when.

References

[1] Zhang Q, Gao T Z, Liu X Y, et al. Public environment emotion prediction model using LSTM network[J]. Sustainability, 2020, 12(4):1-16.

[2] Zou Y, Zhao T D, Qian W B. An improved model for spam user identification[P]. DEStech Transactions on Computer Science and Engineering, 2018.

[3] Nawangsari R P, Kusumaningrum R, Wibowo A. Word2Vec for indonesian sentiment analysis towards hotel reviews: an evaluation study [J]. Procedia Computer Science, 2019, 157: 360-366.

[4] Liu P, Qiu X, Huang X. Recurrent neural network for text classification with multi-task learning[J]. arXiv preprint arXiv:1605.05101, 2016.

[5] Yang M, Zhao W, Ye J, et al. Investigating capsule networks with dynamic routing for text classification [C]//Proceedings of the 2018 conference on empirical methods in natural language processing. 2018: 3110-3119.

[6] Lin R, Fu C, Mao C, et al. Academic News Text Classification Model Based on Attention Mechanism and RCNN [C]// Springer, Singapore. Springer, Singapore, 2018.

[7] Feng G., Zhang X., Liu S. Research on Chinese text classification based on CapsNet [J]. Data Analysis and Knowledge Discovery, 2019, 2(12).

[8] Zhao Q., Du Y. H., Lu T. L., et al. A text similarity analysis algorithm based on capsule-BiGRU [J]. Computer Engineering and Applications, 2020, 11(27):1-9.

[9] Lei K., Fu Q., Yang M., et al. Tag Recommendation by Text Classification with Attention-Based Capsule Network [J]. Neurocomputing, 2020.

[10] Sel L., Karci A., D Hanbay. Feature Selection for Text Classification Using Mutual Information[C]// 2019

International Artificial Intelligence and Data Processing Symposium (IDAP). IEEE, 2019.

[11] Asim M. N., Wasim M., Ali M. S., et al. Comparison of feature selection methods in text classification on highly skewed datasets[C]// International Conference on Latest Trends in Electrical Engineering & Computing Technologies. 2017:1-8.

[12] Wang Shuang. Research on automatic text classification based on machine learning [D]. University of Electronic Science and Technology, 2020.

[13] Devlin J, Chang M W, Lee K, et al. Bert: pre-training of deep bidirectional transformers for language understanding [J]. ar Xiv preprint ar Xiv:1810.04805, 2018.

[14] Liu Wanjun, Liang Xuejian, Qu Haicheng. Study on the learning performance of convolutional neural networks with different pooling models [J]. Chinese Journal of Graphical Graphics, 2016, 21(9):1178-1190.

[15] Hochreiter S, Schmidhuber J. Long short-term memory [J]. Neural computation, 1997, 9(8): 1735-1780.

[16] Chung J, Gulcehre C, Cho K H , et al. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling [J]. Eprint Arxiv, 2014.