Deep Learning Based Text Classification Methods

Zhaoguo Wang *

School of computer science, University of Glasgow, UK

* Corresponding Author Email: 2515112W@student.gla.ac.uk

Abstract. Text classification tasks are indispensable in natural language processing. With the development of Internet technology, the way people transmit information has changed from letters to the Internet. With the increase in the amount of information, manual data annotation is inefficient. After 2010, the emergence of deep learning methods has brought text classification into an epoch-making stage. ReNN-> MLP-> RNN-> CNN-> Attention -> Transformer-> GNN and other text classification methods are gradually being developed and known by everyone, which also shows that text classification is developing towards a text feature that does not rely on manually acquired text features and is directly learning from the text content and modelling. This paper will start with basic knowledge, first let everyone understand the nature, application and historical background of text classification, will briefly introduce shallow learning, then enter deep learning, Select the classic model from the six classification methods from ReNN to Transformer for brief analysis, briefly analyse the model from the principle of the model, what problems it is good at solving, problems or shortcomings that it is not good at solving. Finally, the paper describes the performance of these models on the dataset.

Keywords: text classification, natural language processing, deep learning.

1. Introduction

Text classification is the classification of text into categories according to certain rules, in other words, assigning a category to an object based on some pre-trained knowledge. According to the different predefined categories, there are two types of text classification: binary classification and multi-classification. Multi-classification can be achieved by binary classification. Text classification tasks can be divided into single-label and multi-label tasks depending on the number of categories to which the text belongs, because many texts can be associated with multiple categories at the same time. The application of text classification is very wide, such as subject classification: whether this document is about technology, sports or lifestyle. Spam detection: Whether this email is spam. Sentiment Analysis: Does the tweet express a negative or positive opinion, etc.

Nowadays, text classification is also divided into traditional learning and deep learning. If the training set and performance evaluation are not considered, we can split the traditional learning into two parts: feature engineering and classifier. In the field of Natural Language Processing, feature engineering is generally implemented in three steps: text pre-processing, feature extraction, and text representation. In traditional learning methods, classification algorithms are relatively fixed, and text representation models are particularly important. In the early development of text classification, Bag-of-words model (BoW model), TF-IDF model, LSA algorithm based on singular value decomposition (SVD) and topic model LDA are the commonly used text representation models in academia. It is also possible to classify traditional learning classifiers, such as rule-based decision tree model, probabilistic model-based naive Bayesian, geometric model-based SVM, statistical model-based KNN, etc, each of which has its own in the final analysis, the traditional learning model has a relatively simple structure and relies on manually acquired text features. Although the model parameters are relatively small, it can often show good results in complex tasks and has good domain adaptability. However, these methods are difficult to apply to sufficiently large data sets due to high labor costs. Furthermore, these models of text representation based on statistical features of word frequency ignores natural language contextual semantic and syntactic information, making the learning of complex semantic features in emotions challenging. Traditional methods are crucial for feature extraction, while deep learning methods can automatically extract features.

In the rest of the paper, the deep learning of text classification will be explained. The paper will choose the classic model from the 6 methods from recurrent neural network to graph network to share. Finally, the challenges and future development of text classification are summarized.

2. Deep neural learning for text classification

2.1. ReNN-based Methods

The emergence of the semi-supervised recursive autoencoder (Semi-Supervised RAE) [1] model has raised the recursive neural network (ReNN) to a level, and they combined the recurrent neural network with Autoencoder to form an unsupervised sentence conversion model. Sentences of indeterminate length can be converted into sentence vectors. Socher et al. applied it to the prediction of sentiment label distribution and outperformed the state-of-the-art methods at the time on commonly used datasets. However, the RAE results are relatively general in terms of accuracy, and the single word vector model is still insufficient for semantic modelling, especially the true semantics of deeper phrases cannot be learned well.

To improve the inability to capture long phrases and enable the model to understand language more deeply, Socher et al. proposed a novel recursive neural network model MV-RNN [2] for semantic compositionality. The main changes are assigning a vector and a matrix to each word, learning an input-specific, non-linear function for computing vector and matrix representations for sentences of arbitrary syntactic type. This method can well reflect the influence between adjacent nodes, but its disadvantage is that the parameter size depends on the size of the vocabulary and the computational efficiency is low.

The RNTN [3] model is mainly an improvement for the large number of MV-RNN parameters. When training in the SST data set, several results have reached the SOTA at that time. The essence of RNTN is to extract an indeterminate amount of information about each word and synthesize it in a matrix. It mainly uses a structure called a parse tree to get word vectors and processes them through the root and gets the class and score and continues to recurse until all the inputs are used up and the network has a parse tree containing each word. However, it has a high time complexity in constructing a text tree, and it is more complicated to express the relationship between documents in the tree structure.

2.2. MLP-based Methods

The researcher points out that the current processing of natural language processing tasks, such as sentiment analysis, intelligent question answering and other tasks, are all based on the vectorized representation of text, that is, how to construct an appropriate composition function. The methods can be mainly divided into two categories: unordered and syntactic, unordered training is fast, but because it is not sensitive to word order, the accuracy is not high; the syntactic method, although the model performance will be better, but the improved performance of the model does not match its cost. Therefore, the author tried to find a combination point, and DAN [4] was born. The first part of the model was summed and averaged, and a word dropout was also proposed, which is to randomly drop some tokens in the input to increase the robustness of the model. The second part adds the semantic information of the multi-layer nonlinear layers to extract the summed and averaged vectors. Each deep layer of the model is more abstract.

2.3. RNN-based Methods

The most fundamental problem with RNN is the short-term memory problem, such as using past chapters in a novel to infer the occurrence of the current chapter, which RNN cannot do. In order to solve this problem, LSTM [5] and GRU were born successively. Specifically, RNN cannot capture long-range dependence due to simple repetition of single neurons, while the repetition module in LSTM contains four special structures of forgetting gates, input gates, output gates and cell states, which allows LSTM to greatly improve the short-term memory problem. The GRU component is a

variant based on LSTM, which replaces RNN as the basic component in the sequence of seq2seq. The purpose is to retain the information of the long-term sequence and reduce the problem of gradient disappearance, The effect is that each hidden layer reduces several matrix multiplications, which speeds up the training speed. GRU combines the forgetting gate and input gate of LSTM into an update gate, which reduces the parameters and speeds up the convergence process while solving the problems of RNN long-term memory and gradient in back propagation [6]. But the two cannot operate in parallel, and the gradient will still vanish when processing a sequence that is too large.

To apply the tree network topology to LSTM, Tree-LSTM [7] is proposed. Similar to the standard LSTM structure, the difference is that the update of the gate vector and cell state in the Tree-LSTM unit depends on the state of all subunits related to it, In addition, compared to the single forget gate of standard LSTM, Tree-LSTM has multiple forget gates, corresponding to each subunit of the current unit, Therefore, Tree-LSTM can selectively obtain information from sub-nodes, for example, in sentiment analysis tasks to save the information of sub-nodes with richer semantic information. Two different organizations, Child-Sum Tree-LSTM(Dependency Tree-LSTM) and N-ary Tree-LSTM(Constituency Tree-LSTM)。The disadvantage of Child-Sum is that the feature extraction will lose the position information of the child nodes (the feature vector of the child node is added, and the position cannot be determined). The disadvantage of N-ary is that when the number of child nodes is uncertain, it is difficult to declare a certain number of weight matrices for child nodes.

RNN is proficient in capturing temporal features, but has no competitive advantage in handling spatial features. Academics have tried to introduce convolutional neural networks (CNN), which are commonly used in computer vision, into the field of natural language processing, and have also achieved competitive results.

2.4. CNN-based Methods

When CNN first came out, it successfully solved two problems in the image: the large amount of data for image processing and the difficulty in retaining the original features of the image in the process of digitization. But in 2014, Kim successfully proposed a convolutional neural network for text classification - TextCNN [8]. The process of TextCNN is to first perform embedding of text words to obtain word vectors, and then pass the word vectors through a layer of convolution, a layer of max -pooling, Finally, the external SoftMax will be output for classification. The advantage of TextCNN is that the model is simple, the training speed is fast, and the effect is good. However, the interpretability of the model is not strong. When tuning the model, it is difficult to adjust specific features according to the training results, which is also its shortcoming.

In 2014, TextCNN has achieved competitive results in text classification tasks before. However, in TextCNN, the Max-Pooling pooling method selected by each convolution kernel can only select one n-gram information. These n-gram information is continuous without interval, similar to the continuous word bag, and the extracted features are too limited. In order to solve the above problems, Kalchbrenner et al. proposed a new model - DCNN [9], which is dedicated to "accurately expressing the semantics of sentences as the core of language understanding". DCNN mainly includes one-dim wide convolutional layer, Dynamic k-max pooling layer, Folding layer and fully connected layer. The pooling layer is also the most important part. As the name suggests, k-max pooling is to select the largest number of top k, ‘Dynamic‘is to dynamically select the k value through the formula according to the network structure and the preset top k. Not only that, DCNN does not require any prior information input, nor does it need to construct very complex artificial features.

Prior to this, many deep learning-based models used high-level units to model text or language, such as word, phrase, sentence level, or to analyse semantic and grammatical structures, but this paper proposes a more fine-grained model. —CharCNN [10] performs text classification at the character level. And proposed two scales of neural network - Large and Small, the main difference is the number of num_feature and hidden_unit is different. Both consist of 6 convolutional layers and 3 fully connected layers, a total of 9 neural networks. Besides, two dropout layers are added between the three fully connected layers for model regularization. CharCNN is based on a smaller-grained text

classification model, which can be used across languages because it does not consider the intrinsic meaning, semantics, and syntax of words. And it is more suitable for less standard texts, such as user-generated data. For example, user reviews, etc., are not very competitive in regular texts such as news.

2.5. Attention -based Methods

The text classification algorithm mentioned above is basically sentence-level classification, using long text and chapter-level, although it is also possible to achieve its classification, but the speed and accuracy will decrease.So some researchers proposed a hierarchical attention [11] classification framework, that is, the model Hierarchical Attention,The researchers said that when classifying documents/longer texts, it is not enough to only pay attention to the word granularity, and it is necessary to learn attention for each sentence (short sentence).Different sentences also need to be assigned different weights, and the words in each sentence are also assigned different weights. The specific process is to first encode each sentence with BiGRU+Att to obtain the sentence vector, and then use BiGRU+Att to obtain the doc-level representation of the sentence vector, and then classify. The researchers used six datasets, the smallest dataset also contains 33w articles, and the largest dataset has ten times as many as the smallest dataset. The researchers used three combinations, HN-AVE and HN-MAX change the method of generating feature vectors of each layer from the weighted summation of Attention to directly doing Average Pooling and Max Pooling, but the effect of Attention is still better than the former two.

Word embedding is the most classical model of word representation in NLP. However, word2vec is essentially a static model. The expression of each word in this model is fixed and does not capture contextual information dynamically. To solve this problem, ELMO dynamically acquires textual contextual information using bidirectional LSTM. Essentially, ELMo [12] is a model based on a pre-trained language model that dynamically characterizes Word embedding according to the current context. Researchers call this a domain transfer. In this way, the word embedding of the word in the current context can be obtained by using the context information of our training data. The researchers tested SQuAD with a baseline model that is an improved version of the bidirectional attention flow model. After adding ELMo to the baseline model, we get very good data. But ELMo is essentially an RNN, which is weak in feature extraction and training time.

2.6. Transformer -based Methods

BERT [13] is a pre-training model proposed by Google AI Research in October 2018. BERT is two-way fine-tuning. Compared to feature-based pre-trained models, fine-tuning makes less architectural changes to pre-trained models. The input of BERT consists of token_embedding , segment_embedding and position_embedding. BERT obtains generic text representations in the pre-training phase by simultaneously training two pre-training tasks of masked language model and next sentence prediction. Although BERT has its own bidirectional function, BERT consumes a lot of hardware resources.

Reasoning about relationships between multiple text words is involved in many NLP tasks. For example, in extractive question answering, the results are not satisfactory. So Mander et al. made a series of improvements to BERT and proposed SpanBERT [14]. They modified BERT's approach: Google BERT extracted 10 different masks for each sequence during data processing, and the reproduced BERT used a different mask for each epoch. And the setting of generating short sentences according to 10% is removed, and each sample is guaranteed to be 512 in length unless it reaches the end of the file. After that, three major improvements were made based on BERT: instead of using a random mask method, a certain continuous token was used to mask; a task of predicting the mask by the boundary (Span Boundary Objective) was added; Through experiments, it is found that it is better to abandon the next sentence prediction task and train directly with long text.

In NLP tasks, a good pre-trained model can improve the performance of the model. The current SOTA model has millions or billions of parameters. If you want to expand the model size, you will encounter the limitations of these computer memory, and the training speed will be limited. There are

currently two solutions: model parallelization, good memory management mechanism. But both methods have communication overhead. Therefore, the paper designs an A lite BERT (ALBERT), which uses fewer parameters than BERT. ALBERT [15] overcomes the main obstacle to scaling pre-trained models by introducing two parameter reduction techniques, factorized embedding parameterization and Cross-layer parameter sharing. And the researchers also introduced an Inter-sentence coherence loss for sentence coherence modelling to solve the problem of inefficient next sentence prediction loss in the original BERT.

3. Result and discussion

Table.1. Performance of the partial models presented in this paper on different training sets. The data are all from the paper, and the hardware configuration in different periods may have some influence on the data [16].

Model	Sentiment						News		Topic
	MR	SST-2	IMDB	Yelp.P	Yelp.F	Amz.F	20NG	AG	DBpedia
RAE	77.30	82.40	-	-	-	-	-	-	-
MV-RNN	77.70	82.90	-	-	-	-	-	-	-
RNTN	75.90	85.40	-	-	-	-	-	-	-
DAN	-	86.30	89.40	-	-	-	-	-	-
LSTM	77.40	84.90	-	-	-	-	-	-	-
Tree-LSTM	-	88.00	-	-	-	-	-	-	-
TextCNN	81.50	88.10	-	-	-	-	-	-	-
DCNN	-	86.80	89.40	-	-	-	-	-	-
CharCNN	-	-	-	95.12	62.05	-	-	90.49	98.45
HAN	-	-	49.40	-	-	63.60	-	-	-
BERT-base	-	93.50	95.63	98.08	70.58	61.60	-	-	91.00
BERT-large	-	94.90	95.79	98.19	71.38	62.20	-	-	91.70

Table 1 presents the performance of the partial models. With the development of NLP, text classification has also evolved from shallow learning to deep learning. From the initial recurrent neural network to LSTM to the attention mechanism and finally to the epoch-making BERT and the emergence of graph neural networks, this undoubtedly brings the development of text classification tasks to a new height, but the new height also faces some new problems and challenges.

The first is the interpretability of deep learning models. With the improvement of people's requirements for text classification tasks, deep learning models are also solving these increasingly complex text tasks, but for most deep learning models are not interpretable, for example, what knowledge is learned by the pre-training model or what features are retained when fine-tuning the training data and classification tasks, we cannot intuitively understand. This makes the improvement and optimization of the model lose clear guidelines, and greatly deepens the difficulty of parameter adjustment for researchers.

The second is a simpler and more efficient model. Models such as BERT in the past two years have several million or more parameters, which is undoubtedly a great challenge in terms of training speed and computer memory. Simplifying models and making tradeoffs between computer memory and performance is a subject of immediate research.

Zero-shot learning and Few-shot learning. The former refers to the ability of the model to classify categories it has never seen before, giving machines the ability to reason and achieve true intelligence. The latter refers to giving the model a small number of samples of the class to predict, and then having the model predict that class by looking at other samples of that class. However, a large amount of labelled data is still required for most models. Appropriately reducing the amount of data so that the machine has a certain reasoning ability is also the focus of current research.

4. Conclusions

This paper introduces the basics of text classification, from the classification process and history of traditional models to 6 major deep learning methods, in which classic models are selected for analysis. So far, deep learning models have dominated, it has successfully helped people solve one problem after another in text classification tasks. In recent years, Most researchers are still exploring attention mechanisms and Transformers, trying to develop better models.in the future ， robustness and graph neural networks have also become the focus of researchers, which also shows that we are developing towards a higher and more comprehensive field for text classification. However, this paper does not introduce the relevant model of graph structure, because the graph structure is still in the beginning stage of research, so this paper does not analyze its model.

References

[1] Socher R et al. Semi-supervised recursive autoencoders for predicting sentiment distributions [C]. In Proceedings of the 2011 conference on empirical methods in natural language processing (pp. 151-161), 2011

[2] Socher R et al. Semantic compositionality through recursive matrix-vector spaces [C]. In Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning (pp. 1201-1211), 2012.

[3] Socher R et al. Recursive deep models for semantic compositionality over a sentiment treebank [C]. In Proceedings of the 2013 conference on empirical methods in natural language processing (pp. 1631-1642), 2013.

[4] Iyyer M et al. Deep unordered composition rivals syntactic methods for text classification [C]. In Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers) (pp. 1681-1691), 2015.

[5] Shi X et al. Convolutional LSTM network: A machine learning approach for precipitation nowcasting [J]. Advances in neural information processing systems, 28, 2015

[6] Cho K et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation [R]. arXiv preprint arXiv:1406.1078, 2014.

[7] Tai K S et al. Improved semantic representations from tree-structured long short-term memory networks [R]. arXiv preprint arXiv:1503.00075, 2015

[8] Kim Y. Convolutional neural networks for sentence classification [R], in Proc. EMNLP, 2014, pp. 1746–1751, 2014.

[9] Kalchbrenner N et al. A convolutional neural network for modelling sentences [R]. arXiv preprint arXiv:1404.2188, 2014.

[10] Zhang, X et al. Character-level convolutional networks for text classification [J]. Advances in neural information processing systems, 28, 2015.

[11] Yang Z et al. Hierarchical attention networks for document classification [R]. In Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies (pp. 1480-1489), 2016.

[12] M. E. P et al. Deep contextualized word representations [C], in Proc. NAACL, 2018, pp. 2227–2237, 2018

[13] Devlin J et al. Bert: Pre-training of deep bidirectional transformers for language understanding [R]. arXiv preprint arXiv:1810.04805, 2018

[14] Joshi M et al. Spanbert: Improving pre-training by representing and predicting spans [J]. Transactions of the Association for Computational Linguistics, 8, 64-77, 2020.

[15] Lan Z et al. ALBERT: A lite BERT for self-supervised learning of language representations in Proc. [C] ICLR, 2020.

[16] Li Q et al. A Survey on Text Classification: From Traditional to Deep Learning [J]. ACM Transactions on Intelligent Systems and Technology (TIST), 13(2), 1-41, 2022.