Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/5/20230612

Performance analysis of sentiment classification based on deep learning methods

Yuting Zhao

Trinity CofE High School, Higher Cambridge Street , Manchester, M15 6HP UK

y17zhao_y@school.trinityhigh.com

Abstract. When it comes to natural language processing, the textual differentiation task is one of the classical and important research problems. Recently, the deep learning model has increasingly become one of the main methods to solve text classification problems. Common deep learning text classification models are convolutional neural networks (CNN), recurrent neural networks (RNN), the BERT model. For comparing the manifestation of various deep learning models in textual differentiation tasks horizontally, the thesis tests the classification accuracy of different deep learning models under the same experimental configuration. The experimental results show that using pre-trained word vectors helps to improve the classification accuracy of deep learning models. In addition, the reasonable design of more complex and larger deep learning models is helpful in enhancing the study capability of the specific model on text data. The experimental results indicate that the text classification model using pre-trained word vectors could gain higher accuracy than the model without pre-trained word vectors. In addition, in the comparison experiment of feedforward neural network (FNN), CNN, RNN and BERT model, BERT model performs best, and the text classification accuracy reaches 0.9232. Compared with a 1-layer FNN, the accuracy rate is increased by about 16%.

Keywords: deep learning model, text data, BERT model

1. Introduction

Natural language processing has already being one vital study dimension in artificial intelligence field. It primarily researches all kinds of theories and approaches for valid communication between people and computers when both of them use natural language. As IT develops fast, natural language processing appears in our daily life of human beings, like machine translation, public opinion monitoring and the like [1]. Textual classification is gradually becoming classical and vital due to the result of fast speed of development of textual data. Text classification completes the summarization of text data by extracting features from massive complex text data. Text classification tasks can be applied in a variety of scenarios, such as spam identification, sentiment analysis, literature classification, etc. Accurate text classification through algorithms has great economic and social value.

DOI: 10.54254/2755-2721/5/20230612

Recently, making full use of deep learning models to solve problems in text differentiation has become popular, among which CNN and RNN models are frequently applied in the text differentiation tasks. Kim et al. first introduced CNN into text classification tasks, completing sentence representation by concatenating word vectors [2]. Yin et al. suggested using various types of pre-trained term vectors and adopt variable size filters to accomplish term characteristic extraction of granularities in different patterns [3]. Peng et al. proposed a Bi-LSTM architecture model based on the attention mechanism, which captured important semantic information in sentences through the attention mechanism [4]. Li et al. put forward that the SAMF-Bi-LSTM model, solving the problem of emotional information loss in sentiment classification tasks effectively [5].

This paper mainly studies the performance differences of different text classification models on the IMDB dataset. Specifically, for this problem, the outcome of this thesis primarily consists of the following two points: 1) Using two different RNN models: RNN models using the word2vec algorithm with pre-trained word vectors and RNN models without pre-trained word vectors. Under the same experimental configuration, the accuracy of the two models in the text classification task is compared. 2) Using multiple deep learning models with different architectures: feedforward neural networks, CNN, LSTM, Bi-LSTM, and BERT. The accuracy of deep learning models with different structures in text classification tasks is compared under the same experimental configuration as much as possible. The test results show the text classification model by using pre-trained word vectors can achieve higher accuracy than the model without pre-trained word embedding. In addition, in the comparison experiment of the feedforward neural network, convolutional neural type, recurrent neural type and BERT model, the BERT model performs best, and the text classification accuracy reaches 0.9232. Compared with a 1-layer feedforward neural network, the accuracy rate is increased by about 16%.

2. Methods

This paper studies the performance of all kinds of deep learning models existing in text differentiation assignments. Those models studied in this paper mainly include CNN, RNN, LSTM, and BERT. This chapter introduced the definition of the text classification task and the four deep learning models included in the study.

2.1. Text classification tasks

The text classification task mainly includes six processes: 1) text preprocessing; 2) text representation; 3) feature extraction; 4) feature dimension reduction; 5) classification model; 6) result evaluation.

Text preprocessing refers to the segmentation of original text data, removal of stop words, etc., settlement of the storage space and improvement of the processing efficiency of subsequent algorithms.

Text representation refers to the representation of text content into feature vectors, which is convenient for computers to process text data. Common text representation methods include One-hot, TF-IDF and word embedding [6-7].

Feature dimensionality decrease refers to the decrease of the characteristic space extracted from text data, usually for avoiding the problems of too complex model calculation and excessive memory consumption caused by the high dimension of feature vector. Scholars will adopt dimensionality decrease approaches like Linear Discriminant Analysis, Non-negative Matrix Factorization, and Principal Component Analysis, which restricts text features to a vector space of a certain dimension [8-10].

Classification model refers to the use of appropriate algorithm model to complete text classification. Common text classification models are mainly separated into deep learning (DL) models and machine learning models. Machine learning models include naive Bayes, k-nearest neighbor algorithm, SVM algorithm, etc, while deep learning models include CNN, RNN, LSTM, etc. This paper pays much attention to the performance of DL models.

DOI: 10.54254/2755-2721/5/20230612

Evaluation results could be applied into evaluating the representation of the model through a series of indicators. Common indicators include F1-score, accuracy, precision, etc. In this paper, accuracy is mainly used as the main evaluation index.

2.2. Convolutional nNeural Networks, CNN

Kim et al. first propose to use textCNN to solve text classification tasks, and the textCNN model is mainly composed of four layers: input, convolutional, pooling and fully connected [2]. The input data is a matrix in $n \cdot k$ size, wherein parameter n is defined as the word number, and the other parameter k is defined as the dimension of word vector. Sentences can be expressed as follows, wherein $\oplus$ indicates splicing operation.

$$x_{1:n} = x_1 \oplus x_2 \oplus \cdots \oplus x_n \tag{1}$$

The convolution layer uses one-dimensional convolution, and each slide of the convolution kernel extracts a local feature. The weights are shared among neurons. If the existing discrete functions $g(x, y)$ and $f(x, y)$ are set, the definition of convolution is indicated as below:

$$f(x, y) \cdot g(x, y) = \sum_u^{\infty} \sum_v^{\infty} f(u, v)\ g(x - u, y - v) \tag{2}$$

The convolutional layer uses ReLu as a nonlinear activation function, whose formula is as follows:

$$\sigma(x) = \mathrm{m}\ \{0, x\} = \begin{Bmatrix} x, & x \geq 0 \\ 0, & x \leq 0 \end{Bmatrix} \tag{3}$$

The third layer even refines the characteristics extracted by the second layer to highlight important features and to reduce the feature dimension. Frequently applied pooling approaches consist of two types of pooling, that are, average pooling and max pooling, etc .

That last layer refers to which is a entirely connected layer, and classified result is output by the function $sof\ tmax$.

2.3. Recurrent neural network

RNN refers to a Deep Learning model that mainly often processed rank data, which can solve the problem that ordinary feedforward neural networks cannot transmit and utilize historical data [6]. Suppose the word embedding representation of an existing text sequence be as: $x_1, \ldots, x_i, \ldots, x_i$. The RNN model can update the covered condition of the moment through x_i input at time *i* and the output condition h_{i-1} at the previous time. The expression of this hidden state is shown below:

$$h_t = g(Uh_{t-1} + Vx_t + b_h) \tag{4}$$

$$o_t = g(Wh_t + b_o) \tag{5}$$

Specifically, g refers to the activation function, and W, V and U mean the parameter matrixes. The argument message of every neuron is used by taking the output of the final moment as the feature vector when it comes to the text sequence. The feature vector of the text sequence is input into the last layer, and the final classification result is outputted through the function $sof\ tmax$.

2.3.1. LSTM. Since the input of each neuron in the RNN model contains the output of the previous unit, "long distance dependence problem" will appear in the recurrent neural network when the sequence data grows to a certain length. The input information at earlier times will have less and less influence on the whole sequence data, and in the process of backpropagation, the gradient at every time node will be passed to the front neuron, so it seems simple to have the obstacle of gradient missing or gradient blast. In order to solve the above problems of recurrent neural networks in processing sequence data, researchers have proposed a variant of RNN: LSTM [11]. The gating mechanism is newly added to LSTM, which controls the information flow through output gate, forgetting gate and input gate to resolve their problems of gradient missing and gradient blast. The specific structure of the gating mechanism is as follows:

(1) Forget gate: it is mainly applied for controlling the retention of message from the previous moment. The forget gate combines the previous moment h_{t-1}as output with at the present time x_t

DOI: 10.54254/2755-2721/5/20230612

as the input at the present time, and obtains the probability that information is forgotten at the front moment through the activation function sigmoid. The formula is as follows:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{6}$$

(2) Input gate: it mainly controls the information input of the neuron, including the processing of the present time data and the memory of the present time data. Current time data x_t obtains the output result i_t through sigmoid and obtains the output result $\hat{C}_t$ through activation function tanh. Specifically, the updated state C_t of the memory cell at the current time can be seen in the following formula:

$$\begin{gathered} i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \hat{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_C) \\ C_t = f_t^* C_{t-1} + i_t * \hat{C}_t \end{gathered} \tag{7,8,9}$$

(3) Output Gate: The Output Gate is mostly used for controlling the cell to discard some unnecessary memory information at the current time, and the specific formula is shown below:

$$\begin{gathered} o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o) \\ h_t = o_t^* \tanh(C_t) \end{gathered} \tag{10,11}$$

2.3.2. BERT. BERT (bi-directional encoder representations from transformers) model is an open-sourcing pattern proposed by Google in 2018, and it adopts masked language model to pre-train the bidirectional Transformer to generate deep bidirectional linguistic representations [12, 13]. Specifically, BERT represents each word in the input pattern as a semantic vector, then inputs it into multiple Transformer encoders for training, and finally obtains the trained term vector. The Transformer encoder used in BERT converts the semantic vector of each word into an enhanced semantic vector of the same length by using key operations such as multi-head attention mechanism, self-attention one, left connection one, layer normalization, and linear transformation one so as to complete the training of the word vector. After training in advance, BERT only needs to an extra layer for output so as to realize state-of-the-art property in different downstream assignments.

3. Experimental results and analysis

This part mainly introduces the experimental process related to the research content of this paper. The experiment mainly uses the IMDB dataset to finish the text differentiation job. The IMDB data set consists of 50000 comments and each comment obviously contains sentiments. Positive sentiment accounts for 50% of comments, and negative sentiment also accounts for 50% of comments. There are two different parts come from that the data set was separated into, tensile part and test part, and each part had 25000 samples. Each record is contained in a tab-separated value (TSV) gzip file in UTF-8 character set.

The experiment is divided into two main parts. In the first part, two different RNN models have used: the RNN model with word2vec algorithm pre-trained word embedding and the RNN model without pre-trained ones to accomplish the text differentiation job [14]. The second part uses multiple deep learning models with different structures: feedforward neural network, CNN, LSTM, Bi-LSTM, and BERT to compare the accuracy of deep learning models with different structures in text classification tasks under the same experimental configuration as much as possible.

In first part, this paper requires comparing the result of two RNN models with and without using the pre-trained word embedding. To finish this task, this paper keeps all parameters of these two models consistent, and this paper uses Word2Vec as word embedding in one of the models. Details of models can be seen in table 1.

DOI: 10.54254/2755-2721/5/20230612

Table 1. All Parameters of these two models.

MODE TYPE	RNN	RNN+WORD2VEC
EMBEDDING	None	Word2Vec
EMBEDDING DIM	100	100
BATCH SIZE	64	64
HIDDEN DIM	256	256
OUTPUT DIM	1	1
OPTIMIZER	SGD	SGD
LEARNING RATE	1e-3	1e-3
LOSS FUNCTION	BCE With Logits Loss	BCE With Logits Loss
EPOCH	100	100
TRAINING LOSS	0.693	0.693
TRAINING ACCURACY	**50.39%**	**50.52%**
TESTING LOSS	0.707	0.689
TESTING ACCURACY	**49.10%**	**56.32%**

As can see from table 1, the pre-trained word embedding does not give a big boost to our outcome when training the model. However, the testing accuracy has improved significantly by 8%, which is reasonable but not as good as expected. This paper thinks there are various reasons, and for example, this word embedding does not fit the model. Maybe this paper can try to use another embedding, such as Glove, and it may give us a better result. Another possible reason is that this paper hasn't updated the parameters of the embedding layer when training. It will better match our data if we do. As to the second part, there are multiple models this paper needs to compare, as shown in Table 2.

Table 2. The Results Of Different Patterns.

Model	Optimizer	Learning rate	Training Loss	Training Accuracy	Testing Loss	Testing Accuracy
1-layer	Adam	1.00E-03	0.574	72.06%	0.578	**70.75%**
2-layer	Adam	1.00E-03	0.497	76.10%	0.502	**75.67%**
3-layer	Adam	1.00E-03	0.491	76.48%	0.499	**75.83%**
CNN	Adam	1.00E-03	0.335	85.76%	0.433	**80.32%**
LSTM	Adam	1.00E-03	0.299	87.56%	0.355	**84.70%**
bi-LSTM	Adam	1.00E-03	0.195	92.22%	0.317	**86.71%**
BERT	Adam	2.00E-05	0.155	94.16%	0.318	**92.32%**

The first three models are all simple feedforward neural networks. When the 1-layer one has 70.75% accuracy in testing data, the 2-layer one is 5% higher than it. Their reason is that a deeper model means a more complex structure as well as a much more powerful feature. However, the 3-layer model doesn't show an obvious improvement over the 2-layer one, and it may be because the same parameters in the 2-layer model do not fit with the 3-layer model. A fine turning in hyper-parameters may help.

Usually, the CNN models are more frequently seen in image processing tasks, but it is also able to use to solve NLP problems. Compared with the simple feedforward neural networks, it doesn't process data word by word, and its filter can find the meaning expressed by the phrases, which can produce a better result.

The LSTM and Bi-LSTM models are both a variant of the RNN model. The RNN model retains information about all processed words for the next word and is better able to show the relationship

DOI: 10.54254/2755-2721/5/20230612

between a single word and the rest of the sentence. The characteristics of LSTM models solve the problem of gradient vanishing in traditional RNN models to some extent, which brings a significant performance improvement. When the current word is processed in unidirectional LSTM, there is no information about unprocessed words in the sentences, and the bi-directional LSTM solves this problem well. It processes the sentence both back and forth simultaneously.

Finally, BERT is quite a big model. It contains the idea of attention and uses the technique of transformer, and it is one of the best models for the NLP task nowadays. It contains almost all the advantages of the previous models, that's why it brings us the best result ever in this task. Figure 1 further illustrates the performance differences between different models.

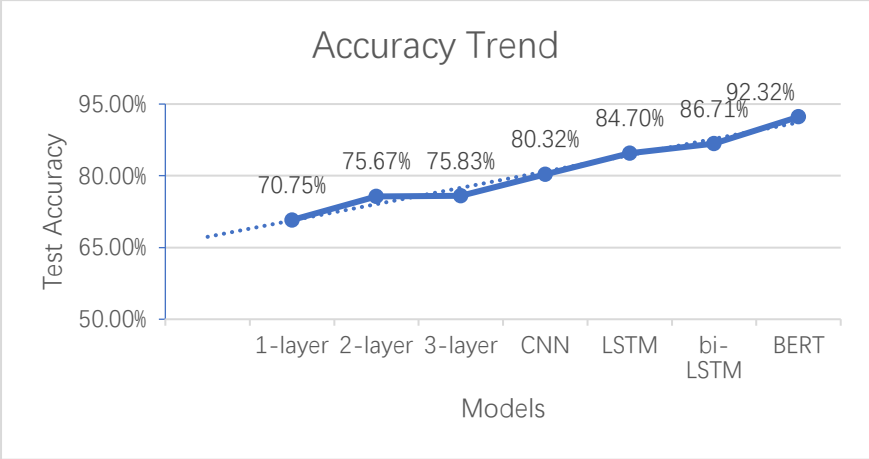

Figure 1. The performance differences between different models.

4. Conclusion

In this paper, we study and compare the precision of different deep learning models of IDMB in classifying important texts. Experimental results indicate that the text classification model using the word carrier before training is more accurate without the word carrier. In addition, in the comparison experiments of neural network feeding, CNN, RNN and Bert model, the best performance and text classification accuracy of Bert model are up to 0.9232. When it compared with single-layer Feedforward neural network, the accuracy ratio is 16%.

In summary, in text classification tasks, using appropriate algorithms to pre-train word vectors and improving the utilization rate of the model on text data can help to enhance the precision of text classification jobs. However, how to choose the appropriate word vector embedding algorithm, how to further improve the differentiating precision of the model still have room for further improvement and how to integrate the merits of deep text classification patterns to improve the differentiating precision could be taken as the tasks of future research being continued to be promoted.

References

[1] Muhammad A . A Methodology for Comparison of User Reviews with Rating of Android Apps using Sentiment Analysis[D]. SWUST (Southwest University of Science and Technology).

[2] Kim Y . Convolutional Neural Networks for Sentence Classification[J]. Eprint Arxiv, 2014.

[3] Yin W , H Schütze. Multichannel Variable-Size Convolution for Sentence Classification[C]// Proceedings of the Nineteenth Conference on Computational Natural Language Learning. 2015.

[4] Peng Z , Wei S , Tian J , et al. Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification[C]// Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2016.

[5] Li W , Qi F , Tang M , et al. Bidirectional LSTM with self-attention mechanism and multi-channel features for sentiment classification[J]. Neurocomputing, 2020, 387:63-77.

DOI: 10.54254/2755-2721/5/20230612

[6] Zaremba W , Sutskever I , Vinyals O . Recurrent Neural Network Regularization[J]. Eprint Arxiv, 2014.
[7] Ruch P , Baud R , Geissbuehler A . Evaluating and reducing the effect of data corruption when applying bag of words approaches to medical records[J]. International Journal of Medical Informatics, 2002, 67(1/3):75-83.
[8] Jolliffe I T . Principal Component Analysis[J]. Journal of Marketing Research, 2002, 87(4):513.
[9] Sugiyama M . Dimensionality Reduction of Multimodal Labeled Data by Local Fisher Discriminant Analysis[J]. J.mach.learn.res, 2007, 8(1):1027-1061.
[10] Pauca V P , Shahnaz F , Berry M W , et al. Text Mining Using Non-Negative Matrix Factorizations[C]// Siam International Conference on Data Mining. DBLP, 2004.
[11] Zia T , Zahid U . Long short-term memory recurrent neural network architectures for Urdu acoustic modeling[J]. International Journal of Speech Technology, 2019, 22(1):21-30.
[12] Devlin J , Chang M W , Lee K , et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[J]. 2018.
[13] Vaswani A , Shazeer N , Parmar N , et al. Attention Is All You Need[C]// arXiv. arXiv, 2017.
[14] Mikolov T , Chen K , Corrado G , et al. Efficient Estimation of Word Representations in Vector Space[J]. Computer Science, 2013.