Chinese Text Classification Based on Bert

Chengwei Zhang[1], Haoyang Ding[2], Xiaofeng Jin[3], Yanwei Chen[4], Zhicheng Zhang[5]

[1]31520221154235 School of Informatics, Xiamen University, Xiamen, China
[2]31520221154245 School of Informatics, Xiamen University, Xiamen, China
[3]31520221154246 School of Informatics, Xiamen University, Xiamen, China
[4]36920221153071 Institute of Artificial Intelligence, Xiamen University, Xiamen, China
[5]31520221154238 School of Informatics, Xiamen University, Xiamen, China

Abstract

In the area of natural language processing (NLP), text classification has always been an important task which attracts a considerable amount of research nowadays. Recent years have also witness the great achievement of various language processing models like GPT, ELMo, ERNIE in the text classification. In our work, we attempt to use a newly introduce language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers, to obtain a better performance in the Chinese text classification problem. Different from other NLP models, BERT is able to pretrain deep bidirectional representations from two special unsupervised tasks including Masked Language Model (MLM) and Next Sentence Prediction (NSP), which can result in a high accuracy in the classification task with only simple fine-tuning based on self-attention according to the datasets. We show the performance of our model on the dataset from THUCNews news headlines and draw a comparison among different language representation models. Experimental results show that the new BERT model can effectively improve the accuracy of the text classification of Chinese news headlines.

Introduction

Text classification refers to the process of human-generated text that come from multiple social media networks using different algorithms, programs, and techniques. It has been under spot light in the field of natural language processing (NLP).

Text classification aims to assign labels or tags to textual units such as sentences, queries, paragraphs, and documents. According to previous works, Language model pre-training has been shown to be effective for improving many natural language processing tasks. And the transformer architecture has dominated the domain of NLP tasks. With the strategies of fine-tuning, we can apply pre-trained model to downstream tasks such as text classification. Approaches to automatic text classification can be grouped into two categories: Rule-based methods and Machine learning based methods. CNN-based and RNN-based models has showed their advantages in text classification. with the word embedding and different architecture of neural network, the precision of text classification reaches a new level.

BERT [2] based on the bidirectional transformer, which consists of 340M parameters, is the current state-of-the-art embedding model. The advent of bidirectional encoder representation from transformers (BERT), which can generate contextualized word vectors, was an important turning point in the development of text classification and other NLP techniques. But with the rapid increase of parameters, the complexity of the architecture of the network needs to be reduced urgently. Many approaches are designed to reduce the redundant parameters and augment the robustness of models, such as data augment, regularization, dropout, and residual structure. Some of those approaches have been applied in BERT and its homogeneous network. we aim to design a transformation network of BERT to achieve high efficiency, accuracy, and easy to train.

In this paper, we propose a BERT-based model for text classification of Chinese news headlines by constructing and inserting BERT into other models as the embedding layer and training this model in THUCNews dataset [8], which aims to better incorporate task-specific knowledge into pretraining BERT and addresses the ask-awareness challenge. Furthermore, we also draw a comparison among different language representation models. Our contributions could be summarized as follows:

- We propose a Chinese text classification model implemented by BERT [2] based on Transformer. Achieve excellent accuracy under low computational cost.
- For comparison with other models, we apply ERNIE to this classification task and implement a cascade paradigm to achieve better performance.

Related Work

BERT. Bidirectional Encoder Representations from Transformers (BERT) [2] is a language representation model introduced by Jacob Devlin and his colleagues from Google in 2018. Since BERT is built as an unsupervised model which can be trained using a large number of plain text corpora available on the web in most languages, this combination of features makes BERT perform well on a variety of natural language processing tasks, including text classification. Text classification is a machine-learning task which divide the categories of text based on their content. It is a fundamental task in natural language processing with

different applications such as sentiment analysis, email routing, offensive language detection, spam filtering, and language identification. Although sufficient progress has been made in the performance of BERT-based models for text classification, there is still a lot of room for research, such as the problem of feeding BERT into other models as an embedding layer.

Text Classification. With the development of deep learning research, the application in the field of text classification has made significant progress. Zhilin Yang, Zihang Dai have published a new unsupervised language representation learning method based on a novel generalized permutation language modeling objective. Zhengyan Zhang and Xu Han have utilized both large-scale textual corpora and KGs to train an enhanced language representation model.

To figure out the problem that denoising autoencoding based pretraining like BERT relies on corrupting the input with masks and neglects dependency between the masked positions and suffers from a pretrain-finetune discrepancy, Zhilin Yang and his colleagues propose XLNet [9], which is a generalized autoregressive pretraining method that enables learning bidirectional contexts by maximizing the expected likelihood over all permutations of the factorization order and overcomes the limitations of BERT thanks to its autoregressive formulation. The neural architecture of XLNet is developed to work seamlessly with the AR objective, including integrating Transformer-XL and the careful design of the two-stream attention mechanism. XLNet achieves substantial improvement over previous pretraining objectives on various tasks. Zhengyan Zhang and his colleagues argue that informative entities in KGs can enhance language representation with external knowledge. ERNIE [10], which means Enhanced Language RepresentatioN with Informative Entities, can take full advantage of lexical, syntactic, and knowledge information simultaneously. For classification tasks, ERNIE modifies the input token sequence by adding two mark tokens to highlight entity mentions. These extra mark tokens play a similar role like position embeddings in the conventional relation classification models. The modified input sequence with the mention mark token [ENT] can guide ERNIE to combine both context information and entity mention information attentively.

According to some existing researches, feeding BERT as an embedding layer into other models will actually lead to its performance degradation in text classification tasks. Following this reason, in this work the authors try to compare the performance of BERT against several BERT-based models in text classification tasks. We wonder whether the similar situation occurs in the classification of long text in order to find out the reason for this problem.

Proposal Method

We make a Chinese text classification project based on a pre-training BERT-base model, using a dataset from THUCNews [8] news headlines with text lengths between 20 and 30. The dataset has a total of 10 categories, including finance, realty, stocks, education, science, society, politics, sports, games, and entertainment. Before text classification,

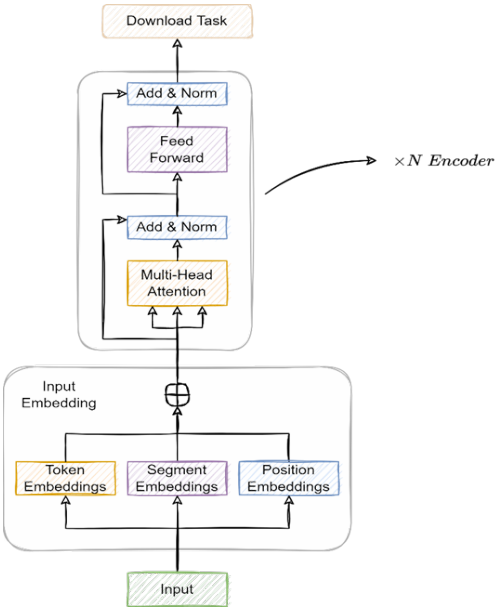

Figure 1: the structure of BERT model

the text should be pre-processed, and we have pre-processed the text with the following steps:

- **Tokenization**. The pre-processing module needs to separate the text into more linguistic feature units (tokens). For Latin languages, words are separated by spaces, and each word is a unit [7]. However, for Chinese, word segmentation is diverse, and the result of word segmentation directly affects the understanding of the text. Usually, we call this process as tokenization.
- **Numbering**. The data pre-processing module converts each language feature unit into a lexicographical order number according to the specified vocabulary, which is convenient for computer processing.
- **Sequence pre-processing**. Usually, the model expects each training sample to have the same length, and the text with insufficient length needs to be padded, and the text with too long needs to be segmented.

We divide the text classification task into two parts. First, we obtain a representation of the news text by a pre-trained BERT-base model. Secondly, the output of the Bert model is connected to different classification layers to get the categories of the text. The specific approach is as follows.

BERT for Text Classification. BERT is a NLP model that was designed to pre-train deep bidirectional representations from unlabeled text, and be fine-tuned for different NLP tasks using labeled text [2]. The BERT model consists of a series of layers, including an input layer, multiple encoder layers, and a final output layer. The structure of BERT is depicted in fig. 1.

BERT takes an input of a sequence of no more than 512 tokens (as shown in fig. 2). The input layer takes in a sequence of words and converts them into numerical representations known as word embeddings. These embeddings

Figure 2: the input of BERT model

capture the meaning of the words in the context of the input sequence. The input of BERT is a combination of three Embeddings, namely Token Embeddings, Segment Embeddings, and Position Embeddings. Token Embeddings adds additional special tokens [CLS] and [SEP] to the beginning and end of sentences. Segment Embeddings are used to distinguish between two different sentences. Position Embeddings are used to provide sequence order information to the model. For the text classification task, the BERT model inserts a [CLS] symbol in front of the text and uses the output vector corresponding to this symbol as the semantic representation of the whole text for text classification.

BERT-based model contains the encoder layers with 12 Transformer encoders, 12 self-attention heads, and the hidden size of 768. The encoder layers then process the input sequence using self-attention mechanisms to extract meaningful representations of the input. The self-attention mechanisms allow the model to attend to different parts of the input sequence simultaneously and extract relevant information from each part.

Finally, the output layer generates a prediction based on the extracted information. In the case of BERT, the output is often a classification or a language modeling prediction.

Masked Language Model. Masked Language Model (MLM) is a technique used to pre-train language models such as BERT. The goal of MLM is to predict the correct word or tokens in a sentence or paragraph given a few masked or replaced words, which allows the model to learn the relationships between words and their meanings, as well as the context in which they are used.

In the BERT model, MLM is used to train the model on a large dataset of text, such as a corpus of books or articles. During training, the model is presented with a sentence or paragraph with a percentage of the words masked out. The model is then required to predict the correct word or token that should go in the masked positions, based on the context provided by the rest of the sentence or paragraph. Generally, BERT select 15% of all tokens for MLM, and there are two main strategies for applying the MLM:

- **Whole word masking.** In this strategy, the model replaces a whole word with the [MASK] token, and the model is required to predict the original word based on the context provided by the rest of the sentence or paragraph.
- **Random word masking.** In this strategy, the model randomly masks out a percentage of the words in the dataset, and the model is required to predict the correct word based on the context provided by the rest of the sentence or paragraph.

Next Sentence Prediction. Next Sentence Prediction (NSP) is a technique used in the BERT model to train the model to understand the relationship between two sentences. During training, the model is presented with a pair of sentences and is required to predict whether the second sentence is a continuation of the first or not. Specifically, we choose the sentence A and sentence B for each pretraining example. In 50% of the cases, Sentence B is the actual next sentence following Sentence A, and is labeled as "IsNext". In the other 50% of cases, Sentence B is a random sentence taken from the corpus and is labeled as "NotNext".

Overall, the use of MLM and NSP in the BERT model allows it to learn the relationships between words and their meanings, as well as the context in which they are used, which can improve the performance of the model on downstream tasks such as language translation or question answering.

Fine-Tuning Strategies. Fine-tuning is a process of adapting a pre-trained machine learning model on a new dataset. This is done by continuing the training process on the new dataset, using the pre-trained model as a starting point. The goal of fine-tuning is to improve the model's performance on the new task by adjusting the model parameters to fit the characteristics of the new dataset. There are a few strategies that can be used when fine-tuning a model:

- **Freezing layers.** This involves keeping the weights of some layers of the pre-trained model fixed, while training the other layers on the new dataset. It is useful when the new dataset is relatively small and the pre-trained model is large, as it allows the model to retain the knowledge it has learned from the original dataset while adapting to the new data.
- **Fine-tuning all layers.** This involves training all layers of the pre-trained model on the new dataset. It is useful when the new dataset is large and the pre-trained model is well-suited to the new task.
- **Fine-tuning some layers.** This involves training only a subset of the layers in the pre-trained model on the new dataset. It is useful when the new dataset is small and the pre-trained model is large, as it allows the model to retain some of the knowledge it has learned from the original dataset while adapting to the new data.

Experiment

Experiment setup

- **Dataset.** We evaluate our model on the THUCNews dataset [8], which is made by filtering the historical data of Sina News RSS subscription channel from 2005 to

Method	mAP	mAR	train acc	test acc
ERNIE	94.78	94.75	95.31	94.76
bert+TextCNN	94.44	94.37	96.88	94.39
bert+TextRCNN	94.51	94.49	96.09	94.49
bert+TextRNN	94.69	94.66	95.31	94.66
bert+DPCNN	94.40	94.34	96.88	94.33
bert	94.91	94.88	98.44	94.88

Table 1: the result of comparison among different models

category	precision	recall	f1-score	support
finance	0.9331	0.9490	0.9410	1000
realty	0.9570	0.9580	0.9575	1000
stocks	0.9238	0.9090	0.9163	1000
education	0.9625	0.9750	0.9687	1000
science	0.8967	0.9290	0.9126	1000
society	0.9560	0.9340	0.9449	1000
politics	0.9339	0.9320	0.9329	1000
sports	0.9840	0.9870	0.9855	1000
game	0.9875	0.9450	0.9658	1000
entertainment	0.9566	0.9700	0.9633	1000
accuracy	-	-	0.9488	10000
macro avg	0.9478	0.9475	0.9476	10000
weighted avg	0.9478	0.9475	0.9476	10000

Table 2: the classification result using BERT

2011. The dataset contains 740,000 news documents in UTF-8 plain text format. We extracted 200,000 news headlines

from THUCNews as the base dataset for the experiment, and set the text length between 20 and 30. Specifically, we utilized a stratified sampling method, with the amount of news data in each category as the weight, randomly selected 10 categories, and stochastically sampled 20,000 data from each of the 10 categories, with data in words as the basic unit. Through the above method, we finally trained the model and evaluated its performance based on 200,000 data from 10 categories: finance, real estate, stock, education, technology, society, current affairs, sports, games, and entertainment. We divide the dataset into training set, validation set, and test set by uniform sampling and random sampling, which contain 180,000, 10,000 and 10,000 data in 100 categories, respectively. Our experiments are tested on the closed-set setting. The results of our experiment are summarized in Table 1 and Table 2.

- **Baselines.** In our experiment, we use some baselines for the comparison against BERT. We compare to ERNIE [10], TextCNN [1], TextRCNN [5], TextRNN [6] and DPCNN [4].

  TextCNN is a model that applies CNN in NLP tasks, with a simple CNN with little hyperparameter tuning and static vectors achieves excellent results on multiple benchmarks. TextRNN uses LSTM to deal with the text classification, and TextRCNN take a further step to generate the new embedding by connecting the word embedding and the context embedding. DPCNN is a low-complexity word-level deep convolutional neural network architecture for text categorization that can efficiently represent long-range associations in text.

- **Evaluation Metrics.** We use mean average Precision (mAP) [3], mean average recall (mAR), and accuracy as the evaluation metrics of the model to evaluate the performance of different models, mAP stands for a set of detections is the mean over classes, of the interpolated average precision for each class, while this per-class AP is given by the area under the precision/recall (PR) curve for the detections. Similarly, mAR stands for the mean over classes, of the interpolated average recall for each class.

  Furthermore, in order to better evaluate the performance on the classification task of each model, we also analyze the precision, recall, and f1-score of each category of the dataset.

- **Implementation Details.** In our experiment, the batch size is set to 128, input length is fixed to 32, the initial learning rate is set to 0.00005, the number of hidden layer units is set to 768, and the training is set to stop when the performance improved by 1000 training sessions is small enough. In the Bert-based model, for the BERT+CNN model, we use convolution cores with sizes [2, 3, 4] and a number of 256 for feature extraction, and use dropout=0.1 to improve the performance of the model. The corresponding BERT+DPCNN increases the number of convolution cores of BERT+CNN to 250, further strengthening feature extraction. BERT+RCNN and BERT+RNN follow the setting of BERT+CNN with two additional RNN layers of 256 and 768 neurons, respectively.

Experiment results. Table 1 demonstrates our main results compared to some other models. Specifically, we choose ERNIE for the comparison. Since ERNIE, a powerful pre-trained model, is utilized to deal with NLP tasks and has achieved sort of the art performance on various Chinese NLP tasks, thus ERNIE could be a obvious comparison against BERT on the task of Chinese text classification. The main difference between BERT and ERNIE is the masking strategy. In our experiment, we train both BERT and ERNIE in both English and Chinese vocabulary. However, as it is shown in the Table 1, we observe that the BERT model performs better than ERNIE on the THUCNews dataset, although the main reason is not yet clear. Also, we attempt to connect the BERT with other networks, that is, to use the embedding output of the BERT as the input of other models to further do the inference. However, it is found that the performance drop on all evaluation metrics on both train and test datasets. The BERT alone can score 94.91% on mAP, 94.88% on mAR, and 94.88% on the accuracy, which proves to be the best model.

Table 2shows the classification result using BERT on the

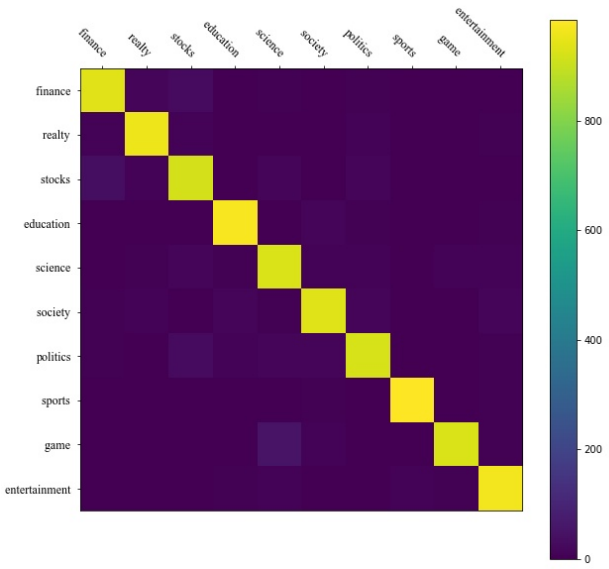

Figure 3: visualization of the confusion matrix

test dataset. The test set consists of 10 categories with 1000 samples in each category. We can observe that The results show that high performance was achieved in all categories. Macro-average and weighted averages were also taken to measure the overall classification performance of the 10 categories.BERT also attains a high performance on these two metrics.

Additionally, we also plot a confusion matrix of the classification result for the BERT model. Figure 3 shows the visualization of the confusion matrix. In this figure, the x-axis represents the input text, and the y-axis represents the output classification results. The color is more yellow, the higher the correlation. From Figure 3, we can see that our model has achieved excellent performance in ten-category text classification task, especially in education and sports corpus.

Conclusion

We present a text classification model based on a pre-training BERT-based model. Both were trained and tested on a dataset that contains 10 categories of the corpus. We pre-process the dataset in three steps, tokenization, numbering, and sequence processing. And then deploy this masked language model aim to fulfill our task with fine-tuning. We also implement ERNIE on this dataset to figure out whether our method is better or not. We obtain results compared to other models on the THUCNews dataset, which show our model achieves the best result without cascading with other components. And acquire the desired result on every category as illustrated in Table 2.

To explore a more suitable model architecture and probe better performance on this task, we cascade TextCNN, TextRCNN, TextRNN, and DPCNN to pre-trained BERT. On the contrary, instead of achieving better performance, the accuracy descended. Maybe these cascade models have higher complexity, resulting in overly complex models overfitting on our dataset.

In future research, we can still use this paradigm but implement a more harsh regularization strategy to counteract this overfitting phenomenon and gain better performance. Besides, we may expand our dataset to improve the diversity of data. And apply our model or cascade model to classify more than just ten categories. Recently, the diffusion model perform fairly wonderfully on text generation tasks, so the diffusion model can be taken into consideration to complete the dataset.

References

[1] Chen, Y. 2015. *Convolutional neural network for sentence classification*. Master's thesis, University of Waterloo.

[2] Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[3] Everingham, M.; Eslami, S.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2015. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1): 98–136.

[4] Johnson, R.; and Zhang, T. 2017. Deep pyramid convolutional neural networks for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 562–570.

[5] Lai, S.; Xu, L.; Liu, K.; and Zhao, J. 2015. Recurrent convolutional neural networks for text classification. In *Twenty-ninth AAAI conference on artificial intelligence*.

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

[7] Riloff, E.; and Lehnert, W. 1994. Information extraction as a basis for high-precision text classification. *ACM Transactions on Information Systems (TOIS)*, 12(3): 296–333.

[8] Sun, M.; Li, J.; Guo, Z.; Yu, Z.; Zheng, Y.; Si, X.; and Liu, Z. 2016. Thuctc: an efficient chinese text classifier. *GitHub Repository*.

[9] Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R. R.; and Le, Q. V. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

[10] Zhang, Z.; Han, X.; Liu, Z.; Jiang, X.; Sun, M.; and Liu, Q. 2019. ERNIE: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*.