A multi-label text classification model based on ELMo and attention

Wenbin Liu, *Bojian* Wen, *Shang* Gao*, *Jiesheng* Zheng and *Yinlong* Zheng

Guangdong Power information Technology Co., Ltd. Yuedian Buliding, 6-8 Shuijun Road, Yuexiu District, Guangzhou, Guangdong, China

Keywords: Sentiment classification, Bidirectional gated recurrent unit, Text classification.

Abstract. Text classification is a common application in natural language processing. We proposed a multi-label text classification model based on ELMo and attention mechanism which help solve the problem for the sentiment classification task that there is no grammar or writing convention in power supply related text and the sentiment related information disperses in the text. Firstly, we use pre-trained word embedding vector to extract the feature of text from the Internet. Secondly, the analyzed deep information features are weighted according to the attention mechanism. Finally, an improved ELMo model in which we replace the LSTM module with GRU module is used to characterize the text and information is classified. The experimental results on Kaggle's toxic comment classification data set show that the accuracy of sentiment classification is as high as 98%.

1 Introduction

Sentiment classification is one of the most important tasks in natural language processing. The Internet is highly developed nowadays, people make comments through Blog, news website, forum and other social apps. The publication of these speeches is characterized by indefinite length, unlimited vocabulary and lack of strict grammar rules, which has a strong subjective tendency. Among them, negative speech or toxic comments are important issues that needs urgent attention. If those toxic comments can't be correctly identified, we can't prevent the occurrence of network violence in time and prevent the behavior which endangers the reputation of individuals and even enterprises. At the same time, the correct speech sentiment classification can help the government understand the public opinions, and enterprises listen to the voice of users. In this context, the sentiment classification for power grid related web text is of great research significance.

Sentiment classification of text is mainly based on two methods: sentiment polarity dictionary and traditional machine learning method, including the construction of sentiment resources, sentence segmentation, feature information extraction, quality analysis, etc. [1]. The advent of the Internet era promotes the birth of various new words, which has a great

* Corresponding author: gaoshang8202@126.com

impact on the sentiment classification model especially for the sentiment polarity dictionary. Existing sentiment polarity dictionary is limited and cannot effectively identify the new words or popular words. Manek and Shenoy used traditional machine learning algorithms to analyze comments' sentiments. They compared the performance of naive Bayes, ME and SVM in terms of accuracy and F1 score. The results show that SVM has the best classification accuracy. With the development of deep learning technology, deep neural network performs well in natural language processing tasks [2]. Kim used convolutional neural network (CNN) to solve the sentiment classification problem and achieved good results [3]. Santos also uses the deep CNN to analyze the sentiment contained in the text and shows that deeper network can improve the classification performance. Irsoy proves that LSTM, as a recurrent neural network, is also an effective method to solve the sentiment classification task [4]. Bahdanau firstly introduced the attention mechanism into the machine translation in NLP [5]. Qu and Wang proposed a sentiment analysis model based on hierarchical attention network, which has a greater improvement than traditional recurrent neural network [6]. However, their method takes long time to train and is lack of efficiency.

Based on existing studies and latent problems, a multi label text classification model based on ELMo and attention mechanism is proposed. ELMo model can be seen as an improvement based BiLSTM model. We modify the original ELMo model and replace the LSTM module with GRU and greatly improves the training speed of the network while ensuring the accuracy of classification. The difference between LSTM and ELMo is that the former uses fixed word embedding and cannot fine tune the embedding vector in context. ELMo can generate deep contextualized word representations and dynamically adjust the representation of word which solves the problem of polysemous words and makes word embedding more contextual. The utilization of attention mechanism enables neural network to focus on the important information in the embeddings, that can further improve the classification effect compared with LSTM. In addition, we use the pretrained word embedding vector and transfer learning technology to shorten the training time and get a better word embedding representation, which can be used for power grid related web text sentiment classification.

2 Related work and proposed model

2.1 BiLSTM

BiLSTM is an extension of ordinary RNN [7]. The difference between RNN and ordinary neural network is that the neuron will not only receive the input of the current time point, but also receive the output of the previous neuron, which solves the problem of considering the past information in the text. In practical application, considering the information in the past is not enough and we also need to bring the past information to present. To solve this problem, bidirectional RNN (BiRNN) was born. BiRNN adopts a reverse strategy on the ordinary RNN, that is, the input sequence is reversed and then the output is calculated again. The final result is the stack of forward RNN and backward RNN results. In theory, BiRNN can take the whole context information into account, but in practical application, it is found that BiRNN is difficult to deal with information with long-term dependence. A simple example is that when generating English sentences, if the sentences are very long, RNN cannot remember the singular or plural forms of subjects and select appropriate predicate verbs. In order to solve this problem, LSTM introduces the gating mechanism, including forget gate, input gate and output gate. Among them, forgetting gate is used to control the proportion of input information passing at the current moment. The specific calculation method is as follows:

$$h_{f_t} = \sigma(W_{xh_f} x_t + W_{h_f h_f} h_{f_{t-1}} + b_{h_f}) \tag{1}$$

$$h_{b_t} = \sigma(W_{xh_b} x_t + W_{h_b h_b} h_{b_{t-1}} + b_{h_b}) \tag{2}$$

$$y_t = W_{h_f} h_{f_t} + w_{h_{by}} h_{b_t} + b_y \tag{3}$$

where y indicates the output vector, $x_t \in \Re^d$ is the input vector with d dimensionality in time point t. W represents the weighting matrix and b is the bias vector. $h_f \in \Re^d$ and $h_b \in \Re^d$ means the input vector and output vector in the LSTM respectively.

2.2 Bidirectional gated recurrent units

Bidirectional gated recurrent neural unit (BiGRU) can be regarded as an extension of BiLSTM, replacing the LSTM module with GRU [8]. GRU combines the hidden state and cell state of LSTM into one state, so it significantly shortens the training time and improves the training speed for large corpus texts. More specifically, after GRU reads the word embedding vector t_i and the hidden layer state vector h_{i-1}, the output vector c_i and the hidden layer state vector h_i are generated through gating calculation. For the specific calculation method, refer to the following formula:

$$z_i = \sigma(W_z t_i + V_z h_{i-1} + b_z) \tag{4}$$

$$r_i = \sigma(W_z t_i + V_z h_{i-1} + b_r) \tag{5}$$

$$c_i = \tanh(W t_i + V(r_i \odot h_{i-1}) + b) \tag{6}$$

$$h_i = z_i \odot h_{i-1} + (1 - z_i) \odot c_i \tag{7}$$

Where $z \in \Re^d$ and $r \in \Re^d$ indicate the input gate and reset gate for a d dimensional input vector, $\{W_z, W_r, W, V_r, V\}$ represent weighting matrices, $\{b_z, b_r, b\}$ are bias vectors, $\odot$ represents point-wise matrix multiplication.

2.3 Attention mechanism

Attention mechanism is first proposed in computer vision, which is inspired by human visual processing process, that is, human brain will not process all information after receiving visual input but focus on specific parts. This mechanism has been widely used in many fields, including image title generation, text classification, speech recognition and machine translation [9].

In neural networks, attention mechanism can be regarded as a resource allocation method, which can allocate more attention or computing resources to important information, which is helpful to solve the problem of information overload. In practice, attention mechanism can be divided into two types: one is top-down focused attention, which is usually conscious and task-related, and focuses on an object actively. The other is the bottom-up unconscious attention, which has nothing to do with tasks and is mainly driven by the outside, also known as saliency-based attention. For example, in convolutional neural networks (CNN) and LSTM, pooling and gating can be seen as saliency-based attention mechanisms.

The input data of neural network is represented by vectors. We use $[x_1, \cdots, x_n]$ to represent task-related input vectors. In order to give more weights to specific data, attention mechanism introduces query vector q, which calculates the correlation between query vector and input vector through a score function. At the same time, an attention variable $t \in [1, N]$ is introduced to represent the selected index position. Detailed calculations are as follows:

$$\begin{aligned} a_i &= p(t=1 \mid X, q) \\ &= soft\,max(s(x_i, q)) \\ &= \frac{exp(s(x_i, q))}{\sum_{j=1}^{N} exp(s(x_i, q))} \end{aligned} \tag{8}$$

where a_i is the attention distribution and $s(x_i, q)$ is the score function. There are various ways to define the score function and in this paper, we adopt the self-attention model based on Scaled Dot-Product operation. Scaled Dot-Product is defined as below:

$$s(x_i, q) = \frac{x_i^T q}{\sqrt{d}} \tag{9}$$

where d represents the dimensionality of input vector. Scaled Dot-Product model can be viewed as an improvement of Dot-Product, the difference lies in the former is divided by the square root of d. We can see that when d is very large, Dot-Product model will have a large variance which leads to a small gradient for the *SoftMax*. This problem can be solved by proposed Scaled Dot-Product model.

2.4 ELMo

Unlike most widely used word embeddings, ELMo word representations are functions of the entire input sentence. We use the top two-layers of BiGRU to compute with character convolutions. Next, using convolutional filters allows us to pick up on n-gram features that build more powerful representations. This then forms the core of the ELMo language model. Now, assuming that we have the k th word embedding in our input, by the trained 2-layer language, we take the word embedding x_k as well as the bidirectional hidden layer representations $h_{1,k}$ and $h_{2,k}$ and combine them. After that, we can get a new weighted task specific word representation. The whole process looks as follows and we use “happy” as an example:

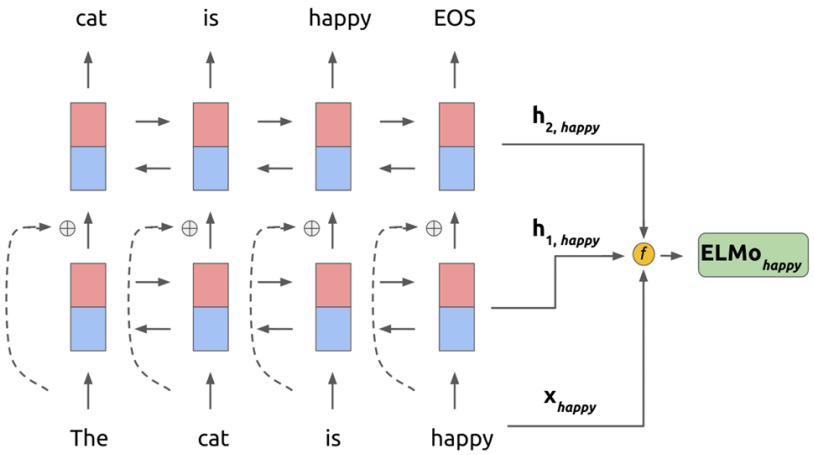

Fig. 1. An example of combining the bidirectional hidden representations and word representation for "happy" to get an ELMo-specific representation.

3 Experiment

3.1 Dataset

In the experiment, we use a dataset from the famous data science contest Kaggle to test our model. The data set is made up of the comments from Wikipedia. The tags are all manually labeled. There are six kinds of tags defined in the experiment, which are "toxic", "severe_toxic", "obscene", "thread", "result" and "identity_hate. Each comment may have multiple tags or no tags. The model needs to give the probability for each comment on six kinds of tags, so this is a multi-label text classification problem. The distribution of all categories on the dataset is shown in Figure 2:

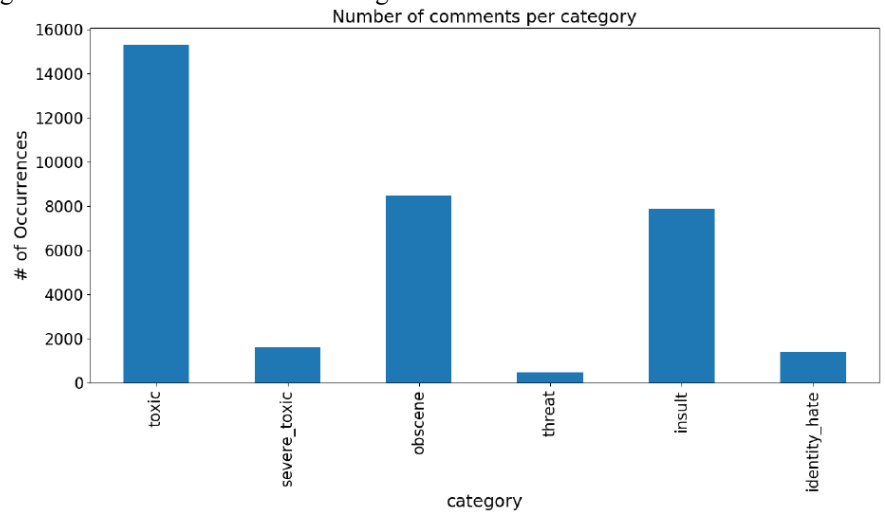

Fig. 2. Number of six categories in toxic comments dataset from Wikipedia.

3.2 Design

Our network structure is shown in the Figure 3:

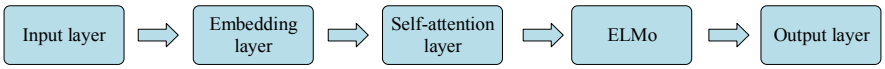

Fig. 3. Structure of proposed ELMo and self-attention model.

In the experiment, the input layer contains 200 neurons, i.e. the first 200 characters of each comment are taken, and if the input comment is less than 200 characters, it will be automatically completed. The number of neurons in embedding layer, attention layer and ELMo are 100, 128 and 256 respectively, and the output layer is a fully connected layer composed of six neurons which calculate the probabilities of input comment on six kinds of categories. We use the embedding layer as a transfer learning technology to load pre-trained word embedding vectors, so as to shorten the training time and get a better representation of input data. The attention layer is trained to assign higher weights to task-related word vectors and improve the classification accuracy. The output layer replaces the SoftMax layers with a fully connected layer. Each neuron outputs a value between [0,1] to represent the probability of a specific category.

In order to make the neural network fully learn the features of the text, we use the pre-training word vectors and self-attention mechanism. In the experiment, we use Glove word vectors, which are generated by co-occurrence matrix decomposition. Each word is represented by a 100-dimension vector. The shorter the distance between vectors, the higher the similarity between two words in meaning. The word vector set is trained from a 6 billion token corpus, which contains a total of 400k characters, and provided by the research team from Stanford University. Self-attention model introduce a Q, K and V query vector sequence and the Scaled Dot-Product is used as the score function to dynamically generate different connection weights, which can be used to deal with the variable-length sequences. We divide the dataset into two parts, training set and test set, in which the number of test set comments accounts for 20% and the number of training set accounts for 80%.

The configurations of experimental platform are listed as below:
Operation system: windows 10
CPU: Intel Core i7-6700
RAM: 32 GB
Deep learning framework: TensorFlow 1.13.1
Developing tool: Visual Studio Code
Programming language: Python 3.6

3.3 Experiment result

We use Adam algorithm to train our neural network. Adam is a first-order optimization algorithm based on stochastic gradient descent (SGD). The learning rate of SGD does not change during the training process. Unlike SGD, Adam estimates the dynamic learning rate by calculating the first order and second-order moments of the gradient, which is an adaptive learning rate optimization algorithm. At the same time, it combines the advantages of AdaGrad and RMSProp. The experimental results show that Adam algorithm has excellent performance, as shown in Figure 4. After using Adam algorithm, the loss on training set and test set can be reduced to about 0.05.

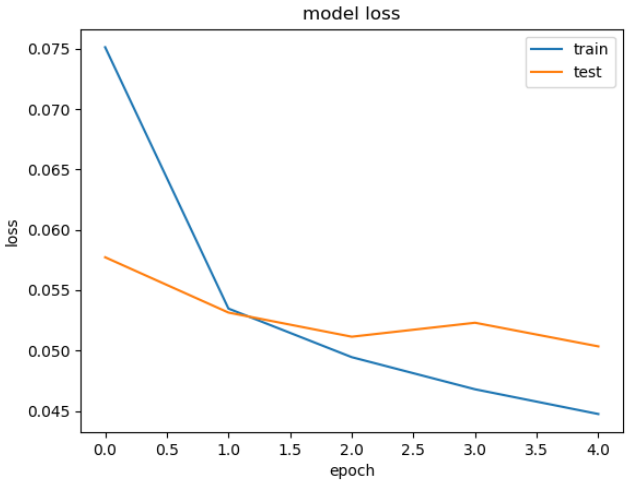

Fig. 4. The model loss by Adam on training set and test set.

In the experiment, we set the batch as 128. After 4 epochs the algorithm almost converge and achieve an accuracy more than 98% on the test set. The results are shown in Figure 5.

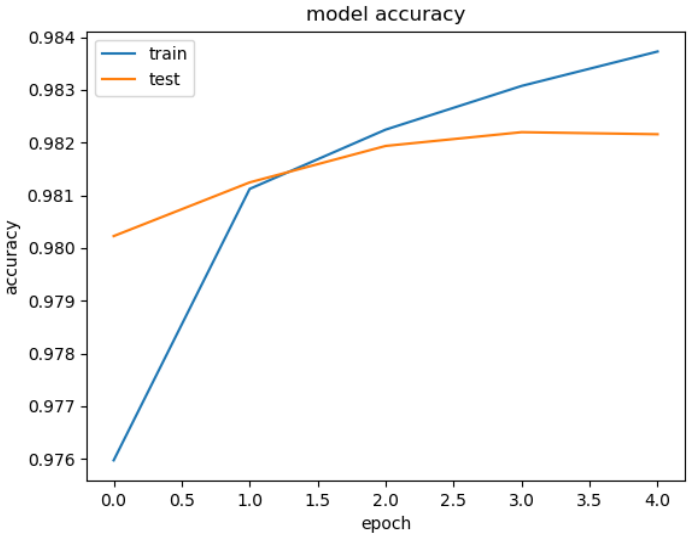

Fig. 5. The training and test accuracy in each epoch.

In a word, the time consuming of training neural network is greatly reduced by using pre-training word vector and ELMo. In this experiment, GPU is not used to accelerate the calculation, and the CPU time consuming is about 10min. At the same time, the use of embedding layer and self-attention mechanism improves the accuracy of classification, which is 2% higher than the baseline model BiLSTM. The test results of the integration of ELMo and self-attention mechanism on the benchmark dataset show that our model is suitable for the task of multi-label text classification and can be applied and deployed in the grid text.

4 Conclusion

This paper presents a multi-label text classification model based on ELMo and self-attention mechanism. Compared with the BiLSTM model, it achieves the higher accuracy with less training time. By using self-attention mechanism, neural network can focus on the important information that improves the classification accuracy. In addition, the pre-trained word vector and transfer learning technology further shorten the training time and obtain better word vector representation. The experimental results show that the model has a good performance in the free text, and it is also suitable for the application scenarios such as sentiment classification in power grid related text.

This research was financially supported by the Self-financing for Guangdong Power Grid Co., Ltd. informatization project under Grants 037800HK42180056.

References

1. Jieru Jia, Qiuqi Ruan, Gaoyun An, Yi Jin. Multiple metric learning with query adaptive weights and multi-task re-weighting for person re-identification [J]. Computer Vision and Image Understanding, 2017, 160.
2. Liu J, Zhang Y. Attention modeling for targeted sentiment. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: 2017, Volume 2, p572-577.
3. Kim Y, Convolutional neural networks for sentence classification, EMNLP: 2014.
4. Trofimovich J. Comparison of neural network architectures for sentiment analysis of Russian tweets. Proceedings of the International Conference Dialogue 2016, RGGU.
5. Bahdanau, Dzmitry, Cho, Kyunghyun, Bengio, Yoshua. Neural Machine Translation by Jointly Learning to Align and Translate [J]. Computer Science, 2014.
6. Zhaowei Qu, Yuan Wang, Xaioru Wang. A Hierarchical Attention Network Sentiment classification Algorithm Based on Transfer Learning [J]. Journal of Computer Applications, 2018, 38(11):3053-3056.
7. Yao Y, Huang Z: Bi-directional LSTM recurrent neural network for Chinese word segmentation. In: International Conference on Neural Information Processing: Springer; 2016: 345-353.
8. Zhang Q, Ma Y, Gu M, Jin Y, Qi Z, Ma X, Zhou Q: End-to-End Chinese Dialects Identification in Short Utterances using CNN-BiGRU. In: 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC): 2019: IEEE; 2019: 340-344.
9. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I: Attention is all you need. In: Advances in neural information processing systems; 2017: 5998-6008.