YNU-HPCC at SemEval-2019 Task 9: Using a BERT and CNN-BiLSTM-GRU Model for Suggestion Mining

Ping Yue, Jin Wang and Xuejie Zhang
School of Information Science and Engineering
Yunnan University
Kunming, P.R. China
Contact: xjzhang@ynu.edu.cn

Abstract

Consumer opinions towards commercial entities are generally expressed through online reviews, blogs, and discussion forums. These opinions largely express positive and negative sentiments towards a given entity; however, they may also contain suggestions for improving the entity. In this task, we extract suggestions from a given unstructured text, in contrast to the traditional opinion mining systems. This type suggestion mining is more applicable and extends capabilities. In this paper, we propose the use of bidirectional encoder representation learned from transformers (BERT) to address the problem of domain specific suggestion mining in task A. In detail, BERT is also used to extract feature vectors and perform fine-tuning for the task . For Task B, we applied an ensemble model to combine the BiLSTM, CNN, and GRU models, which can perform cross domain suggestion mining. Officially released results show that our system performs better than the baseline algorithm does.

1 Introduction

Suggestion mining is used to extract advice from text such as that provided in online reviews, blogs, discussion forums, and social media platforms where consumers share their opinions towards commercial entities like brands, services, and products. Most of the traditional sentiment analysis methods are emotion classifications. Opinion mining can improve service and quality. Such systems have become an effective way for marketing, economics, politics, and advertising. The application of suggestion mining provides the motivation, for the SemEval 2019 Task 9 (Negi et al., 2019), which contains two subtasks that classify given sentences into suggestion and non-suggestion classes. Subtask A requires a system to achieve domain specific training, whereby the test dataset will belong to the same domain as the training and development datasets. This was part of a suggestion forum for windows platform developers. Subtask B applies the system to cross domain training, where training, development, and test datasets will belong to different domains. Training and development datasets will remain the same as Subtask A, while the test dataset will belong to the domain of hotel reviews.

There are many methods in sentiment analysis. In many reports on this subject, it has been implied that these models help improve classification. Successful models include convolutional neural networks (CNN), long short-term memory (LSTM), and bi-directional LSTM (BiLSTM). CNN can capture local n-gram features, while LSTM can maintain memory in the pipelines and solve the problem of long sequence dependence in neural networks.

In this paper, we propose a bidirectional encoder representation learned from transformers (BERT) model (Devlin et al., 2018) for Task A. It comprises two phases. The first phase is called pre-training and is similar to word embedding. The second phase is called fine-tuning and uses a pre-trained language model to complete specific NLP downstream tasks. We used a pre-trained model that was provided by Google AI team. It included weights for the pre-trained model and a vocab file that maps component words of sentences to indexes of words. It also included the JSON file, which specifies the model hyper-parameters. Fine-tuning was applied to sequence classification: the BERT directly takes the final hidden state of the first [CLS] token, adds a layer of weight, and then softmax predicts the label probability. The structure is shown in Figure 2.

For Task B, we apply the bert model to test data, the score is 0.343, it is very low. the reason is that the task B is cross-domain training, so we intro-

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 1277–1281
Minneapolis, Minnesota, USA, June 6–7, 2019.

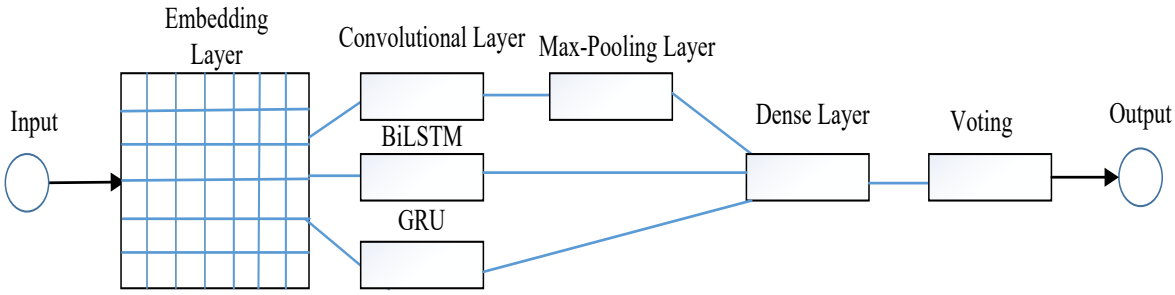

Figure 1: The CNN-BiLSTM-GRU architecture

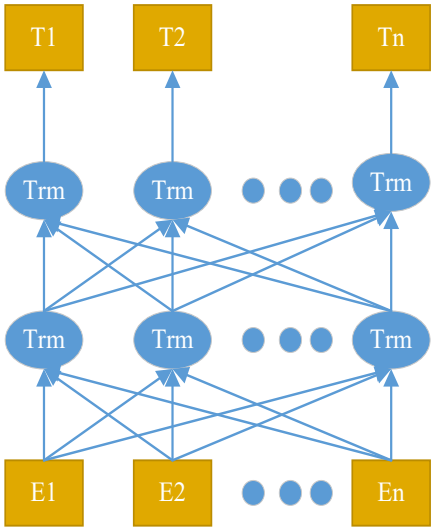

Figure 2: The architecture of BERT

duced an ensemble model that includes the CNN, BiLSTM, and GRU model. The structure is shown in Figure 1. We constructed the word vectors from 300-dimensional Glove vector. Then, a word vector matrix was loaded into the embedding layer. After this, the CNN applies the convolutional layer and max pooling layer to extract n-gram features, and passes through the dense layer to classify the sentence. BiLSTM can obtain the semantic information from the context. The forward and backward layers are connected to the output layer. GRU has a structure similar to that of LSTM, but is simpler. Finally, we combined CNN with BiLSTM and GRU using a soft-voting method, and output the results. The experimental results show that our model has good performance. According to the official review, we achieved sixth place among the 34 teams working on Task A.

The rest of the paper is organized as follows. In Section 2, we describe the BERT model. There, we also detail CNN, BiLSTM, and GRU and their combination. The comparative experimental results are presented in Section 3. Conclusions are drawn in Section 4.

2 The BERT and CNN-BiLSTM-GRU model

Figure 2 shows the BERT model. First, for each token, a representation is generated by its corresponding token embedding, segment embedding, and position embedding. Word-Piece was embedded (Wu et al., 2016) along with 30,000 token vocabularies. Finally, an output layer was used to fine-tune the parameters. Figure 1 shows the ensemble model used to combine the CNN, BiLSTM, and GRU models. First, all component words were transformed to a feature matrix by an embedding layer. Then a convolutional layer and a max pooling layer, were used for feature extraction. To avoid over-fitting, a dropout layer was used after both convolution and max-pooling layers. BiLSTM outputs predictive label sequence directly to input sentences. GRU is a variant of LSTM that has fewer parameters and is relatively easier to train. We embedded these models with the vote method and finally output the result.

2.1 Bidirectional Encoder Representations from Transformers (BERT)

Input characterization. For the task of sentence classification, BERT will add the [CLS] and [SEP] identifiers to the beginning and end of each input text; thus, the maximum sequence length can be described as follows.

$$max\_seq = S_t + 2 \quad (1)$$

where *max_seq* denotes the maximum sequence length, and *St* is the set text length. We set the S_t=78, and *max_seq*=80. For every input sentence, BERT introduces masked language mode, and next sentence prediction. Input embedding is

the sum of token embedding, segmentation embedding, and position embedding.

Transformer. The multi-layer transformer Vaswani et al. (2017) structure operates through the attention mechanism to convert the distance between two words at any position into the numeral 1. Owing to the transformer's overall architecture, the input sequence will first be converted into a word embedding vector, which can be used as the input of the multi-head self-attention module after adding the position coding vector. The output of the module can be used as the output of the encoder module after passing through a fully connected layer.

Output.The highest hidden layer of [CLS] is directly connected to the output layer of softmax as a sentence. The output result of BERT is label probability. The sum of the probabilities of all labels is 1, and the probability value of returning a label is the same as the order of setting labels in MrpcProcessor. This task sets the labels to 0 and 1. The probability of the first column returned in this experiment is 0, and the probability of the second column is 1.

2.2 CNN

Embedding Layer. The embedding layer is the first layer of model. Load Glove (Lee et al., 2016; Cun et al., 1990) with word embedding (Zahran et al., 2015) and is used for model initialization of online reviews. The embedded layer converts a positive integer (subscript) into a vector of fixed size N. N is defined as 80; any sentence exceeding this size is reduced to 80, and any sentence with a size less than 80 is padded to 80 by adding 0s.

Convolution Layer. The convolution layer is used to extract n-gram features from the embedding matrix. The calculation method of the convolution layer is as follows,

$$conv = \sigma(Mat \circ W + b) \tag{2}$$

where σ is an activation function, *Mat* indicates an embedding matrix, W and b respectively denote convolution kernel and bias. Here, $\circ$ is a convolution operation. We use 3*3 convolution kernels. The activation function is ReLU (Nair and Hinton, 2010)

Max pooling layer. Pooling is selecting a part of the input matrix and is used to choose the best representative for the region. The max pooling layer selects the max feature.

Dropout Layer. To avoid over-fitting, we introduce the dropout layer (Hinton et al., 2012) after both a convolution layer and max pooling layer, which is to randomly throw away some weight of the current layer. It can reduce model complexity and enhance the generalization ability of the model.

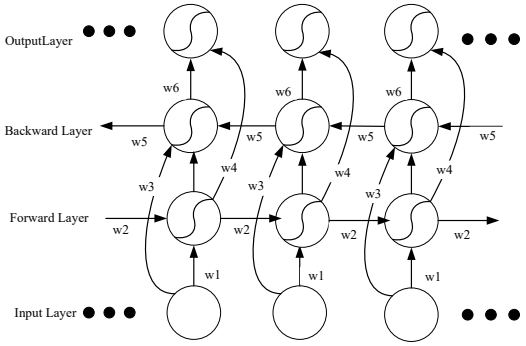

Figure 3: The architecture of BiLSTM

2.3 BILSTM and GRU

Bi-directional Long Short-Term Memory (BiLSTM) (Brueckner and Schuller, 2014; Li and Chang, 2015) is a variant of Recurrent Neural Network (RNN). Owing to its design characteristics, BiLSTM is ideal for modeling time-series data such as text data. Figure 3 shows the BiLSTM structure. BiLSTM is an abbreviation of LSTM Graves (2012); Greff et al. (2016); Graves (2012), which is a combination of forward LSTM and backward LSTM. Both are often used to model context information in natural language processing tasks.

Gated Recurrent Unit (GRU)(Cho et al., 2014) is a variant of LSTM, although the model is simpler than the standard LSTM model. It combines a forget gate and input gate into a single update gate. It also mixes cell state with hidden state.

$$\begin{aligned} z_t &= \sigma(W_z \cdot [h_{t-1}, x_t]) \\ r_t &= \sigma(W_r \cdot [h_{t-1}, x_t]) \\ \tilde{h}_t &= \tanh(W \cdot [r_t * h_{t-1,} x_t]) \\ h_t &= (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t) \end{aligned} \tag{3}$$

where h_t is hidden states, x_t is the input vector, σ is the sigmoid function, and r_t and z_t are the reset and update doors, respectively.

2.4 Ensemble

Each classifier is independently classified, and the integrated model can improve the correct rate. In this task, each base learner has a predicted value, and we used a soft-voting classifier as the final predicted value. The soft-voting classifier predicts the class label based on the sums of the predicted probabilities. The corresponding type with the highest probability is the final prediction result.

Model	Trial	Test
CNN	0.505	0.216
BiLSTM	0.498	0.180
CNN-BiLSTM	0.667	0.210
BERT	**0.851**	**0.735**

Table 1: The experiment results.

Parameters	BiLSTM/GRU	CNN
Neurons	60	120
Dropout rate	0.4	0.0
Weight	2	5
Activation	softmax	softmax
Init mode	LeCun	LeCun
Learning rate	0.001	0.2
Momentum	0.4	0.4

Table 2: The best-tuned parameters.

3 Experiments and Evaluation

In this section, we report the experiments were conducted to evaluate the proposed models on both sub-tasks. We also report the results of the official review. The details of the experiment are described as follows.

3.1 Data Preparation

Subtask A. Organizers provided training data from online forum comments that included three parts: id, sentence, and label. The given label is 0, indicating that the suggestion is not recommended. Here, 1 indicates a positive suggestion. This is equivalent to the suggestion mining used to discover sentences with suggestions. The positive and negative emotional statements of a given data set are unbalanced, and the negative emotions are three times more abundant than the positive emotions. According to this situation, we used oversampling to process the data; the positive emotional sentences were randomly copied from the training set in the same proportion as the negative emotional sentences. We extracted 0.2 ratio data as a validation set in the training set. In this experiment, we used the BERT-Base model which is pre-trained by Google AI team to process the text.

Subtask B. To address the problem of imbalance in data distribution, Task B uses the define loss function. In our model, we introduced a focal-loss function (Lin et al., 2017) that reduced the weight of many negative samples in training. This loss function is a dynamically scaled cross entropy loss function. As the correct classification increases, the scale factor in the function is reduced to zero. This scale factor can automatically reduce the impact of simple samples during training. Quickly focus your model on difficult samples. Data processing removes stop words, replaces URLs with <urls>, and removes characters except for alphanumeric characters and punctuation.

3.2 Implementation Details

Subtask A. In this experiment, TensorFlow (GPU backed) was used. We used the BERT-Base model to process the data. We introduced other three models (CNN, BiLSTM, and BiLSTM) as baseline algorithms. We combined commonly used parameters to tune-in the training. For the task, the batch size was 30, the learning-rate set was 2e-5, and the number of the training epoch was 10.

Subtask B. We used Scikit-Learn to perform a grid search (Pedregosa et al., 2013) to tune the hyper-parameters, by which we could find the best parameters to evaluate the system. The weight indicates the weight_constraint. The LeCun indicates LeCun uniform. The fine-tuned parameters are summarized in Table 2.

3.3 Evaluation Metrics

Classification performance of the submissions will be evaluated based on binary F_1-score for the positive class. Binary F_1-score will range from 1 to 0.

3.4 Results and Discussion

The trial and test data for the baseline model and the BERT model shows that our model has the best score in Table 1.

Subtask A. Our system achieved the F1 score of 0.7353 on Subtask A, and the baseline score was 0.2676. The results show that our proposed sys-

tem is a significant improvement over the baseline. The main reason is that not only the BERT is a multi-layer bidirectional transformer encoder, but also the BERT-Base model is powerful pre-training model.

Subtask B. Our model score was 0.5035, while the baseline score was 0.7329. There is a need to do more for improvement. For cross-domain suggestion mining, it is necessary to increase the generalization ability of the training model to achieve use in multiple domains.

4 Conclusion

In this paper, we describe a task system that we submitted to SemEval-2019 for suggestion mining. For Subtask A, we use the BERT model. For Subtask B, we introduced CNN combined with BiLSTM and GRU. The experimental results show that the models we introduced achieved good performance in the final evaluation phase. In future research, we will attempt to generalize models with better capabilities to obtain more better results.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) under Grants No.61702443 and No.61762091, and in part by Educational Commission of Yunnan Province of China under Grant No.2017ZZX030. The authors would like to thank the anonymous reviewers and the area chairs for their constructive comments.

References

Raymond Brueckner and Bjrn Schuller. 2014. Social signal classification using deep blstm recurrent neural networks. In *IEEE International Conference on Acoustics*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Y. Le Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. 1990. Handwritten digit recognition with a back-propagation network. *Advances in Neural Information Processing Systems*, 2(2):396–404.

Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Alex Graves. 2012. *Long Short-Term Memory*.

K Greff, R. K. Srivastava, J Koutnik, B. R. Steunebrink, and J Schmidhuber. 2016. Lstm: A search space odyssey. *IEEE Transactions on Neural Networks & Learning Systems*, 28(10):2222–2232.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Yang Yin Lee, Hao Ke, Hen Hsen Huang, and Hsin Hsi Chen. 2016. Less is more: Filtering abnormal dimensions in glove. In *International Conference Companion on World Wide Web*.

Tianshi Li and Baobao Chang. 2015. Semantic role labeling using recursive neural network.

Tsung Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. 2017. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, PP(99):2999–3007.

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *International Conference on International Conference on Machine Learning*.

Sapna Negi, Tobias Daudert, and Paul Buitelaar. 2019. Semeval-2019 task 9: Suggestion mining from online reviews and forums. In *Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019)*.

Fabian Pedregosa, Gal Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, and Vincent Dubourg. 2013. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(10):2825–2830.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Cao Yuan, Gao Qin, and Klaus Macherey. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation.

Mohamed A. Zahran, Ahmed Magooda, Ashraf Y. Mahgoub, Hazem Raafat, Mohsen Rashwan, and Amir Atyia. 2015. *Word Representations in Vector Space and their Applications for Arabic*.