YNU-HPCC at SemEval-2022 Task 4: Finetuning Pretrained Language Models for Patronizing and Condescending Language Detection

Wenqiang Bai, Jin Wang and Xuejie Zhang
School of Information Science and Engineering
Yunnan University
Kunming, China
Contact: ynubwq@mail.ynu.edu.cn, {wangjin, xjzhang}@ynu.edu.cn

Abstract

This paper describes a system built for the SemEval-2022 competition. As participants in Task 4: Patronizing and Condescending Language Detection, we implemented the text sentiment classification system for two subtasks in English. Both subtasks involve determining emotions; subtask 1 requires us to determine whether the text belongs to the PCL category (single-label classification), and subtask 2 requires us to determine to which PCL category the text belongs (multi-label classification). Our system is based on the bidirectional encoder representations from transformers (BERT) model. For the single-label classification, our system applies a BertForSequenceClassification model to classify the input text. For the multi-label classification, we use the fine-tuned BERT model to extract the sentiment score of the text and a fully connected layer to classify the text into the PCL categories. Our system achieved relatively good results on the competition's official leaderboard.

1 Introduction

Text classification is an area of natural language processing (NLP) that aims to classify text using certain features. Previous studies on text classification tasks used traditional machine learning methods, which require researchers to manually design features. Feature extraction methods such as term frequency–inverse document frequency (TF-IDF) (Hakim et al., 2014) and N-Gram (Cavnar et al., 1994) are used to extract features from original documents, and then the features are input into classifiers such as naive Bayes(Berrar, 2019), support vector machines (SVMs) (Hearst et al., 1998), and decision trees (Vens et al., 2008). Since the advent of deep learning, text classification tasks are achievable without manual extraction of text features. Researchers must simply pretreat the text and incorporate it into a deep learning model for training. For text classification using deep learning methods, the classification accuracy is often higher than that of traditional machine learning methods. With their continuous improvement, deep learning models, such as recurrent neural networks (RNNs)(Zaremba et al., 2014), multi-channel CNN-LSTM (Zhang et al., 2017),gate recurrent units (GRUs) (Rana, 2016), long short-term memory (LSTM) (Shi et al., 2015), bidirectional long short-term memory (Bi-LSTM) (Zhang et al., 2015), and attention-based Bi-LSTM (Zhang et al., 2018) networks, can be used to solve text classification problems. In recent years, bidirectional encoder representations from transformers (BERT) (Devlin et al., 2018), a new deep learning model, has achieved, or even surpassed, human performance in multiple tasks within the NLP domain, including text classification.

Task 4 of the SemEval-2022 consists of the following two subtasks.

- Subtask 1: identifying whether the sentence contains any kind of PCL.
- Subtask 2: identifying which types of PCL the sentence contains.

In this paper, we introduce a deep learning system for SemEval-2022 Task 4: Patronizing and Condescending Language Detection (Pérez-Almendros et al., 2022). We applied the pretrained BERT model as the base model. This task contains two subtasks: single-label classification and multi-label classification. To accomplish both subtasks, we used fine-tuning methods on the base model with an additional classification layer. Our contributions are as follows:

- For the sentiment analysis task, we used the pretrained BERT model as the base model.
- To obtain the classification results, we added a fully connected layer at the end of the base model.

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 454 - 458
July 14-15, 2022 ©2022 Association for Computational Linguistics

The remainder of this paper is organized as follows. Section 2 provides an overview of our system for the two subtasks. Section 3 presents the specific details of our system. Section 4 discusses the results of the experiments, and finally, we draw our conclusions in Section 5.

2 Overview

This section presents an overview of our system and experiments, consisting of the following steps:

1. The data processing step, in which we use text processing tools to clean the text content, such as removing HTML tags in the text.

2. The model training step, in which we build, train, and evaluate the model.

3. The result generating step, in which we evaluate the model and predict the results on the test dataset.

Task description. The two subtasks involved text sentiment analysis and classification. The difference between them is that subtask 1 only requires us to determine whether the text contains any kind of PCL. Subtask 2 is the multi-label classification task, and the data of subtask 2 are marked by a list of 0s and 1s, which indicate the type of linguistic techniques (Unbalanced_power_relations, Shallow_solution, Presupposition, Authority_voice, Metaphors, Compassion, The_poorer_the_merrier) used to express condescension.

2.1 Data processing

To use the original text as much as possible and reduce the impact of meaningless text on the model, we built text cleaning tools that can be used to remove redundant text from the original. In addition, to complete the text classification task, a special token is added to the front of the original sentence.
Preprocessing. The texts may have been retrieved from the Internet by an automated program and inevitably there will be some unnatural language in the text. Text processing tools, such as regular expressions and Beautiful Soup, are used to remove impurities, such as HTML tags and redundant punctuation, from the text. Because the original sentence cannot be used in the pretrained BERT model, a special token **[CLS]** is added to the front of the sentence, and the model receives the new sequence (with the added token) as input.

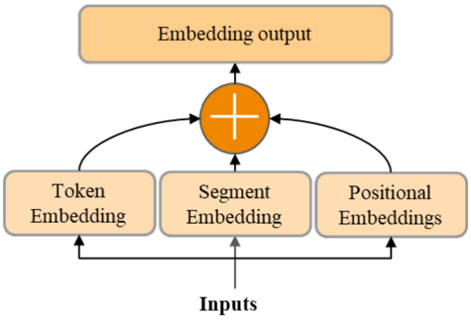

Figure 1: Embedding blocks

2.2 Deep learning models

In recent years, the use of deep learning for NLP text classification has become the most commonly adopted method in the industry. We used the pretrained BERT model to accomplish the tasks mentioned in the task description.
Bidirectional Encoder Representations from Transformers (BERT). As the name suggests, the BERT model is the encoder of the bidirectional transformer. BERT uses masked LM and next-sentence prediction to capture the representation at the word and sentence levels, respectively, and pretrains the model in a self-supervised manner.

Since the BERT model was proposed by Google in 2018, the entire field of NLP has entered a new stage. With BERT, we can easily fine-tune a pretrained model to achieve outstanding results that may even surpass human performance.

BERT consists of two main blocks: the embedding block and transformer encoder block, whose details are as follows.

1. **Embedding Block.** After preprocessing the original text, the output is fed to the embedding block, whose structure is shown in Figure 1.

   The embedding block has three embedding layers: the Token Embeddings, which convert each word into a fixed-dimensional vector similar to most deep learning models; Segment Embeddings, which distinguish between the two sentences; and Position Embeddings, which represent the position of each word in the sentence. These embedding layers transform the input text into a three-dimensional matrix $X \in R^{N \times n \times d}$, where N is the number of sentences in the text, n is the number of words in the sentence, and d is the dimension of the embedding vector.

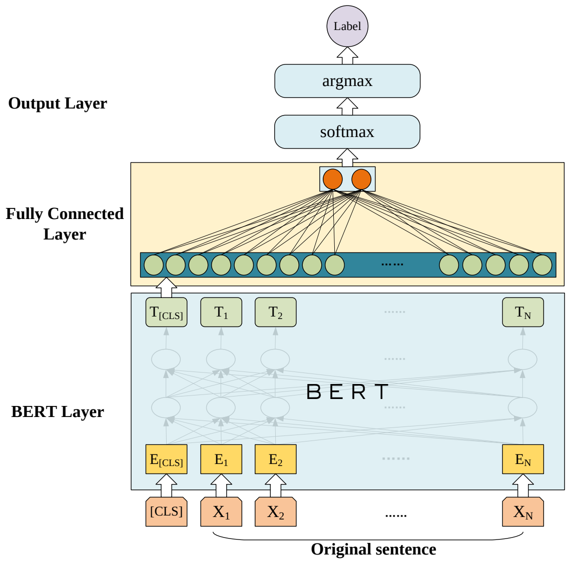

Figure 2: Single-label classification system

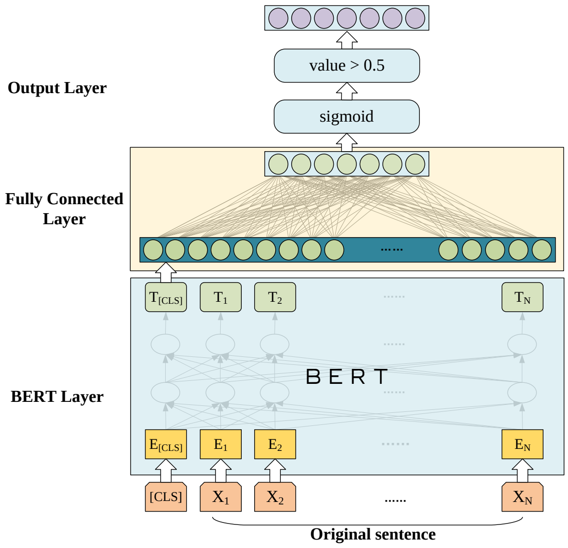

Figure 3: Multi-label classification system

2. **Encoder Block.** The encoder block comprises a series of transformer encoder blocks. Each transformer encoder block comprises two layers: the multi-head self-attention and feed-forward layers. The self-attention layer included in the encoder block of the transformer allows each word in the sentence to use the information of all other words in the sentence. The output of the current word does not need to depend on the output of the previous word, making the training well parallelized and greatly reducing the time to train the model. Because each word has a different impact on the sentence category, the attention mechanism can dynamically change the weight of each word.

3 Model Description

A pretrained BERT model is used to accomplish both subtasks with the two independent datasets. The details of the model built for these two subtasks are as follows.

3.1 Subtask 1: single-label classification

The architecture of the system built for subtask 1 has three different layers, as shown in Figure 2.

3.2 Subtask 2: multi-label classification

The system built for subtask 2 is similar to that for subtask 1, and the architecture of this system is only slightly different in the output layer. The structure is shown in Figure 3.

3.3 Details of the model architecture

BERT Layer. After preprocessing, the texts are input into the z BERT model, which contains the embedding and encoder blocks. Each word in the input sequence will output a fixed-dimensional (d) vector. In our BERT model (bert-based-uncased), d is 768.

Fully Connected Layer. The fully connected layer is used to convert a d-dimensional vector into a vector with the number of categories or labels as the dimension. In the text classification task, only the output of the first word, which is **[CLS]** at the BERT layer, is fed to the fully connected layer because it integrates the semantic information.

Output Layer. A matrix $X \in R^{N \times c}$ is output by the fully connected layer, in which N is the number of sentences and c is the number we manually set. In the single-label two-category classification task, it is set to 2, and the fully connected layer converts the 768-dimensional vector into a 2-dimensional vector. In the multi-label two-category classification task, it is set to the number of labels, 7, and the fully connected layer converts the 768-dimensional vector into a 7-dimensional vector.

To obtain the final result for the single-label classification task, the output of the fully connected layer is input into the *softmax* function to calculate the probability of the sentence belonging to the category, and the outcomes of the *softmax* function are fed to the *argmax* function to obtain the classification result.

$$class = \begin{cases} 0, value_0 \geq value_1, \\ 1, value_0 < value_1 \end{cases} \quad (1)$$

If the output value is 1, the sentence belongs to the label, that is, this sentence contains some kind of PCL; otherwise, the sentence does not contain any kind of PCL.

For the multi-label classification task, we input the result of the fully connected layer into a *sigmoid* function that maps each value in the output vector to a value between 0 and 1. Each value in the vector is then mapped to 0 or 1 according to the rounding rules.

$$label_i = \begin{cases} 0, label_i \leq 0.5, \\ 1, label_i > 0.5. \end{cases} \quad (2)$$

The output is a 7-dimensional vector that consists of 0 or 1. If the value is 1, the sentence used the technique corresponding to the vector element number to express the condescension.

3.4 Training and Hyperparameters

For these two classification tasks, we used the BCEwithLogits loss function and Adam (Kingma and Ba, 2017) optimizer to train both models. Both models use a stochastic gradient with mini-batches of size 16. The hyperparameters are as follows:

Hyperparameters The maximum input sequence length of the BERT model is 512, the dimension of word embeddings (d) is 768, the dropout ratio is 0.1 at each layer in the models, the learning rate is 1e-5, and the number of epochs is 15.

4 Experiment

Dataset. For the two subtasks, the corpus we used to train the model are from the competition(Pérez-Almendros et al., 2020), without other external data.

dontpatronizeme_pcl.tsv This dataset contains 10,469 paragraphs, and each paragraph is annotated with a label ranging from 0 to 4. In the single-label classification subtask, the original label annotated as either 0 or 1 is replaced with 0, and the other labels with 1.

dontpatronizeme_categories.tsv This dataset contains 993 unique paragraphs with a total of 2,760 instances of PCL. In the multi-label classification task, each paragraph is annotated with 7 labels ranging from 0 to 1.

Table 1: Subtask 1 result

Precision	Recall	F1_Score
0.5097	0.4132	0.4564

Table 2: Subtask 2 result

Label	Score
Unbalanced_Power_Relations	0.1600
Shallow_Solution	0.1245
Presupposition	0.0721
Authority_Voice	0.0968
Metaphor	0.0696
Compassion	0.1139
The_poorer_the_merrier	0.0385
Average	0.0965

Evaluation Methods. For subtask 1 (single-label classification), the competition metrics given by the competition organizer are precision, recall, and F1 score. For subtask 2 (multi-label classification), there are two competition metrics: prediction accuracy of each label and average prediction accuracy of all labels.

Results. The results of the two subtasks are shown in Tables 1 and 2.

For subtask 1, we ranked 42/81 in precision, 47/81 in recall, and 52/81 in F1 score.

For subtask 2, we ranked 35, 33, 35, 34, 33, 34, and 24 out of 81 for the seven labels: Unbalanced_power_relations, Shallow_solution, Presupposition, Authority_voice, Metaphors, Compassion, and The_poorer_the_merrier, respectively.

Experiments and Analysis. We used 80% of the training data as the training set and 20% of the training data as the validation set. We trained our model on the training set and used the validation set to evaluate the accuracy of the model. Our system achieved relatively good results on the competition's official leaderboard, which is inseparable from the excellence of the pretrained BERT model. The outstanding advantage of the pretrained model is that it can learn the language from a large amount of unlabeled data and then fine-tune on a small amount of labeled data. Thus, downstream tasks often lead to better learning of language and task-specific features.. Compared to traditional RNN and LSTM models, BERT can perform concurrently and simultaneously extract relational features of words in a sentence at several different levels, thus comprehensively reflecting the sentence semantics. Compared to word2vec, the meanings of words can also be obtained according to the context of the sentence, which would avoid ambiguity.

5 Conclusion

In this paper, we described our system, which is based on the pretrained BERT model, for the text classification task SemEval 2022 Task 4: Patronizing and Condescending Language Detection. We added a classification layer to the pretrained BERT model to address both subtasks. The results generated by the proposed system achieved a relatively good ranking. In the future, we hope to explore other models and methods in the sentiment analysis field.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (NSFC) under Grants No. 61966038. The authors would like to thank the anonymous reviewers for their constructive comments.

References

D. Berrar. 2019. Bayes' theorem and naive bayes classifier - sciencedirect. *Encyclopedia of Bioinformatics and Computational Biology*, 1:403–412.

William Cavnar, , William B. Cavnar, and John M. Trenkle. 1994. N-gram-based text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova Google, and A I Language. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ari Aulia Hakim, Alva Erwin, Kho I Eng, Maulahikmah Galinium, and Wahyu Muliady. 2014. Automated document classification for news article in bahasa indonesia based on term frequency inverse document frequency (tf-idf) approach. In *Proceedings of 6th International Conference on Information Technology and Electrical Engineering (ICITEE)*, pages 1–4.

M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. 1998. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Carla Pérez-Almendros, Luis Espinosa-Anke, and Steven Schockaert. 2020. Don't Patronize Me! An Annotated Dataset with Patronizing and Condescending Language towards Vulnerable Communities. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5891–5902.

Carla Pérez-Almendros, Luis Espinosa-Anke, and Steven Schockaert. 2022. SemEval-2022 Task 4: Patronizing and Condescending Language Detection. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics.

Rajib Rana. 2016. Gated recurrent unit (gru) for emotion classification from noisy speech. *arXiv preprint arXiv:1612.07778*.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, Wang-Chun Woo, and Hong Kong Observatory. 2015. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*, volume 28, pages 802–810. Curran Associates, Inc.

Celine Vens, Jan Struyf, Leander Schietgat, Sašo Džeroski, Hendrik Blockeel, C Vens, J Struyf, L Schietgat, H Blockeel, and S Džeroski. 2008. Decision trees for hierarchical multi-label classification. *Machine Learning 2008 73:2*, 73:185–214.

Wojciech Zaremba, Ilya Sutskever, Oriol Vinyals, and Google Brain. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

Haowei Zhang, Jin Wang, Jixian Zhang, and Xuejie Zhang. 2017. YNU-HPCC at SemEval 2017 task 4: Using a multi-channel CNN-LSTM model for sentiment classification. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 796–801, Vancouver, Canada. Association for Computational Linguistics.

Shu Zhang, Dequan Zheng, Xinchen Hu, and Ming Yang. 2015. Bidirectional long short-term memory networks for relation classification. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*, pages 73–78, Shanghai, China.

You Zhang, Jin Wang, and Xuejie Zhang. 2018. YNU-HPCC at SemEval-2018 task 1: BiLSTM with attention based sentiment analysis for affect in tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 273–278, New Orleans, Louisiana. Association for Computational Linguistics.