Sapphire at SemEval-2022 Task 4: A Patronizing and Condescending Language Detection Model Based on Capsule Networks

Sihui Li
Yunnan University, Yunnan, P.R. China
sihui_li@mail.ynu.edu.cn

Xiaobing Zhou *
Yunnan University, Yunnan, P.R. China
zhouxb@ynu.edu.cn

Abstract

This paper introduces the related work and the results of Team Sapphire's system for SemEval-2022 Task 4: Patronizing and Condescending Language Detection. We only participated in subtask 1. The task goal is to judge whether a news text contains PCL. This task can be considered as a task of binary classification of news texts. In this binary classification task, the BERT-base model is adopted as the pre-trained model used to represent textual information in vector form and encode it. Capsule networks is adopted to extract features from the encoded vectors. The official evaluation metric for subtask 1 is the F1 score over the positive class. Finally, our system's submitted prediction results on test set achieved the score of 0.5187.

1 Introduction

Patronizing and Condescending Language (PCL) can be considered when someone's language has a superior attitude towards others, demeans others, or describes the situation of others in a compassionate way. Such expressions are often unconscious, and are used by people to try to induce action or raise awareness. Because of its subtlety and often well-meaning when used, users often overlook the demeaning elements of this expression. Such elements may contribute to the stereotyped influence of society on a group, making discrimination normalized and even leading to stronger exclusion (Pérez-Almendros et al., 2022).

Detecting PCL in media text is a challenging task. Recognizing PCL based on Natural Language Processing (NLP) can alert speakers to examine the rationality of their speeches, so that speeches can be more inclusive and constructive, which in turn leads to more responsible communication.

When processing corpus, the pre-trained model can convert text information into vector representation, making it more suitable for NLP tasks. Early pre-trained models were designed to learn representational word embeddings, such as Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014). Although such methods can capture the semantics of words through word embeddings, they cannot capture the concepts in the context. With the introduction of new technologies, there are now pre-trained models that can learn to represent contextual word embeddings, such as the ELMo (Peters et al., 2018) model based on LSTM (Shi et al., 2015) and the BERT (Devlin et al., 2018) model based on Transformer Encoder (Vaswani et al., 2017).

In recent years, using deep neural networks in NLP, such as Convolutional Neural Networks (CNNs) in text classification (Kim, 2014), has become mainstream. Capsule networks (Sabour et al., 2017), as a structure proposed on the basis of CNNs to improve spatial sensitivity in computer vision, is also used in text classification tasks (Yang et al., 2019; Ding et al., 2019). Kim et al. (2020) further suggest a simple routing method that effectively reduces the computational complexity of dynamic routing.

This task aims to predict whether each news text for each ID contains PCL. Text is represented as a vector and encoded using a pre-trained BERT model. It mainly uses the capsule networks to extract the encoded vector, and uses the output of the fully connected layer to represent the label probability. The rest of the paper is organized as follows: Section 2 introduces the system architecture. Section 3 describes the dataset, implementation details settings and experimental results. The summary and outlook for future work will be presented in Section 4.

2 System Architecture

In this section, we will introduce the system architecture we use in the task, which will consist of two parts. One is embedding and coding, and the other is feature extraction and prediction. We call the model that is ultimately used to test the class pre-

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 405 - 408
July 14-15, 2022 ©2022 Association for Computational Linguistics

diction results of the dataset as BERT-Caps. The architecture of BERT-Caps model is shown in Figure 1.

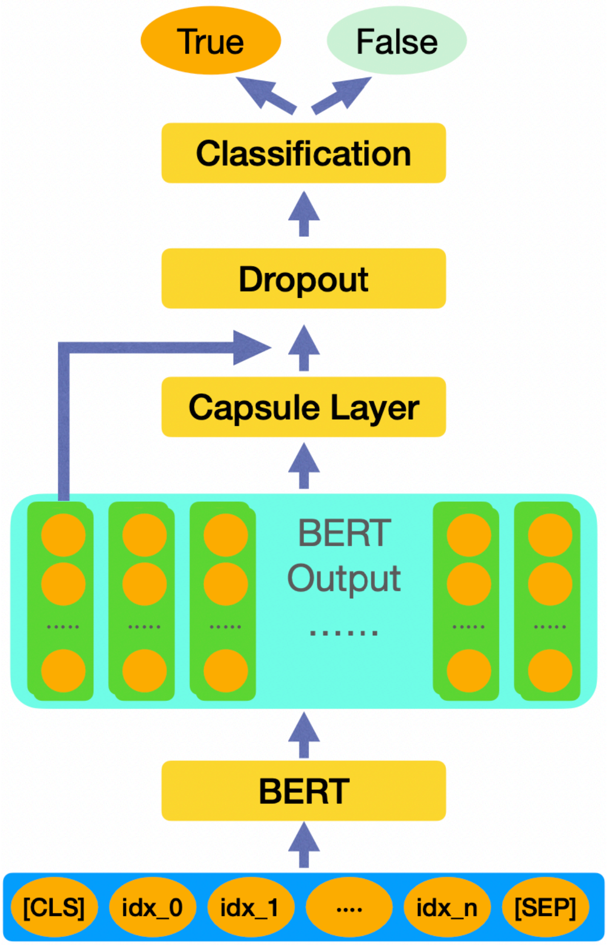

Figure 1: The architecture of the BERT-Caps model.

2.1 Embedding and Encoding

When using the early pre-trainied model for text classification, the text is usually represented as a word embedding and then the vector matrix is sent to a bidirectional recurrent network for encoding to improve the system's ability to perceive contextual information. In this paper, we mainly use the BERT-base model to represent text into vector form and encode it.

In order to fit the pre-trained model, we need to preprocess the text in the dataset accordingly. As standard news text, we do only a little text processing on the news text: unify the text to lowercase. Add markers to the beginning and end of the text. For example, when using BERT as the pre-trained model, [CLS] and [SEP] will be used to mark the beginning and end of the text, respectively. Then according to the dictionary information, the words are converted into a list of their position numbers in the dictionary. Collate to get a list that marks the beginning and the end of each sentence.

This part of the work is mainly achieved through the tokenizer attached to the module used when importing the pre-trained model. For the imported model, we set the trainable value of each layer of the model to True.

2.2 Feature Extraction and Predict

We use the capsule networks to perform feature extraction on the hidden state of the last layer of the pre-trained model. In the capsule layer, the input is firstly processed by the Conv1d function. The convolution output is treated as a set of capsules, and a new set of capsules of the specified shape is derived through the dynamic routing algorithm. The result is the output of the capsule layer.

The flattened capsule layer output and the text vector corresponding to the first bit in the pre-trained model are linked together. In order to improve the generalization ability of the model, dropout is used. During training, the concatenated outputs are first processed by dropout and then fed into the fully connected layer to predict the probability that the news text has PCL. The loss function of this model adopts categorical crossentropy.

3 Experiment and Result

3.1 Dataset and Official Evaluation Metrics

The dataset used in the experiment is provided by SemEval-2022 Task4, Patronizing and Condescending Language Detection.(Pérez-Almendros et al., 2020)

In this dataset, the degree of PCL is divided into five levels from 0 to 4. In subtask 1, the level of 0-1 is regarded as a negative example, and 2-4 is regarded as a positive example. Participants were asked to predict the presence or absence of PCL component in the text. The differentiated test set contains 9476 negative labels and 993 positive labels, almost reaching 10:1. Due to the imbalance of samples in the dataset, the F1 score over the positive class was adopted as the official evaluation metric.The formula for F1 score is as following:

$$F1 = 2 * \frac{precision * recall}{precision + recall} \tag{1}$$

Precision means the ratio of correctly predicted positive observations to the total predicted positive observations. Recall means the ratio of correctly predicted positive observations to all observations in the real class.

3.2 Implementation Details

In terms of data segmentation, we import the train_test_split function from the Scikit-learn(Pedregosa et al., 2011) module to divide the dataset into training set and validation set, set *test_size* to 0.2, *random_state* to 35.

All experiments in this paper are based on using the TensorFlow2 backend.

When using BERT-base-uncased[1] as the pre-trained model, we use the Keras-BERT (Shorten and Khoshgoftaar, 2021) module to implement the Tokenizer and import the model.

We also tried other BERT-based models, such as RoBERTa-base and DeBERTa-base. When implementing Tokenizer and importing models, we use the Transformers (Wolf et al., 2020) module.

The number of capsules, the number of hidden neurons, and the number of iterations of the dynamic routing algorithm are set to 10, 64, and 3, respectively.

The fully connected layer that outputs the final result in each model uses softmax as the activation function. The hyperparameters used are mentioned in Table1.

Parameters	**subtask 1**
Epochs	8
Batch_size	8
Max_length	128
Drop_rate	0.25
Optimizer	Adam
Initial *lr*	1e-5

Table 1: Hyperparameters

In actual training, in order to alleviate the overfitting situation, ReduceLROnPlateau is introduced. Also set ModelCheckpoint to save each model with the smallest loss on the existing basis.

3.3 Experiment and Result

The system uses the dataset provided by the task organizer for training. The BERT-Caps model that finally gets the submitted prediction results is saved at the end of the 8th epoch training.

The results are shown in Table 2. The values of RoBERTa_baseline comes from the result published on the Competition Page[2]. As can be seen from the table, as a result, our model has a greater improvement in precision than the baseline. We also tried to train some BERT-Caps models that reduced the number of capsules in the capsule layer and increased the number of hidden neurons, but there was no significant improvement in metric.

Table 2 shows that without the capsule networks, the performance of the model will be greatly reduced compared to the original model. Without dropout, the prediction performance decreases less than without the capsule networks.

We also tried to keep almost the same system architecture, only replacing the pre-trained model and tokenizer. Unexpectedly, in the experimental environment of this paper, both DeBERTa-Caps and RoBERTa-Caps are not as good as BERT-Caps.

The best test set predictions submitted by our team were produced by the BERT-Caps model. Considering with the F1 scores obtained by the top four teams in the English data are all over 0.6400, indeed, there is a gap. Team Sapphire's final ranking is 35th.

4 Conclusion

This paper describes the experiments conducted by Team Sapphire in subtask 1 of SemEval-2022 Task 4: Patronizing and Condescending Language Detection. We introduced the system architecture, experimental dataset situation and results in Section 2 and 3, respectively. From the experimental results, the BERT-Caps model can achieve better results on the test set. In future work, we will improve our method to achieve better results. For example, using other text representations, and adjusting the weight of the loss function.

References

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

Yunxia Ding, Xiaobing Zhou, and Xuejie Zhang. 2019. Ynu_dyx at semeval-2019 task 5: A stacked bigru model based on capsule network in detection of hate. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 535–539.

[1] https://storage.googleapis.com/bert_models/2018_10_18/uncased_L-12_H-768_A-12.zip

[2] https://sites.google.com/view/pcl-detection-semeval2022/ranking

Model	Precision	Recall	F1 score
BERT-Caps	0.5935	0.4606	0.5187
BERT-Caps w/o dropout	0.6	0.4542	0.5170
BERT-Caps w/o capsule networks	0.5782	0.4195	0.4862
RoBERTa-Caps	0.625	0.3943	0.4835
DeBERTa-Caps	0.5870	0.4574	0.5141
RoBERTa_baseline	0.3935	0.6530	0.4911

Table 2: Results: subtask 1

Jaeyoung Kim, Sion Jang, Eunjeong Park, and Sungchul Choi. 2020. Text classification using capsules. *Neurocomputing*, 376:214–221.

Y. Kim. 2014. Convolutional neural networks for sentence classification. *Eprint Arxiv*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

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.

Matthew Peters, M. Neumann, M. Iyyer, M. Gardner, and L. Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. *Advances in neural information processing systems*, 30.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. 2015. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28.

Connor Shorten and Taghi M Khoshgoftaar. 2021. Kerasbert: Modeling the keras language. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 219–226. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Min Yang, Wei Zhao, Lei Chen, Qiang Qu, Zhou Zhao, and Ying Shen. 2019. Investigating the transferring capability of capsule networks for text classification. *Neural Networks*, 118:247–261.