Top3 Solution of Shanghai-HK Interdisciplinary Shared Tasks (2022) Track 1 : Trigger Identification

Peng Xia
Soochow University
richardxp888@gmail.com

Abstract

In this paper, we describe the solution to the Shanghai-HK Interdisciplinary Shared Tasks (2022) Track 1. We use the BigBird model for the message representation and mean pooling to propagate the classification model of tree representation. After data cleansing, we try to select a pretraining model with better performance and improve the network structure. Our full pipeline, with only single models, scores 1.313 and 1.264 on the leaderboard, which we achieved the 1st place in the phase 1 and 3rd place in the phase 2 respectively.

1 Introduction

Nowadays, rumors tend to spread quickly and widely on the Internet, and automatically verifying rumors has become an urgent need for individuals and society. From social platforms (such as Twitter and Weibo), we can crawl information cascades that consist of source posts and corresponding reposts. The task of rumor verification aims at classifying rumor cascade as true, false or unverified. However, predicting at cascade level is too coarse to organize massive messages and opinions. Therefore, Fudan DISC Lab supplement message-level annotations to exiting rumor corpus PHEME and propose a sub-task named trigger identification, which aims at identifying messages that have prominent effects on rumor proliferation and dominate the judgment of cascade credibility. We summarize message roles into 4 categories, i.e. amplify, deny, clarify and null. Amplify indicates tweets that initiate new concerns or enlarge the discussion scale related to the social event. Deny means presenting doubt or rejection towards previous messages. Clarify introduces factual or substantial information. Other messages are left as null which means they are insignificant for rumor propagation or verification.

1.1 Dataset

The extended dataset contains 1,929 cascades and 26,871 messages annotated. Train, validation, test (phase 1) and test (phase 2) sets are split randomly with a proportion of 7:1:1:1.

1.2 Evaluation metric

We adopt macro F1 score (Opitz J and Burst S, 2019) as evaluation metrics.

2 Methodology

2.1 Data cleansing

Due that texts are collected from the web, there are a lot of abbreviations in the dataset. So firstly we convert the abbreviations in the text to the full version. And then url, html tags and @ are removed. In addition, to make the text cleaner, special characters and numbers between null characters are removed as well.

2.2 Modeling

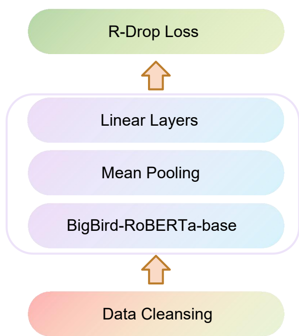

Figure 1: Overview of our proposed framework

Figure 1 gives an overview of our approach. As shown in the Figure, we regard the task as a text classification task. We use a sparse-attention transformer (Vaswani A, et al, 2017) encoder called BigBird (Zaheer M, et al, 2020) as the backbone to obtain message representation. Finally, we apply mean pooling to representations from encoder to acquire cascade representation.

BigBird, is a sparse-attention based transformer which extends Transformer based models, such as BERT (Devlin J, et al, 2018) to much longer sequences. Moreover, BigBird comes along with a theoretical understanding of the capabilities of a complete transformer that the sparse model can handle. It is a pretrained model on English language using a masked language modeling (MLM) objective. We use it through Transformers[1].

2.3 Network improvement

Initially, we try to add an extra linear layer to the network, which makes small improvements. Moreover, we focus on the loss function. R-Drop (Wu L, et al, 2021) has proven to be a simple and efficient regularization technique based on Dropout (Srivastava N, et al, 2014). In order to better adapt to the current task, we try to replace kL-Divergence loss with MSE loss, and control the proportion of task learning loss and R-Drop loss through a parameter. On this basis, we also explore the influence of different forward times on model performance. The formulas are shown below.

$$L_R^i = MSE\left(y_1^i, y_2^i\right) \tag{1}$$

$$L_B^i = \frac{1}{2} \cdot \left(MSE\left(y_1^i, \hat{y}^i\right) + MSE\left(y_2^i, \hat{y}^i\right)\right) \tag{2}$$

$$L^i = \alpha \cdot L_R^i + (1-\alpha) \cdot L_B^i \tag{3}$$

Where y_1^i and y_2^i are the predicted values of the two forward of the sample, $\hat{y}^i$ is the true value of the sample, and α controls the relative intensity of the two losses.

3 Experiment

Actually we did not record the scores of each experiment, only remember the general changes. So please refer only to our conclusions, not to the data.

[1] `https://huggingface.co/google/bigbird-roberta-base`.

System	macro F1 score
Baseline	1.194
RoBERTa	1.215
BigBird	**1.263**
+Extra Linear Layers	1.268
+R-Drop	**1.313**

Table 1: Performance of different experiments

4 Future prospects

Due to our own reasons, there are a lot of things we haven't tried, such as improving the pooling layer. In addition, from the perspective of data, we did not explore the data itself too much. Methods like data augmentation can be tried to apply the task. Finally, we think the task is interesting and meaningful, and hope that more people can focus on it and there will be more good work as well in the future.

References

Zaheer M, Guruganesh G, Dubey K A, et al. 2020. Big bird: Transformers for longer sequences *Advances in Neural Information Processing Systems*, 33: 17283-17297.

Opitz J, Burst S 2019. Macro f1 and macro f1 *arXiv preprint*, arXiv:1911.03347.

Vaswani A, Shazeer N, Parmar N, et al. 2017. Attention is all you need *Advances in Neural Information Processing Systems*, 30.

Wu L, Li J, Wang Y, et al. 2021. R-drop: regularized dropout for neural networks *Advances in Neural Information Processing Systems*, 34.

Srivastava N, Hinton G, Krizhevsky A, et al. 2014. Dropout: a simple way to prevent neural networks from overfitting *The journal of machine learning research*, 15(1): 1929-1958.

Devlin J, Chang M W, Lee K, et al. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding *arXiv preprint*, arXiv:1810.04805.