Zhegu@SMM4H-2022: The Pre-training Tweet & Claim Matching Makes Your Prediction Better

Pan He[1], YuZe Chen[3], Yanru Zhang[1,2]*
[1]University of Electronic Science and Technology of China, Chengdu
[2]Shenzhen Institute for Advanced Study, UESTC
[3]Chengdu University of Technology
newtonysls@gmail.com
chen.yuze@student.zy.cdut.edu.cn
yanruzhang@uestc.edu.cn

Abstract

SMM4H-2022 (Weissenbacher et al., 2022) Task 2 is to detect whether containing premise in the tweets of users about COVID-19 on the social medias or their stances for the claims. In this paper, we propose **T**weet **C**laim **M**atching (**TCM**), which is a new pre-training task constructed by the tweets and claims similarly to Next Sentence Prediction (NSP). We first continue to pre-train the standard pre-trained language models on the labelled dataset and then fine-tune them for obtaining better performance. Compared with the solid baseline (Glandt et al., 2021), we achieve the absolute improvement of 7.9% in Task 2a and obtain the SOTA results.

1 Introduction

Since its appearance in 2019, COVID-19 has generated a wide range of discussions on the social medias. Users make statements that may contain their stances and arguments, and these Twitter posts are value for estimating the level of cooperation with the mandates. Research on this information has attracted widespread attention, and various automated approaches based on machine learning have been proposed. In this paper, we present the solution of our team Zhegu for identifying the stance and premise of Twitter posts in SMM4H-2022 Task 2. To summarize, our main contribution are:

- We constructed **TCM**. A new tweet & claim matching task for pre-training in SMM4H-2022 Task 2.
- Our system obtains the SOTA performance on Task2a of the stance prediction.

2 Data

All data sets (Davydova and Tutubalina, 2022) contain texts from Twitter about three health mandates related to the COVID-19 pandemic: Face Masks, Stay At Home Orders, and School closures. There are 3,556 of the training data, 600 for validation, and 2,000 for testing. Task2a needs to determine whether the attitude of the authors of the tweets are supported, against or neutral to the claims, which is a triple classification. Task2b is a bi-classification, which aims to figure out whether the tweets contain at least one premise or argument.

3 Methodology

In this section, we firstly present our model and strategies we use. Secondly, we give the basic idea and the construction of our **TCM** pre-training task.

3.1 Fine tune

We selected two pre-trained language models (PLMs) as our backbones: RoBERTa-large (Liu et al., 2019) and COVID-Twitter-BERT-v2 (Müller et al., 2020). They both have 24 hidden layers, with the hidden size of 1024. We also design various strategies to obtain the representation of sentences, including CLS, averaging the last layer of PLMs, attention computation for the last four layers and concatenating the CLS of the last four layers. The small amount of the training dataset and the large scale of backbone can easily lead to over-fitting. In this case, we also used FGSM (Goodfellow et al., 2014), R-Drop (Wu et al., 2021), Exponential Moving Average (EMA) to improve the generalization of our model.

3.2 TCM

It has been proved that continuously pre-training on the domain corpus can improve the performance of fine-tuning (Gururangan et al., 2020). We perform further pre-training on backbone based on the training dataset. BERT (Devlin et al., 2018) contains two pre-training tasks including Masked Language Model (MLM) and NSP, while RoBERTa (Liu et al., 2019) only has the MLM. Consequently, We pre-train the MLM for both PLMs. Given a data set

*Corresponding author

Proceedings of the 29th International Conference on Computational Linguistic, pages 38–41
October 12–17, 2022.

RoBERTa-large (ours)							
FGSM	EMA	R-Drop	MLM	TCM	Pseudo-label	Validation	Test
√	-	-	-	-	-	0.788	-
√	√	-	-	-	-	0.798	-
√	√	√	-	-	-	0.800	-
√	√	√	√	-	-	0.812	-
√	√	√	√	√	-	0.841 (+0.051)	-
COVID-Twitter-BERT-v2 (ours)							
√	√	√	-	-	-	0.853	-
√	√	√	√	√	-	0.868	-
√	√	√	√	√	√	**0.869** (+0.079)	0.634
Baseline (COVID-Twitter-BERT)						0.790	-

Table 1: The average F1 scores of various strategies in the validation and test dataset. We report the performance of baseline from (Glandt et al., 2021) directly. The bolded and underlined: the SOTA results of all methods. The underlined: the second best.

of the Task2a $((\boldsymbol{T}, \boldsymbol{C}), \boldsymbol{Y})$, where $\boldsymbol{T_i}$ denotes the tweet text, $\boldsymbol{C_i}$ represents the corresponding claim, and $\boldsymbol{Y_i}$ is the label. And $\boldsymbol{C_i} \in \mathcal{C} =$ {Face Masks, Stay At Home Orders, School Closures}. Naturally, $\boldsymbol{T_i}$ only corresponds to its unique $\boldsymbol{C_i}$ rather than $\mathcal{C} - \boldsymbol{C_i}$. This is a typical bi-classification as well as NSP. We construct the dataset of TCM by randomly sampling $\boldsymbol{C}^{\mathbf{sample}}$ for $\boldsymbol{T_i}$ from $\mathcal{C}$ in a certain probability of $p = 0.5$. $((\boldsymbol{T_i}, \boldsymbol{C}^{\mathbf{sample}}), \mathbf{1})$ is a positive sample if $\boldsymbol{C}^{\mathbf{sample}} = \boldsymbol{C_i}$, otherwise negative sample $((\boldsymbol{T_i}, \boldsymbol{C}^{\mathbf{sample}}), \mathbf{0})$. This TCM task incorporates the idea of contrastive learning, which enables the PLMs to determine whether the tweet matches the claim, thus improving the performance of the model.

4 Experiments

4.1 Experiments Settings

For fine-tuning, different learning rates are used for the backbone and other layers. The backbone is set to 2e-5 and the other layers are 1e-4. The epochs of the RoBERTa-large and COVID-Twitter-BERT-v2 are set to 11 and 4 respectively. Others detailed hyperparameters can be found in the Table 2. The evaluation metric we use is the average F1 score of the three claims, which is different from the website.

4.2 Main Results & Analysis

Table 1 presents the final results on the validation set and test set of Task2a. Task2b can be found in the Table 4. By comparing the effects of four different sentence representations, we finally adopted averaging the hidden states of the last layer of PLMs, as shown in the Table 3. The pre-training corpus of COVID-Twitter-BERT-v2 are the tweets related to COVID-19, while RoBERTa is pre-trained on the general corpus. Overall, the results of RoBERTa are worse 2.8% than COVID-Twitter-BERT-v2. However, RoBERTa still outperformed 5.1% compared to the baseline(Glandt et al., 2021), which takes COVID-Twitter-BERT (Müller et al., 2020) as the backbone. Meanwhile, FGSM, R-Drop and EMA bring different degrees of improvement. It brings relative small promotion when we merely pre-train the MLM task. The improvement is about 2.9% than the MLM only after adding our TCM task. Similarly, for COVID-Twitter-BERT-v2, the boost of 1.6% from the TCM and MLM task is also impressive. It is worth mentioning that we do not perform K-fold training, nor do we perform any model ensemble. The experimental results strongly prove that the TCM we have constructed is simple yet effective.

5 Conclusion

We present the solution of our team Zhegu in SMM4H-2022 Task 2. We construct a new TCM pre-training task by incorporating the idea of contrastive learning for the relationship between tweets and claims. The effectiveness of our TCM has been demonstrated by two PLMs. Even in RoBERTa with only the MLM, the addition of TCM brings a decent improvement. In summary, absolute improvement of 5.1% and 7.9% are achieved in both backbones compared to baseline. And we propose to investigate the relationship between Twitter text and its claim when fine-tuning in our future work.

References

Vera Davydova and Elena Tutubalina. 2022. Smm4h 2022 task 2: Dataset for stance and premise detection in tweets about health mandates related to covid-19. In *Proceedings of the Seventh Social Media Mining for Health Applications (#SMM4H) Workshop Shared Task*, pages –.

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*.

Kyle Glandt, Sarthak Khanal, Yingjie Li, Doina Caragea, and Cornelia Caragea. 2021. Stance detection in covid-19 tweets. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1596–1611.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Martin Müller, Marcel Salathé, and Per E Kummervold. 2020. Covid-twitter-bert: A natural language processing model to analyse covid-19 content on twitter. *arXiv preprint arXiv:2005.07503*.

Davy Weissenbacher, Ari Z. Klein, Luis Gascó, Darryl Estrada-Zavala, Martin Krallinger, Yuting Guo, Yao Ge, Abeed Sarker, Ana Lucia Schmidt, Raul Rodriguez-Esteban, Mathias Leddin, Arjun Magge, Juan M. Banda, Vera Davydova, Elena Tutubalina, and Graciela Gonzalez-Hernandez. 2022. Overview of the seventh social media mining for health applications smm4h shared tasks at coling 2022. In *Proceedings of the Seventh Social Media Mining for Health Applications (#SMM4H) Workshop Shared Task*, pages –.

Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, Tie-Yan Liu, et al. 2021. R-drop: Regularized dropout for neural networks. *Advances in Neural Information Processing Systems*, 34:10890–10905.

A Implement Details

All experiments were run on a RTX 3090 GPU with 24G of video memory. The rest of hyperparameters is shown in the Table 2. The main performance met-

FGSM	epsilon	0.5
EMA	start step	0
	decay	0.99
R-Drop	alpha	0.5
Max sequence length	128	
Batch size	16	
Pseudo label rate	0.88	
Optimizer	AdamW	
Adam epsilon	1e-6	
Weight decay	0.1	
Scheduler	linear warmup	
Warmup rate	0.1	
Max grad norm	1	

Table 2: The hyperparameters of various strategies for fine-tuning. The same parameters are used for both Task2a and Task2b. Besides, we directly apply the same pre-training hyperparameters of BERT (Devlin et al., 2018).

ric is calculated according to the following formula:

$$F1 = \frac{1}{3}\sum_{C}\frac{1}{2}(F_1^{macro} + F_1^{micro}) \quad (1)$$

where $C \in \mathcal{C}$. The evaluation metric is different from the official website, so the results we obtained are not the same as the official website. For the

Sentence representation	Validation
CLS	0.818
Mean	**0.841**
Attention	0.797
CLS-4	0.801

Table 3: CLS: the [CLS] representation. Mean: average the hidden states of the last layer. Attention: attention calculation of the hidden states of the last 4 layers. CLS-4: concatenate the [CLS] of the last 4 layers.

sentence representation, we evaluate four common methods in validation dataset of Task2a as it shown in Table 3. There is no the best representation which will always yield the best results for different downstream tasks. And we choose to average the last hidden states as the sentence representations for the model by comparing the results obtained from experiments under the same parameters.

B Another Results

Model	Validation
RoBERTa-large	0.820
COVID-Twitter-BERT-v2	0.831

Table 4: The final average F1 scores in the validation dataset of Task2b. It is worth noting that the results we present here obtained by using the same optimization strategy as Task2a.

Task	Validation (ours)	Test (ours)	Test (Median)	Test (Mean)
Task2a	0.869	0.634	0.550	0.491
Task2b	0.831	0.698	0.647	0.574

Table 5: The results of the test dataset. Median: the median score of all teams. Mean: the average score of all teams. Our results are excessively higher than the median and mean score in the test dataset of both Task2a and Task2b.