PingAnTech at SMM4H task1: Multiple pre-trained model approaches for Adverse Drug Reactions

Xi Liu[1], Han Zhou[1,2], Chang Su[1]
[1] PingAn Technology
[2] Chengdu University of Technology
doufuxixi3@163.com
zhouhan@stu.cdut.edu.cn
SHUCHUANG254@pingan.com.cn

Abstract

This paper describes the solution for the Social Media Mining for Health (SMM4H) 2022 Shared Task. We participated in Task1a., Task1b. and Task1c. To solve the problem of the presence of Twitter data, we used a pre-trained language model. We used training strategies that involved: adversarial training, head layer weighted fusion, etc., to improve the performance of the model. The experimental results show the effectiveness of our designed system. For task 1a, the system achieved an F1 score of 0.68; for task 1b Overlapping F1 score of 0.65 and a Strict F1 score of 0.49. Task 1c yields Overlapping F1 and Strict F1 scores of 0.36 and 0.30, respectively.

1 Introduction

Mining adverse drug events (ADEs) from social media is one of the most researched topics in the field of social media pharmacovigilance. To promote the research on this topic, the Health Language Processing Lab of the University of Pennsylvania organized Social Media Mining for Health Applications (SMM4H) shared tasks. This year, the SMM4H shared tasks included ten subtasks(Davy Weissenbacher, 2022). Our team focused on three subtasks in Task 1, which are (1) classifying tweets reporting ADEs (Adverse Drug Events); (2) detect ADE spans in the tweets; (3) map these colloquial mentions to their standard concept IDs in the MedDRA vocabulary. The main challenges of this task are as follows: (1) how to handle unbalanced data and (2) the presence of a large amount of noise in the data, e.g., emojis, redundant punctuation, desensitized usernames, some link addresses, etc. In addition, medical expressions in the text are often expressed in non-professional colloquial expressions, which are common problems in social media data and usually mislead the trained model. To address these issues, we use pre-trained language models as the basis; many text pre-processing and adversarial training are described in detail in the following sections, with corresponding experimental results.

2 Task 1a: Classify Tweets Reporting ADEs

This subtask aims to identify whether sentences contain adverse drug reactions, which can be modeled as a classification task. There are a total of 17385 samples in this dataset, where the positive to negative ratio is 1:11 (mentions of ADEs are labeled as 0), a total of 915 samples in the validation set (87 ADE labels and 828 NoADE labels), and 10,984 samples in the test set(Magge et al., 2021).

2.1 Method

Preprocessing Due to the spoken Twitter data, we first cleaned the data. (1) We remove the "@USER" and the placeholder "_" carried after it, and we consider that the URL information does not bring additional information to ADE-related content, so we choose to remove it. (2) We remove the redundant symbolic expressions and keep only one symbol, for example, "!!!", "???" transformed to "!","?". However, since the "...... " symbol is used as a label in task 1b, so we keep this symbol for all tasks. (3) We remove the emoticons from the text. (4) Finally, we remove the extra space symbols from the text. (5) We use an oversampling strategy to increase the number of positive samples. We use the following method: using the cleaned data, we construct duplicate positive samples by randomly combining positive samples with positive sample text or positive samples with negative sample text so that the model focuses more on the text with ADE label information. It is worth noting that we join two samples together in a random order, and the length of the new samples does not exceed the length of the pre-trained model, so we do not need additional processing.

Proceedings of the 29th International Conference on Computational Linguistic, pages 4–6
October 12–17, 2022.

Model set-up	Precision	Recall	F1
Deberta + over-sampling + FGM	0.79	0.59	0.67
Deberta + over-sampling + FGM + weighted-fusion	0.79	0.61	0.69
Average scores	0.65	0.50	0.56

Table 1: Results of Task 1a on the test set.

Model We use the Deberta-v3-large (He et al., 2021) model as a text encoder and then use adversarial training and a learning rate cosine transform strategy to aid the training. The [CLS] vectors of all hidden layers are weighted and fused, and the weights keep increasing as the number of layers increases. Finally, binary classification is output by linear layer mapping.

2.2 Experiments and Results

We set the batch size to 64 and use the Adamw (Loshchilov and Hutter, 2017) optimizer for training. For the Deberta parameter, We set the learning rate of the pre-trained model to 2e-6 and set the learning rate of the other layers to 1e-3. The learning rate decay strategy uses the Cosine Annealing Warm Restarts (Loshchilov and Hutter, 2016) method, and the weight decay factor is set to 0.001. The adversarial training is performed using FGM (Miyato et al., 2016). The number of iterations is 20 rounds using five-fold cross-validation training. The experimental results are shown in Table 1. As can be seen, the results show that our use of [CLS] weight summation does improve the robustness of the model and our model effect exceeds the average score.

3 Task 1b: ADE Span Detection

This subtask task extracts the location of ADE entities from the text and can be considered a sequence annotation task. Data distribution is the same as task 1a.

3.1 Method

Preprocessing We process the data using the method described in Section 2.2. We only use the data with label information for training, so the model of task 1a is used to identify the text labeled ADE as the input of task 1b during inference.

Model The pre-training model selection is the same as task 1a, and we use the W2NER (Li et al., 2022) model architecture instead of the traditional BERT+CRF architecture. We combine character information and location information at the encoding side, use inflated convolution at the decoding side to obtain entity information of different sizes, and finally use matrix location decoding instead of CRF (Lafferty et al., 2001).

3.2 Experiments and Results

We set the batch size to 8 and used the Adamw optimizer for training. The learning rate size is 1e-5, the learning rate decay strategy uses cosine decay, ten epochs are trained, the decay factor is 0.001, the size of the expanded convolution is set to [1,2,3,4], the dropout size is 0.5, and the character information and location information is set to 50. The experimental results are shown in Table 2. The results show that the solution we use exceeds the average score in all three metrics: accuracy, recall, and F1, indicating the effectiveness of our strategy.

Model set-up	Precision	Recall	F1
Ours	0.68	0.62	0.65
Average scores	0.54	0.52	0.53

Table 2: Results of Task 1b on the test set.

4 Task 1c: Normalization

The task is to extract the ADE keywords in tweet data and match these ADE keywords to the correct standard ADE terms. There are two difficulties in this task. First, there are only 1000+ tweet data containing ADE keywords, which constrains us not from using overly complex models. Secondly, there are more than 30,000 standard ADE terms, and most of the standard ADE terms are not match the tweet data.

4.1 Method

Preprocessing We use the pipeline model for task1c. We use the recall model(Huang et al., 2013) to recall the standard ADE terms, and then we use the ranking model to rank the standard ADE terms and select the best matching standard ADE term. In the recalled model, we found that the model tends to confuse similar terms, so we set the labels of these similar terms to 0.5 and added them to the training data as pseudo-data.

Model In the recalled model, we extract ADE keywords from the tweet data by task1b and tokenize the keywords, output word vectors by the pre-trained DeBERTa model, and average pool these word vectors into a keyword vector. We do the same for the standard ADE term to get the corresponding word vector. We calculate the dot product on these two vectors, sort the results, and set a threshold to filter the standard ADE terms. In the ranking model, we believe that the contextual information of the ADE keywords also contributes to word matching. We use [SEP] to concatenate the ADE keyword with the corresponding tweet sentence, feed it into the DeBERTa model, and adopt the output vector corresponding to [CLS] as the vector for that keyword. We do the same for the candidate standard ADE terms to obtain the vectors of candidate words. We dot the product keyword vector with candidate word vectors and get the highest scored standard ADE term predicted by the model.

4.2 Experiments and Results

We train and test on three pre-trained models, the Bert-base model, the ALBERT model, DeBERTa model, and the best result is obtained on the DeBERTa pre-trained model. Moreover, by adjusting the learning rate of CLS layer to 1e-2, dropout rate to 0.3, label smooth rate to 0.1, adding FGM perturbation, R-Drop loss function(Wu et al., 2021), warn up epoch to 0.3, the model achieves the best result on the seventh epoch. The experimental results are shown in Table 3.

Model set-up	Precision	Recall	F1
Ours	0.40	0.34	0.37
Average scores	0.12	0.11	0.12

Table 3: Results of Task 1c on the test set.

5 Conclusion

In this paper, we use task-specific methods for each of the three subtasks of classification, extraction, and normalization. We enhance the effectiveness of our model through various strategies and demonstrate the effectiveness of the model. In future work, we will specifically target pre-training tasks in the negative drug effect vertical, such as how to make the model acquire prior knowledge of drugs. .

References

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 Graciela Gonzalez-Hernandez Davy Weissenbacher, Ari Z. Klein. 2022. Overview of the seventh social media mining for health applications smm4h shared tasks at coling 2022. In *In Proceedings of the Seventh Social Media Mining for Health (SMM4H) Workshop and Shared Task*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *international conference on machine learning*.

Jingye Li, Hao Fei, Jiang Liu, Shengqiong Wu, Meishan Zhang, Chong Teng, Donghong Ji, and Fei Li. 2022. Unified named entity recognition as word-word relation classification.

Ilya Loshchilov and Frank Hutter. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Arjun Magge, Elena Tutubalina, Zulfat Miftahutdinov, Ilseyar Alimova, Anne Dirkson, Suzan Verberne, Davy Weissenbacher, and Graciela Gonzalez-Hernandez. 2021. Deepademiner: a deep learning pharmacovigilance pipeline for extraction and normalization of adverse drug event mentions on twitter. *Journal of the American Medical Informatics Association*, 28(10):2184–2192.

Takeru Miyato, Andrew M. Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification. *Learning*.

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.