Fake-EmoReact-Yao: A Robust Model for Fake News Detection on Twitter

Wei-Yao Wang
Department of Computer Science
National Chiao Tung University
Hsinchu, Taiwan
sf1638.cs05@nctu.edu.tw

Abstract

The challenge of Fake-EmoReact 2021 is to predict the label of tweet as real or fake. To mitigate the effect of fake news, we adopt DeBERTa to tackle the aforementioned problems. We further apply a post-processing method based on the characteristic of the dataset to improve our prediction performance. From the experiments, we expose the overfitting problems of the baselines, while DeBERTa still achieves state-of-the-art results. We further illustrate using different types of data between training data and testing data can alleviate overfitting problems, which indicates the importance of the model's generalization may be a potential issue for fine-tuning. As a result, our model reached the highest in the main track for the challenge. The source code of our model is publicly available[1].

1 Introduction

Natural language is often indicative of one's emotion. Hence, detecting emotions in textual conversations has been one of popular topics in the field of natural language processing (NLP) sentiment domain. Sentiment classifier can help researchers study such information on user's feeling. With the development of the Internet, Socia media like Twitter has been one of the most popular microblogging platforms where users can share real-time information related to all kinds of topics and events. The enormous and plentiful Tweet data has been proven to be a widely-used and real-time source of information in various important analytic tasks.

However, at the same time, social media enables users to get exposed to a myriad of misinformation and disinformation, including fake news thus misinformation gains a lot of attention in research fields and social issues. There are various aspects of dealing with fake news detection, such as propagation-based (Khoo et al., 2020; Ma et al., 2018), hybrid-based (Shu et al., 2019). Futhermore, dEFEND (Shu et al., 2019) and PLAN (Khoo et al., 2020) simultaneously enabled explainable of the models to gain further analysis.

A real-world problem, predicting real or fake of the given tweet, is the challenge called Fake-EmoReact[2] hosted by SocialNLP 2021. In this paper, we adopted DeBERTa (He et al., 2020) as the model to the shared task. We applied post-processing which turned the labels of the tweet with same *idx* into same labels via majority voting. Besides, we also examined pre-processing strategies, and used extensive baselines to illustrate these models suffering from overfitting problems, while DeBERTa still achieved state-of-the-art results.

In summary, the main results of our paper are as follows.

- Normalizing data can not raise much the proportion of the coverage of tokenizers in the challenge;
- DeBERTa demonstrates another paradigm of language models generalization;
- Post-processing brings us performance enhancement;
- Predicting normalized data on models trained with original data has better performance results than using same types between training and testing data.

2 Dataset and Metric

The Fake-EmoReact 2021 challenge dataset is collected from Twitter threads, which consists the original tweets, and the response tweets (which some include an animated GIF). Each sample includes the text of the original tweet, and information about the response: the reply text, the categories of the GIF response, and the label of the tweet. The detail of each field is as follows:

[1] https://github.com/yao0510/Fake-EmoReact-2021

[2] https://sites.google.com/view/covidfake-emoreact-2021

- *idx*: Running index of the samples.
- *text*: The text of the original tweet; may include mentions (@user), hashtags (#example), emojis etc. Emojis are presented by Unicode.
- *context_idx*: The index of the response tweets to the idx. There might be several different context_idx under same idx, which refer to different response numbers under a same original tweets.
- *reply*: The text content of the response tweet. In cases where the reply only contained a GIF response, this field will be an empty string.
- *categories*: The GIF category (or categories) of the GIF response which was included in the reply tweet, from a list of 44 categories. Not all of the replies contain a GIF response, thus categories may be empty.
- *label*: The label of tweet which is only in the training data. *fake* represents the original tweet is a fake news, and *real* states a real news.
- *mp4*: The hashed file name of the response GIF.

Table 1 illustrates an example of the tweet. The dataset is split into three files including **train.json**, **dev.json**, and **eval.json**.

The metric of the challenge is *Macro-F1* from Scikit-learn[3], which is computed as follows.

$$\begin{aligned} F1 &= \frac{2 * P * R}{P + R}, \\ Macro\text{-}F1 &= \sum \frac{F1}{N}, \end{aligned} \tag{1}$$

where P is precision, R is recall, and N is the number of classes.

3 Related Work

3.1 Graph-based Models

Graph neural networks or graph embeddings has been used on various domains including recommendation (Jin et al., 2020) and chemistry (Hao et al., 2020). Recently, Kipf and Welling (2017) proposed a simplified graph neural network model, called graph convolutional networks (GCN), which achieved state-of-the-art classification results on a number of benchmark graph datasets. Yao et al. (2019) proposed Text GCN by applying GCN to text classification problem in NLP. They built a single text graph for a corpus based on word co-occurrence and document word relations, and framed this problem as a node classification problem. Liu et al. (2020) and Ragesh et al. (2021) proposed TensorGCN and HeteGCN, seperately, to demonstrate the success of using graph-based models on NLP tasks. Lin et al. (2021) proposed BERTGCN by combining the advantages of large-scale pre-trained models and graph neural networks. They jointly trained the BERT and GCN modules to learn the representations from the massive amount of pre-trained data, and the label influence through the edges by propagating.

3.2 Pre-trained Language Models (PLMs)

Transformer-based pre-trained language models have significantly improved the performance of many NLP tasks. After BERT (Devlin et al., 2019) is presented, we have seen the rise of a set of large-scale PLMs such as GPT-3 (Brown et al., 2020), RoBERTa (Liu et al., 2019), XLNet (Yang et al., 2019), ELECTRA (Clark et al., 2020), and DeBERTa (He et al., 2020). These PLMs have been fine-tuned using task-specific labels and created new state of the art in many downstream tasks. To fill the gap of pre-training on a large corpus of English Tweets, Nguyen et al. (2020) proposed BERTweet, which trained on a 80GB corpus of 850M English Tweets on the RoBERTa pre-training procedure. BERTweet demonstrated the effectiveness by conducting experiments compared to RoBERTa$_{\text{base}}$ on three downstream Tweet NLP tasks of part-of-speech (POS) tagging, Named-entity recognition (NER), and text classification.

4 Method

The main goal of the challenge is to predict the label as true of fake by giving tweet and its GIF response. We applied pre-processing techniques to normalize data. It is noted that in final model, we did not use the normalized data, which is discussed in experiments. To cooperate with the characteristic of the dataset, we introduce a voting method in post-processing stage.

4.1 Problem Formulation

Given a tweet and its GIF response, the model should predict the label of tweet as real or fake,

[3] https://scikit-learn.org/stable/

idx	*text*	*categories*	*context_idx*	*reply*	*label*
35896	Academics and professors of twitter, if someone (say, a PhD student) emails ...	["dance"]	0	And, of course, email the pdf.	real

Table 1: An example eliminated *mp4* of the Fake-EmoReact 2021 challenge dataset.

which can be framed as a binary classification problem. Specifically, we utilized *text* and *reply* as the input, the output $Y \in \{0, 1\}$ from the model indicates fake or real.

4.2 Pre-Processing

Tweet data has different structure compared with formal coupus (e.g., Wikipedia). Since the data format is same as the competition last year, we follow the cleaning techniques introduced by (Wang et al., 2020) to normalize data and add a mapping of users and URLs. Specifically, the cleaning techniques are six steps in order: mapping users and urls, cleaning weird punctuations, transforming apostrophes, mapping unknown symbols to known punctuations, demojizing, and detweetizing. Besides, converting all tokens into lower case is not suggested according to their observation.

4.3 The DeBERTa Architecture

DeBERTa, decoding-enhanced BERT with disentangled attention, is composed of two novel techniques: a disentangled attention mechanism, and an enhanced mask decoder. Figure 1 shows differences between BERT and DeBERTa. Rather than mixing word embeddings and position embeddings in the beginning in BERT, DeBERTa represent a token by two vector to perform attention. That is, the attention weight of a word pair can be computed as a sum of three attention scores using disentangled matrices on their contents and positions as content-to-content, content-to-position, and position-to-content. Besides, DeBERTa uses relative position instead of absolute position that can improve the ability to model relative dependency between the tokens. To address the limitation of using relative position on masked language models, DeBERTa apply absolute position only when decoding, which is called enhanced mask decoder (EMD). We used DeBERTa-base as the pre-trained model, and fine-tuned to the tweet dataset with binary cross entropy loss.

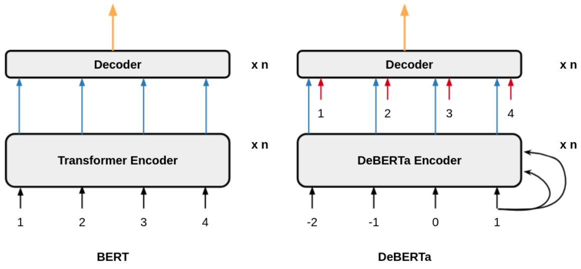

Figure 1: Differences between BERT and DeBERTa.

4.4 Post-Processing

In the description of *label*, it is mentioned the same *idx* will contain the same label. Therefore, a post-processing technique is introduced to cooperate with the characteristic of the dataset by considering all tweets in the same *idx* as the same importance. To decide the final labels, we applied a voting technique that is quite same as majority voting. Specifically, we replaced all labels to the major label in the same *idx*.

5 Experiment

5.1 Experimental Setup

5.1.1 Dataset Description

In Fake-EmoReact, we only had ground truth labels in training data, and we used **dev.json** as our validation data. That is, we fine-tuned hyperparameters based on validation data and chose best models from tuning to predict testing data, **eval.json**.

5.1.2 Parameter Settings

Our best model were trained for 3 epochs, the learning rate was 5e-7, the training batch size was 8, maximum sequence length the model will support was 113, the optimizer was AdamW (Loshchilov and Hutter, 2019), and the manual seed was 42. Most of these configurations were set as default arguments. Through validation, we selected the model checkpoint 48,000 as the final model.

5.1.3 Evaluation Platform

All of our evaluation processes were performed on a machine with Intel® Core™ i7-8700 3.2GHz CPU, Nvidia GeForce GTX 2070 GPU, and 32GB RAM, while the methods were implemented in Python 3.8.8 with the Pytorch framework. The operation system is Ubuntu 20.04. All of the models are implemented from Simple Transformers[4].

5.1.4 Baselines

To evaluate and compare the performance of DeBERTa, we also applied several BERT-based baselines. We applied the same optimization procedure with default parameter settings on each of them:

- BERT is a multi-layer bidirectional Transformer encoder;
- RoBERTa uses same architecture as BERT but optimizes with several techniques and more corpus;
- XLNet introduces permutations of the factorization to overcome the limitations of autoregressive language models and denoising auto-encoding;
- ELECTRA proposes a sample-efficient pretraining task by GAN-like architecture and the discriminative model is used.

5.2 Pre-processing Analysis

Follow the idea mentioned by (Wang et al., 2020), we applied DeBERTa-base tokenizer to validate the coverage shown in Table 2. As can be seen, the coverage of three data improved only a little after the pre-processing method, which indicates the dataset composition may be quite different as the dataset used by (Wang et al., 2020). Hence we did not use pre-processing data for training models.

5.3 Evaluation Results

5.3.1 Overfitting Examination

To evaluate whether the models overfit, we first used RoBERTa in Round 1 in both dev and eval phases. From the results, it is shown that RoBERTa achieved 0.9985 on dev data, while only had 0.58 on eval data. This can be a critical information to deduce the model overfit the data, hence we calculated the proportion of real and fake on test data in RoBERTa to observe whether the model is overfitting. It is noted that since leaderboard only showed the lastest submission, we only applied this examination offline. The proportion of real and fake on eval data predicted by RoBERTa is about 1:8, and all baselines except ELECTRA were nearly as the same proportion. ELECTRA predicted all tweets as fake, while DeBERTa had the proportion about 1:1.

5.3.2 Main Track Results

In the main track, we only submitted DeBERTa to verify the performance. Besides, we trained one DeBERTa model with original data followed by (Wang et al., 2020), which enhanced pre-trained language model and fine-tuned for downstream tasks, then utilized normalized data as one of the baselines, denoted as DeBERTa$_{\text{adversial}}$.

Table 3 is the results of DeBERTa and its variants. It is evident that DeBERTa achieves stable performance in the main track, and our post-processing technique slightly improved the performance, which produced the highest results in the challenge. We also tried to use enhanced DeBERTa (same types between training and testing data), but the model suffered from overfitting problems as well. However, feeding different types of data into enhanced DeBERTa can have better performance, which suggests adversarial methods for fine-tuning may be a potential factor to improve the performance.

6 Conclusion

In this paper, we applied DeBERTa and introduced a post-processing method according the characteristic of the dataset to mitigate the effect of fake news for Fake-EmoReact tasks, which achieved the highest performance in the challenge. From the experiments, we also illustrated other baselines suffered from overfitting problems, which again demonstrated the generalization and the robustness of DeBERTa. Besides, we found using different types of data between training and testing had potential of improving models, which indicates adversarial training algorithm for fine-tuning may be a potential factor for improving the results.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child,

[4] https://github.com/ThilinaRajapakse/simpletransformers

	Train		**Dev**		**Eval**	
Steps	*text*	*reply*	*text*	*reply*	*text*	*reply*
Original	0.7314	0.6498	0.6980	0.6274	0.7013	0.6176
Map users and URLs	0.7314	0.6500	0.6978	0.6273	0.7011	0.6176
Clean weird punctuations	0.7314	0.6500	0.6978	0.6273	0.7011	0.6176
Transform apostrophes	0.7492	0.6687	0.7132	0.6460	0.7190	0.6370
Map unknown to known punctuations	0.7512	0.6701	0.7150	0.6479	0.7207	0.6389
Demojize and unique same emojis	0.7668	0.6951	0.7415	0.6860	0.7472	0.6762
Transform more words	0.7673	0.6958	0.7423	0.6869	0.7478	0.6769

Table 2: Coverage of DeBERTa-base tokenizer in Round 2 Dataset

Model	Dev	Eval
DeBERTa$_{\text{adversial}}$	0.89	
DeBERTa	0.93	
DeBERTa$_{\text{post}}$	**0.94**	**0.9390**

Table 3: Results of dev and eval in main track. DeBERTa is to directly fine-tune for the downstream task, DeBERTa$_{\text{adversial}}$ is to use different types of data on enhanced DeBERTa, and DeBERTa$_{\text{post}}$ states DeBERTa after applying the post-processing technique.

Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020*. OpenReview.net.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019)*, pages 4171–4186. Association for Computational Linguistics.

Zhongkai Hao, Chengqiang Lu, Zhenya Huang, Hao Wang, Zheyuan Hu, Qi Liu, Enhong Chen, and Cheekong Lee. 2020. ASGN: an active semi-supervised graph neural network for molecular property prediction. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 731–752. ACM.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced BERT with disentangled attention. *CoRR*, abs/2006.03654.

Jiarui Jin, Jiarui Qin, Yuchen Fang, Kounianhua Du, Weinan Zhang, Yong Yu, Zheng Zhang, and Alexander J. Smola. 2020. An efficient neighborhood-based interaction model for recommendation on heterogeneous graph. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 75–84. ACM.

Ling Min Serena Khoo, Hai Leong Chieu, Zhong Qian, and Jing Jiang. 2020. Interpretable rumor detection in microblogs by attending to user interactions. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference*, pages 8783–8790. AAAI Press.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017*. OpenReview.net.

Yuxiao Lin, Yuxian Meng, Xiaofei Sun, Qinghong Han, Kun Kuang, Jiwei Li, and Fei Wu. 2021. Bertgcn: Transductive text classification by combining GCN and BERT. *CoRR*, abs/2105.05727.

Xien Liu, Xinxin You, Xiao Zhang, Ji Wu, and Ping Lv. 2020. Tensor graph convolutional networks for text classification. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference*, pages 8409–8416. AAAI Press.

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. *CoRR*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019*. OpenReview.net.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2018. Rumor detection on twitter with tree-structured recursive neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018*, pages 1980–1989. Association for Computational Linguistics.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. Bertweet: A pre-trained language model for english tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos*, pages 9–14. Association for Computational Linguistics.

Rahul Ragesh, Sundararajan Sellamanickam, Arun Iyer, Ramakrishna Bairi, and Vijay Lingam. 2021. Hetegcn: Heterogeneous graph convolutional networks for text classification. In *WSDM '21, The Fourteenth ACM International Conference on Web Search and Data Mining*, pages 860–868. ACM.

Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. 2019. defend: Explainable fake news detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019*, pages 395–405. ACM.

Wei-Yao Wang, Kai-Shiang Chang, and Yu-Chien Tang. 2020. Emotiongif-yankee: A sentiment classifier with robust model based ensemble methods. *CoRR*, abs/2007.02259.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, pages 5754–5764.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019*, pages 7370–7377. AAAI Press.