BERT based Adverse Drug Effect Tweet Classification

Tanay Kayastha
IIT Bombay, India
kayasthatanay@gmail.com

Pranjal Gupta
BITS Pilani - Hyderabad, India
pranjalgupta2199@gmail.com

Pushpak Bhattacharyya
IIT Bombay, India
pb@cse.iitb.ac.in

Abstract
This paper describes models developed for the Social Media Mining for Health (SMM4H) 2021 shared tasks (Magge et al., 2021). Our team participated in the first subtask that classifies tweets with Adverse Drug Effect (ADE) mentions. Our best performing model utilizes BERTweet followed by a single layer of BiLSTM. The system achieves an F-score of 0.45 on the test set without using any supplementary resources such as Part-of-Speech tags, dependency tags, or knowledge from medical dictionaries.

1 Introduction
In this effort, we focus on detecting tweets that have ADE mentions as a part of the Social Media Mining for Health (#SMM4H) 2021 shared tasks (Magge et al., 2021). Our team participated in the first subtask that classifies tweets with Adverse Drug Effect (ADE) mentions. Our best performing model utilizes BERTweet followed by a single layer of BiLSTM. The system achieves an F-score of 0.45 on the test set without using any supplementary resources such as Part-of-Speech tags, dependency tags, or knowledge from medical dictionaries.

2 Data
2.1 Dataset
The dataset consists of a training set (18,000 tweets), validation set (953 tweets), and test set (10,000 tweets). The dataset is highly imbalanced, with only 7% of the tweets containing ADE mentions. We tackle this challenge using sampling and per-class penalties in the objective function.

2.2 Preprocessing
We performed following preprocessing on the dataset:

1. Replace emoji with its text string (for example, ':)' with 'slightly smiling face')
2. Strip '#' from hashtags in tweets
3. Drop user-mentions and URLs
4. Lowercase all words

We used emoji1 package to translate emoji to text string.

3 Method
We explore three BERT-based models for classification: (i) BERT (Devlin et al., 2019), (ii) RoBERTa (Liu et al., 2019), and (iii) BERTweet (Nguyen et al., 2020). We pass the input through our BERT-based models to get token representations. To compute the sentence representations, we consider two cases - i) [CLS] token (fine-tuning) ii) we pass token representations without [CLS] and [SEP] through a single layer BiLSTM and concatenate the forward and backward context. The sentence representation is passed through a fully connected neural network layer followed by a sigmoid activation to predict probabilities.

To tackle class imbalance, we experiment with oversampling, undersampling, and addition of per-class penalties in the objective function. For oversampling approach, we randomly sampled positive examples with replacement until each class contained 10,000 tweets. For the undersampling approach, we randomly sample negative examples to create a balanced training dataset.

4 Experiments
For the classification task, each BERT model is trained for 10 epochs with a learning rate of 1 * 10^{-5} using Adam optimizer (Kingma and Ba,

1https://pypi.org/project/emoji/
Table 1: Task1a results on Validation set

| Name | Validation set |
|------|----------------|
|      | P   | R   | F1  |
| BERT\_base - Fine Tune | 0.697 | 0.708 | 0.702 |
| BERT\_base - unweighted | 0.742 | 0.708 | 0.724 |
| BERT\_base | 0.77 | 0.723 | 0.746 |
| RoBERTa | 0.845 | 0.754 | 0.797 |
| RoBERTa\_over | 0.637 | **0.892** | 0.743 |
| RoBERTa\_under | 0.659 | 0.862 | 0.747 |
| BERTweet\_raw | **0.864** | 0.784 | 0.823 |
| BERTweet | 0.812 | 0.862 | **0.836** |

Table 1: Task1a results on Validation set

We set the batch size to 32 and the maximum sequence length to 128. To tackle class imbalance, we add weights to the standard cross-entropy loss. We set weights as 0.7 and 0.3 for ADE and NoADE classes, respectively. We utilize PyTorch\(^2\) implementation of BERT for training. We train RoBERTa\_over, RoBERTa\_under and BERT\_base - unweighted, using standard unweighted cross-entropy loss. We conduct model selection for every 200 steps against the validation set using the F1-score of the ADE class for comparison.

5 Discussion

It is evident from Table 1 that BERT\_base outperforms BERT\_base - Fine Tune, and validates that the use of BiLSTM layer on top of BERT improves both precision and recall. Table 1 also shows that use of per-class penalties in the objective function (BERT\_base) results in better performance as compared to the model with unweighted objective function (BERT\_base - unweighted).

Table 2 shows that retaining emoji and hashtags in tweets help in achieving better performance on BERT\_base as against excluding those.

Table 1 shows that RoBERTa outperformed BERT\_base in all the evaluation metrics. However, RoBERTa\_over and RoBERTa\_under gave results comparable to BERT\_base. The results show that the ADE class’s oversampling and the NoADE class’s undersampling did not handle the class imbalance problem well. Hence, we resort to adding class-weights in our objective function.

BERTweet outperforms BERTtweet\_raw, which uses preprocessing techniques described in (Nguyen et al., 2020). Our preprocessing steps are inspired by (Nguyen et al., 2020) with the only difference being that we remove all user mentions and web/URL links from the tweet. We empirically validate our intuition that the user mentions, web links act as noise in the text and do not provide any valuable information needed for the classification task.

Table 3 shows the performance of BERTtweet on the test set. Our model’s performance is relatively poor on the Test set compared to the validation set, which can be attributed to overfitting. This overfitting can be reduced by adding dropout in the model. Table 3 shows the performance of BERTtweet on the Test set in the post-evaluation phase after the addition of dropout to the BiLSTM layers.

6 Conclusion

In this work, we explore an application of BERT to the task of binary classification on English Tweets. We validate that use of per-class penalties in the objective function helped in overcoming the class imbalance problem. We have empirically evaluated differently tuned model versions and preprocessing methods against F1-score for the "ADE" class. Experiments have shown that our model has achieved an F1-score of 0.46, precision of 0.523, and recall of 0.409 on the test set.

The future directions would be to evaluate the potential of supplementary resources in our model, such as Part-of-Speech Tags, Dependency Tags, knowledge from medical dictionaries (such as MedDRA).

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep
bidirectional transformers for language understanding.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

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.

Arjun Magge, Ari Klein, Ivan Flores, Ilseyar Alimova, Mohammed Ali Al-garadi, Antonio Miranda-Escalada, Zulfat Miftahudinov, Eulàlia Farré-Maduell, Salvador Lima López, Juan M Banda, Karen O’Connor, Abeed Sarker, Elena Tutubalina, Martin Krallinger, Davy Weissenbacher, and Graciela Gonzalez-Hernandez. 2021. Overview of the sixth social media mining for health applications (#smm4h) shared tasks at naacl 2021. In Proceedings of the Sixth Social Media Mining for Health Applications Workshop & Shared Task.

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, pages 9–14.