Automatic Extraction of Medication Names in Tweets as Named Entity Recognition

Carol Anderson[§], Bo Liu[§], Anas Abidin[§], Hoo-Chang Shin[§], Virginia Adams[§]
NVIDIA / Santa Clara, California, USA
{carola;boli;aabidin;hshin;vadams}@nvidia.com

Abstract—**Social media posts contain potentially valuable information about medical conditions and health-related behavior. Biocreative VII Task 3 focuses on mining this information by recognizing mentions of medications and dietary supplements in tweets. We approach this task by fine tuning multiple BERT-style language models to perform token-level classification, and combining them into ensembles to generate final predictions. Our best system consists of five Megatron-BERT-345M models and achieves a strict F1 score of 0.764 on unseen test data.**

Keywords— entity recognition, NER, BERT, Megatron, BioMegatron, RoBERTa, BERTweet, text mining

I. Introduction

Posts on social networks represent an enormous source of potentially useful health-related information. Twitter users currently generate an estimated 500 million tweets per day[1]. Studies have shown that tweets can be used to monitor various health-related phenomena, including infectious disease outbreaks (*1*), adverse drug events (*2*, *3*), and drug abuse (*4*). However, extraction of information from tweets is particularly challenging due to several characteristics of the tweet format. First, because tweets are short (limited to 140 characters), it can be difficult to unambiguously identify the topics or entities mentioned. Second, tweets are extremely noisy, often containing abbreviations, misspellings, emojis, hashtags, and urls. Third, tweets are lexically and syntactically quite different from the text typically used to pretrain the language models, such as BERT (*5*), that are the basis of current state-of-the-art information extraction methods. Fourth, any particular entity type is only found in a small fraction of tweets, meaning that even a large collection of labeled tweets may contain only a few examples of any entity type.

In Biocreative VII Task 3, we are asked to extract mentions of medications or dietary supplements from tweets by pregnant users. The training and development sets together contain 127,125 tweets, of which 311 contain at least one mention of a medication or supplement. We approach this task purely as a token-level classification problem. We use no handcrafted features other than a minor customization of the tokenizer used to preprocess tweets. We experiment with various BERT-style models, finding that the best performance is obtained with the Megatron-BERT-345M model (*6*). Finally, we boost performance by using multiple models together as ensembles (*7*).

II. Methods

A. Preprocessing

In the data provided for the challenge, entity labels were given in the form of character indices. We converted these into token labels by tokenizing the tweets with the standard spaCy English tokenizer, and assigning labels of `B-DRUG` or `I-DRUG` to tokens that were the first or non-first tokens, respectively, in a labeled entity. Based on analysis of errors during initial experiments, we added a short list of custom tokens to the spaCy tokenizer as infixes and prefixes, which ensured they would be split apart from surrounding text and treated as tokens. For example, we found that the medication `Zofran` appeared multiple times in the challenge data and was sometimes embedded within a larger token, as in the hashtag `#LifeWithAZofranPump`. Since partial tokens cannot be tagged as entities, Zofran would be missed in the preceding example unless the hashtag was split apart. The custom token list we used was [`zofran`, `Zofran`, `Concerta`, `shots`, `nitrous`, `\U000feb14`, and `/`].

Each of the models we used required an additional tokenization step prior to training. Megatron and BioMegatron models use the WordPiece tokenizer (*8*), while RoBERTa and BERTweet use a byte-level version of byte-pair encoding (*9*). In each case, the tokens produced by the spaCy tokenizer were further split into subtokens using the relevant tokenizer prior to training.

The authors of BERTweet (*10*) reported that they performed additional preprocessing steps before pretraining on tweets. These steps included using the NLTK TweetTokenizer, converting user mentions and urls into the special tokens `@USER` and `HTTPURL`, respectively, and converting emojis into text strings. Although we fine tuned BERTweet, we did not perform any of these additional steps prior to fine tuning; instead, due to time constraints, we used the same spaCy-tokenized data for all the models we fine tuned.

B. Models

We approached the extraction of medication mentions as a token-level classification task, as is common practice. We used three token labels: `B-DRUG`, `I-DRUG`, and `O`. We used the NeMo [2] code base for fine tuning and inference.

We first experimented with several different BERT-style models that differ mainly in their pretraining data or pretraining methods. These models included Megatron-BERT-345M (a 345-million parameter model pretrained on general domain text) (*6*), BioMegatron-BERT-345M (the same architecture as Megatron-BERT-345M, but pretrained on text from PubMed) (*11*), RoBERTa large (pretrained on general domain text) (*12*), and BERTweet large (the same architecture as RoBERTa large, but pretrained on tweets) (*10*). For all models, we used a classification head consisting of a single fully-connected layer with a dropout level of 0.5. We trained the models using a batch size of 64 on eight V100 GPUs, using the adam optimizer and the learning rates shown in Table III. We applied warmup annealing with a warmup ratio of 0.1 to the learning rate. We trained for the maximum number of epochs shown in Table III and saved the checkpoint with the highest token-level F1 score on the development set. The numbers reported in Table I were calculated using the evaluation script provided by the challenge organizers. The test set metrics in Table II are the official results provided by the challenge organizer.

[§] authors contributed equally.

[1] retrived from https://www.internetlivestats.com/twitter-statistics/, October 8, 2021

[2] https://github.com/NVIDIA/NeMo

TABLE I

Performance of models trained on the training set and evaluated on the development set. The ensemble listed here is the first ensemble described in II-C.

		Overlap			Strict		
Model	Vocabulary	Precision	Recall	F1	Precision	Recall	F1
Megatron-BERT-345M	BERT large uncased	0.88	0.85	0.86	0.85	0.82	0.84
Megatron-BERT-345M	BERT large cased	0.81	0.84	0.83	0.77	0.79	0.77
BioMegatron-BERT-345M	BERT large uncased	0.91	0.83	0.87	0.82	0.77	0.79
BioMegatron-BERT-345M	BERT large cased	0.88	0.74	0.80	0.84	0.71	0.77
RoBERTa large	RoBERTa large	0.85	0.82	0.83	0.78	0.77	0.78
BERTweet large	RoBERTa large	0.81	0.86	0.83	0.78	0.83	0.81
Ensemble of five different models		**0.92**	**0.86**	**0.89**	**0.90**	**0.84**	**0.87**

TABLE II

Performance of ensembles on the test set.

	Overlap			Strict		
Ensemble	Precision	Recall	F1	Precision	Recall	F1
Five different models	0.847	0.714	0.775	0.823	0.694	0.753
Megatron-BERT-345M uncased	**0.835**	**0.755**	**0.793**	**0.805**	**0.728**	**0.764**

TABLE III

Hyperparameters

Model	Learning rate	Max epochs
Megatron-BERT-345M	5 x 10^{-5}	30
BioMegatron-BERT-345M	8 x 10^{-5}	30
RoBERTa large	8 x 10^{-6}	40
BERTweet large	8 x 10^{-6}	40

C. Ensembles

For our submissions, we created two different ensembles. The first ensemble consisted of five different models: Megatron-BERT-345M-uncased, Megatron-BERT-345M-cased, BioMegatron-BERT-345M-uncased, RoBERTa large, and BERTweet large. Each model was trained on the training set, and we used the checkpoint that performed best based on the development set. To generate final token labels, we calculated a weighted average of the class probabilities produced by each model for each token, using the following weights: BioMegatron-BERT-345M-uncased: 1, Megatron-BERT-345M-uncased,: 2, Megatron-BERT-345M-cased: 1.2, RoBERTa large: 0.4, BERTweet large: 1.4. We chose these weights through a random search constrained to the range between 0 and 2 (inclusive) in increments of 0.1. We initially also included a BioMegatron-BERT-345M-cased model in the ensemble, but we found that a weight of zero for that model gave the best performance on the development set, so we excluded it from the final ensemble.

The second ensemble consisted of five Megatron-BERT-345M models trained using the "out-of-fold" method. In this approach, we combined the training and development sets and then divided them randomly into five subsets. To train each model, we used four of these subsets as the training set and held out the fifth as a validation set. We used the checkpoint that performed best on the held-out set in each of the five runs in our final ensemble. At inference time, we calculated the mean of the token class probabilities from each of the five models and chose the token class with the highest probability.

III. Results and Conclusions

As shown in Table I, Megatron-BERT-345M-uncased gave the highest F1 scores of any single model when trained on the training set and evaluated on the development set. This ran counter to our expectation that BioMegatron-BERT would be better able to detect medications, given its pretraining on biomedical literature. A possible explanation for the poorer performance of BioMegatron-BERT compared to Megatron-BERT could be that tweets are linguistically more similar to the general domain text used to pretrain Megatron-BERT than to biomedical literature. In this regard, it is interesting to compare the peformance of the RoBERTa and BERTweet models, which share the same architecture but differ in their pretraining data. BERTweet modestly outperforms RoBERTa in the strict evaluation metrics. In terms of strict F1 scores, BERTweet was the second-best performing model after Megatron-BERT-345M-uncased. The performance of BERTweet could possibly be improved if the same preprocessing steps used prior to BERTweet pretraining (described in II-A above) were also applied before fine tuning.

The performance of our two ensembles on the test set is shown in Table II. The ensemble consisting entirely of Megatron-BERT-345M models outperformed the ensemble of different models, which again was contrary to our expectations. We expected that the ensemble of different models trained on the same data would outperform the ensemble consisting of a single model type trained on different tranches of data. Our results suggest that performance on this task is limited more by the training data than by model architecture. We also note that the performance of the ensemble of different models on the test set was much lower than its performance on the development set. This could be partially explained by the fact that we effectively overfitted to the development set by using it to choose the "best" checkpoint from each training run. However, the gap is so large that we suspect it indicates a significant difference in the distribution of text found in the development and test sets.

References

1. A. Signorini, A. M. Segre, P. M. Polgreen, *PloS one* **6**, e19467 (2011).

2. A. Cocos, A. G. Fiks, A. J. Masino, *Journal of the American Medical Informatics Association* **24**, 813–821, ISSN: 1067-5027, eprint: https://academic.oup.com/jamia/article-pdf/24/4/813/34148877/ocw180.pdf, (https://doi.org/10.1093/jamia/ocw180) (Feb. 2017).

3. K. O'Connor *et al.*, presented at the AMIA annual symposium proceedings, vol. 2014, p. 924.

4. D. M. Kazemi, B. Borsari, M. J. Levine, B. Dooley, *Journal of Public Health* **39**, 763–776 (2017).

5. J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, *arXiv preprint arXiv:1810.04805* (2018).
6. M. Shoeybi *et al.*, *arXiv preprint arXiv:1909.08053* (2019).
7. G. E. Hinton, O. Vinyals, J. Dean, *ArXiv* **abs/1503.02531** (2015).
8. R. Sennrich, B. Haddow, A. Birch, *arXiv preprint arXiv:1508.07909* (2015).
9. A. Radford *et al.*, presented at the.
10. D. Q. Nguyen, T. Vu, A. T. Nguyen, presented at the Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 9–14.
11. H.-C. Shin *et al.*, *BioMegatron: Larger Biomedical Domain Language Model*, 2020, arXiv: 2010.06060 (cs.CL).
12. Y. Liu *et al.*, *CoRR* **abs/1907.11692**, arXiv: 1907.11692, (http://arxiv.org/abs/1907.11692) (2019).