A Joint Training Approach to Tweet Classification and Adverse Effect Extraction and Normalization for SMM4H 2021

Mohab Elkaref
IBM Research Europe
Daresbury, United Kingdom
mohab.elkaref@ibm.com

Lamiece Hassan
University of Manchester
Manchester, United Kingdom
lamiece.hassan@manchester.ac.uk

Abstract

In this work we describe our submissions to the Social Media Mining for Health (SMM4H) 2021 Shared Task (Magge et al., 2021). We investigated the effectiveness of a joint training approach to Task 1, specifically classification, extraction and normalization of Adverse Drug Effect (ADE) mentions in English tweets. Our approach performed well on the normalization task, achieving an above average f1 score of 24%, but less so on classification and extraction, with f1 scores of 22% and 37% respectively. Our experiments also showed that a larger dataset with more negative results led to stronger results than a smaller more balanced dataset, even when both datasets have the same positive examples. Finally we also submitted a tuned BERT model for Task 6: Classification of Covid-19 tweets containing symptoms, which achieved an above average f1 score of 96%.

1 Introduction

Social media platforms such as Twitter are regarded as potentially valuable tools for monitoring public health, including identifying ADEs to aid pharmacovigilance efforts. They do however pose a challenge due to the relative scarcity of relevant tweets in addition to a more fluid use of language, creating a further challenge of identifying and classifying specific instances of health-related issues. In this year's task as well as previous SMM4H runs (Klein et al., 2020) a distinction is made between classification, extraction, and normalization. This is atypical of NER systems, and many other NER datasets present their datasets, and are consequently solved in a joint approach.

Gattepaille (2020) showed that simply tuning a base BERT (Devlin et al., 2019) model could achieve strong results, even beating ensemble methods that rely on tranformers pretrained on more academic texts such as SciBERT (Beltagy et al., 2019), BioBERT (Lee et al., 2020) or ensembles of them, while approaching the performance of BERT models specifically pretrained on noisy health-related comments (Miftahutdinov et al., 2020).

2 Methods

2.1 Pre-processing

Despite the noisy nature of Twitter data, for Task 1 we attempted to keep any pre-processing to a minimum. This was motivated by the presence of spans within usernames and hashtags, in addition to overlapping spans and spans that included preceding or trailing white-spaces. For training and validation data we ignored overlapping and nested spans and chose the longest span as the training/tuning example.

We also compiled a list of characters used in the training data for use in creating character embeddings. This was not limited to alpha-numeric characters, but also included emojis, punctuation, and non-Latin characters. We then removed any character appearing less than 20 times[1] in the training set, and a special `UNK` character embedding was added. Additionally for the training, validation, and testing data we tokenized the tweets and obtained part-of-speech tags using the default English model for the Stanza (Qi et al., 2020) pipeline.

Our training set was supplemented with the CSIRO Adverse Drug Event Corpus(CADEC) (Karimi et al., 2015) and was processed in the same manner as above.

For Task 6 no pre-processing was done.

2.2 Task 1 Model

Word Representation The BERT vectors produced for each tweet are not necessarily aligned with the tokens produced by the Stanza tokenizer. For this reason we additionally compile a sub-word token map to construct word embeddings from the token embeddings produced by our BERT model

[1]This threshold was arrived at through trial and error.

Proceedings of the Sixth Social Media Mining for Health Workshop 2021, pages 91–94

(excluding the [CLS] vector). The final word embedding is a summation of the component vectors.

POS Tags & Char-LSTM We use randomly initialized trainable embeddings for universal POS (UPOS) tags predicted by the Stanza pos-tagger. For each word we also use a 1-layer LSTM to produce an additional representation. The input to this LSTM would be the embeddings for each character in a word in order of appearance. This is intended to capture both the recurring patterns indicating prefixes/suffixes and to also learn to disregard repeated letters and misspellings so as to overcome the noisiness of the data.

Bi-LSTM Hidden Layer While BERT is itself a bi-directional context-aware representation of a given sentence, we experimented with the addition of a bidirectional Lstm (Bi-LSTM) layer in order to incorporate the additional pos tag and char-LSTM embeddings, and model the interactions between them across the whole context of a tweet.

ADE Identification Subtask 1(a) requires simple classification as ADE or NoADE and so we simply used the [CLS] vector output from our BERT model as input to a softmax layer with two nodes.

ADE Extraction & Normalization Task 1(b) and 1(c) were approached jointly. The training data was reformulated a BIO labelling scheme that incoporates associated MedDRA tags, as is common for other NER tasks. Thus the final classification layer for both tags is a softmax with $(\{B, I\} \times MedDRA\,Tags + \{O\})$-nodes. We use a greedy approach to obtain the final tweet classification and token classification from the corresponding softmax layers. The spans are determined based on the longest uninterrupted sequence of tokens receiving the same *normalization* tag. Interruptions in this context mean classified as either O or B-$\star$. Additionally, uninterrupted spans consisting only of I-$\star$ but having the same normalization tag are considered valid spans. Thus, the following two sequences ([O,O,B-1234,I-1234,O] and [O,O,I-1234,I-1234,O]) translate to the same final span.

2.3 Task 6 Model

Task 6 proved to be a substantially easier challenge than Subtask 1(a), as can be seen in Subsection 3.2. Our approach was to simply tune a BERT model, with the [CLS] vector being used as input to a softmax classification layer.

Parameter	
Task 1	
POS embedding dimension	8
Character embedding dimension	16
Character LSTM dimension	8
Bi-LSTM hidden layer dimension	256
Training Epochs	50
Mini-batch size	32
Update Strategy	Adam
Learning Rate	$1 \times 10^{-4}/2 \times 10^{-5}$
Task 6	
Training Epochs	10
Mini-batch size	8
Update Strategy	Adam
Learning Rate	$1 \times 10^{-5}/2 \times 10^{-5}$

Table 1: Training parameters for Tasks 1 & 6

3 Experiments & Results

We implemented our models using the PyTorch (Paszke et al., 2019) framework, and for the core BERT model we used the pretrained bert-base model from the Huggingface transformers (Wolf et al., 2020) library. For both tasks we optimize parameters using Adam (Kingma and Ba, 2014). We experiment with different learning rates but keep default parameters for β_1, β_2, and ϵ.

3.1 Task 1

One of the largest challenges of Task 1 is the huge imbalance of tweets containing ADEs vs tweets that do not. This is demonstrated in Table 3 where just over 7% of tweets in both training and validation sets contain ADEs. In contrast, the CADEC dataset has $\approx$ 37% of examples with ADEs. To explore the effect of this distribution we constructed two training sets. The first is a dataset containing all the CADEC data in addition to training data tweets containing ADEs. This results in a dataset with $\approx$ 46% of examples with ADEs, which we will refer to as the *Partial datatset* going forward. The second dataset we use for training is all of the task training data and the whole CADEC dataset, which we will be referring to as the *Full dataset*, with the proportion of ADE examples being $\approx$ 16%.

We train the model jointly over all three subtasks, minimizing over the sum of negative log likelihood losses ($L_{SUM} = L_{DET} + L_{NER}$) for both the classification ($L_{DET} = -\sum_i^N \sum_c^{C_{DET}} y_{ic} log(\hat{y}_{ic})$) and extraction & normalization ($L_{NER} = -\sum_i^N \sum_c^{C_{NER}} y_{ic} log(\hat{y}_{ic})$) layers. Where N is the total number of minibatches, C_{DET} and C_{NER} are the classes for classification and extraction & normalization respectively, and y_* and $\hat{y}_*$ are the target and predicted classes.

Train dataset	Classification			Extraction			Normalization		
Target dataset	f1	p	r	f1	p	r	f1	p	r
Partial dataset									
Validation (1×10^{-4})	14.9	8.0	100.0	10.5	6.1	37.9	19.1	11.1	69.0
Validation (2×10^{-5})	14.8	8.0	100.0	9.3	5.5	29.9	18.2	10.8	58.6
Full dataset									
Validation	70.1	78.8	63.1	26.9	27.4	26.4	50.3	51.2	49.4
Test	22.0	35.9	16.4	37.0	58.0	27.5	24.0	37.1	17.8
Median of all submissions									
Test	44.0	50.5	40.9	42.0	49.3	45.8	22.0	23.1	21.8

Table 2: Task 1 Experimental Results.

Dataset	Total tweets	ADE tweets
CADEC	7597	2853
Training data	17358	1235
Validation data	915	65

Table 3: Task 1 dataset statistics.

α	Validation			Test		
	f1	p	r	f1	p	r
1×10^{-5}	98.3	98.2	98.3	94.0	94.1	94.1
2×10^{-5}	98.6	98.6	98.6	94.0	93.7	93.7
Median of all submissions				93.0	93.2	93.2

Table 4: Task 6 Experimental Results.

Our experiments on the *partial datasets* yielded weak results, with only a slight improvement when using a learning rate of 1×10^{-4} over 2×10^{-5}. Training on the *full dataset* with a learning rate of 2×10^{-5} produced far stronger results, with the f1 score for tweet classification increasing to 70.1% from 14.9% on the validation set, and to 26.9% from 10.5% for span extraction, and finally to 50.4% from 19.1% for span normalization. Training our model with a learning rate of 1×10^{-4} yielded unusable results and an unstable model, which suggests that this is too high a learning rate for larger datasets. It is interesting to note that while training on the full datatset dramatically improved f1 scores for all three subtasks, there was a general drop in recall and an increase in precision. This suggests that the model trained on the partial dataset was far more likely to produce false positives, and was unable to recognize the absence of ADEs despite negative examples constituting $\approx$ 53% of examples. The results of our experiments are summarized in Table 2.

Our final submission was trained on the full dataset and showed a similar pattern on the Test set producing better precision, beating the arithmetic mean of all submissions for extraction and normalization, but showed worse recall for all three subtasks. This resulted in the model only achieving an above average f1 score on subtask 1(c).

3.2 Task 6

Our approach to Task 6 is essentially the same as that for subtask 1(a), but with a smaller, more balanced dataset. We experiment with two learning rates, 1×10^{-5} and 2×10^{-5}, and minimize over a negative log likelihood loss $L = -\sum_i^N \sum_c^C y_{ic} log(\hat{y}_{ic})$.

The resulting models produced strong results, as shown in Table 4, with close validation f1 scores (98.6% and 98.3%). We used classifications by both models as our final submission, and both beat the median of all submissions with an f1 score of 94% for both models.

4 Conclusion

In this work we explored the efficacy of jointly training a BERT model to jointly learn to perform classification, extraction, and normalization of ADE in tweets provided for Task 1 in SMMH 2021 Shared Task. While this approach did not produce classification and extraction above the median submission, it did achieve a normalization score that is. Additionally our experiments show that the seemingly lopsided ratio of tweets with/without ADEs resulted in stronger performance than a more "balanced" dataset. Finally, we showed that tuning a BERT model produces very strong results on Task 6, in classifying tweets related to Covid-19.

Acknowledgements

This work was part of STFC Hartree Centre's Innovation Return on Research programme, Contract Number: 4070116091. LH was funded via the Medical Research Council (Ref: MR/S004025/1).

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Lucie Gattepaille. 2020. How far can we go with just out-of-the-box bert models? In *Proceedings of the Fifth Social Media Mining for Health Applications Workshop & Shared Task*, pages 95–100.

Sarvnaz Karimi, Alejandro Metke-Jimenez, Madonna Kemp, and Chen Wang. 2015. Cadec: A corpus of adverse drug event annotations. *Journal of biomedical informatics*, 55:73–81.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ari Klein, Ilseyar Alimova, Ivan Flores, Arjun Magge, Zulfat Miftahutdinov, Anne-Lyse Minard, Karen O'Connor, Abeed Sarker, Elena Tutubalina, Davy Weissenbacher, and Graciela Gonzalez-Hernandez. 2020. Overview of the fifth social media mining for health applications (#SMM4H) shared tasks at COLING 2020. In *Proceedings of the Fifth Social Media Mining for Health Applications Workshop & Shared Task*, pages 27–36, Barcelona, Spain (Online). Association for Computational Linguistics.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Arjun Magge, Ari Klein, Ivan Flores, Ilseyar Alimova, Mohammed Ali Al-garadi, Antonio Miranda-Escalada, Zulfat Miftahutdinov, 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*.

Zulfat Miftahutdinov, Andrey Sakhovskiy, and Elena Tutubalina. 2020. KFU NLP team at SMM4H 2020 tasks: Cross-lingual transfer learning with pretrained language models for drug reactions. In *Proceedings of the Fifth Social Media Mining for Health Applications Workshop & Shared Task*, pages 51–56, Barcelona, Spain (Online). Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.