InfoMiner at WNUT-2020 Task 2: Transformer-based Covid-19 Informative Tweet Extraction

Hansi Hettiarachchi **Tharindu Ranasinghe**

School of Computing and Digital Technology, Birmingham City University, UK

Research Group in Computational Linguistics, University of Wolverhampton, UK

Introduction

By 15th November 2020, coronavirus COVID-19 is affecting 213 countries around the world infecting more than 54 million people and killing more than 1.3 million.

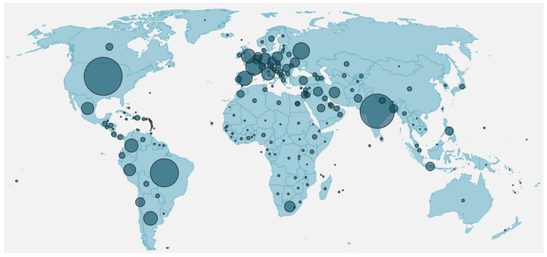

Figure 1:Affected countries from COVID-19. Circles show number of confirmed coronavirus cases per country. Source `www.bbc.co.uk`

- Recently, much attention has been given to build monitoring systems to track the outbreaks of the virus and these monitoring tools have begun to use social media as the medium to get information since they are more efficient.
- Since the manual approaches to identify the informative tweets require significant human efforts, an automated technique to identify the informative tweets will be invaluable to the community.
- The objective of this shared task is to **automatically identify whether a COVID-19 English tweet is informative or not**. Such **informative Tweets provide information about recovered, suspected, confirmed and death cases as well as location or travel history of the cases**.

Tweet	Label
Update: Uganda Health Minister Jane Ruth Aceng has confirmed the first #coronavirus case in Uganda. The patient is a 36-yearold Ugandan male who arrived from Dubai today aboard Ethiopian Airlines. Patient travelled to Dubai 4 days ago. #CoronavirusPandemic	Informative
Indonesia frees 18,000 inmates, as it records highest #coronavirus death toll in Asia behind China HTTPURL	Uninformative

Table 1: Sample Tweets for each label in the dataset.

Methodology

Predicting whether a certain tweet is informative or not can be considered as a sequence classification task. Therefore, the main idea of the methodology is that we train a classification model with several transformer models in-order to identify informative tweets.

Transformers for Text Classification

For text classification tasks, transformer models take the final hidden state **h** of the [CLS] token as the representation of the whole sequence. A simple softmax classifier is added to the top of the transformer model to predict the probability of a class c as shown in Equation 1 where W is the task-specific parameter matrix.

$$p(c|\mathbf{h}) = \text{softmax}(W\mathbf{h}) \quad (1)$$

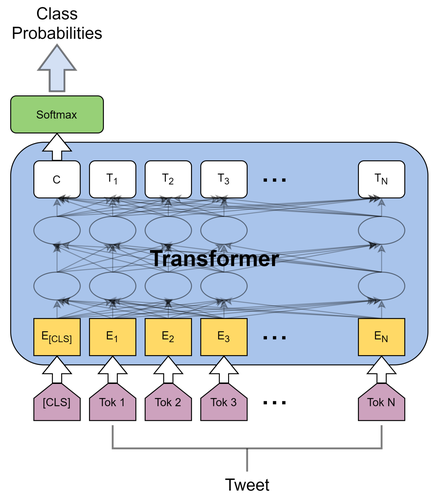

Figure 2:Text Classification Architecture

We used several pre-trained transformer models in this task. These models were used mainly considering the popularity of them (e.g. BERT [3], XLNet [7], RoBERTa [5], ELECTRA [1], ALBERT [4] and relatedness to the task (e.g. COVID-Twitter-BERT (CT-BERT) [6] and BERTweet [2]).

Data Preprocessing

We employed several data preprocessing techniques.

1. **Removing or filling usernames and URLs.** In WNUT-2020 Task 2 data set, mention of a user is represented by *@USER* and a URL is represented by *HTTPURL*. For all the models except CT-BERT and BERTweet, we removed those mentions. For CT-BERT and BERTweet, we used the corresponding fillers to replace usernames and URLs in the data set.
2. **Converting emojis to text** For all the models except CT-BERT and BERTweet, we used *demoji* python package to convert emojis to text. For CT-BERT and BERTweet *emoji* was used, because these models are trained on correspondingly converted Tweets.

Also, for uncased pretrained models (e.g. *albert-xxlarge-v1*), **all tokens were converted to lower case**.

Fine-tuning Strategies

To improve the models, we experimented different fine-tuning strategies.

1. **Self-Ensemble (SE)** - Same model architecture is trained or fine-tuned with different random seeds or train-validation splits. Then the output of each model is aggregated to generate the final results. As the aggregation methods, we analysed majority-class and average in this research.
   - *Majority-class SE (MSE)* - As the final class, we computed the mode of the classes predicted by each model.
   - *Average SE (ASE)* - Final probability of class c is calculated as the average of probabilities predicted by each model as in Equation 2 where *h* is the final hidden state of the [CLS] token. Then the class with highest probability is selected as the final class.

$$p_{ASE}(c|h) = \frac{\sum_{k=1}^{N} p_k(c|h)}{N} \quad (2)$$

2. **Entity Integration (EI)** - We replaced the unknown tokens with their named entities using spaCy.
3. **Language Modelling (LM)** - We retrained the transformer model on task data set before fine-tuning it for the downstream task; text classification.

Results

- **CT-BERT** model outperformed the other models. Following this, we limited the further experiments only to CT-BERT model.
- We experimented that **increasing the epoch count from 3 to 5 increases the results**. However, increasing it more than 5 did not further improved the results.
- Out of two self ensemble strategies, **ASE is given a higher F1 than MSE**.
- Entity Integration and Language Modelling did not improve the results for this data set.

Considering the evaluation results on validation data set, as InfoMiner 1 we selected the fine-tuned CT-BERT model with ASE and $2e^{-5}$ learning rate. As InfoMiner 2 same model and parameters with MSE was picked. Highest F1 we received is for MSE strategy.

Model	Precision	Recall	F1-score
Top-ranked	0.9135	0.9057	0.9096
InfoMiner 1	0.9107	0.8856	0.8980
InfoMiner 2	0.9102	0.8909	0.9004
Task baseline	0.7730	0.7288	0.7503

Table 2: Results of test data predictions

Conclusions

- Our experiments show that the CT-BERT is the most successful transformer model from several transformer models we experimented for this task.
- We presented several fine tuning strategies: self-ensemble, entity integration and language modelling that can improve the results.
- Overall, our approach is simple but can be considered as effective since it achieved **10th place** in the leader-board out of 55 participants.
- The code, and the trained classification models will be freely available to everyone interested in working on identifying informative tweets on `https://github.com/hhansi/informative-tweet-identification`.

References

[1] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning.
ELECTRA: Pre-training text encoders as discriminators rather than generators.
In *International Conference on Learning Representations*, 2020.

[2] Thanh Vu Dat Quoc Nguyen and Anh Tuan Nguyen.
BERTweet: A pre-trained language model for English Tweets.
arXiv preprint, arXiv:2005.10200, 2020.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova.
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, June 2019. Association for Computational Linguistics.

[4] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut.
Albert: A lite bert for self-supervised learning of language representations.
In *International Conference on Learning Representations*, 2020.

[5] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov.
Roberta: A robustly optimized bert pretraining approach.
arXiv preprint arXiv:1907.11692, 2019.

[6] Martin Müller, Marcel Salathé, and Per E Kummervold.
Covid-twitter-bert: A natural language processing model to analyse covid-19 content on twitter.
arXiv preprint arXiv:2005.07503, 2020.

[7] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le.
Xlnet: Generalized autoregressive pretraining for language understanding.
In *Advances in neural information processing systems*, pages 5753–5763, 2019.