SU-NLP at W-NUT 2020 Task 2: Ensemble Models for Informative Tweet Classification

Kenan Fayoumi
Sabancı University
Istanbul, Turkey
kenanf@sabanciuniv.edu

Reyyan Yeniterzi
Sabancı University
Istanbul, Turkey
reyyan@sabanciuniv.edu

Abstract

In this paper, we address the problem of identifying informative tweets related to COVID-19 in the form of a binary classification task as part of our submission for W-NUT 2020 Task 2. Specifically, we focus on ensembling methods to boost the classification performance of classification models such as BERT and CNN. We show that ensembling can reduce the variance in performance, specifically for BERT base models.

1 Introduction

After the recent virus outbreak, social media platforms, like Twitter, have been flooded with COVID-19 related content. Some of those content contain useful and valuable information such as the current status of the outbreak in particular locations. However, like any other topic being discussed in social media, the majority of the content is unrelated, subjective, or uninformative. Finding informative content would require extensive manual search, which is not scalable due to the size of the content. Therefore, automatically identifying both relevant and informative content has become an important task. In this paper, we study this specific problem as part of the W-NUT 2020 shared-task on identification of informative COVID-19 English tweets, and report our findings.

We specifically focused on transformer based neural network architectures and also simple but still effective CNN models. We also experiment with ensembling different models as we attempt to reduce the performance variance introduced from a model and combine strength of models in order to improve the final classification performance. Code to our classification and ensemble methods are available at our Github repository [1] .

2 Data and Task Description

Twitter, which is full of noisy user-generated content, was specifically chosen by the organizers and 10K COVID-19 related tweets were collected and labeled (Nguyen et al., 2020). Collected tweets were labeled as either informative or not. Informative tweets are tweets that provide information about recovered, suspected, confirmed and death cases as well as location or travel history of the cases. Within the collected 10K tweets, 4719 of them were labeled as informative, while the rest 5281 as uninformative. The dataset was split into 70/10/20 for training/validation/test. The test set has not been released publicly, therefore most of our experimental results are over the validation set. Informative class F1 score was used as the official evaluation metric for this task.

3 Classification Models

We approached the problem by first training different classification models, and then combining their strengths. As our first model, we chose a simple yet effective CNN text classification model. As for the rest, three pre-trained transformer-based models were adapted for our particular task. In this section, we discuss these models and their performances.

3.1 CNN

As our baseline experiment, we chose a very common CNN architecture by Kim (2014). CNN has previously proven successful in NLP tasks and we believe it's a strong baseline for text classification tasks. In this article, Kim (2014) showed the useful effects of pre-trained word embeddings on several text classification tasks. Similarly, given that our data is from Twitter, we used two pre-trained embeddings which had been trained on tweets. These are the Glove Twitter 200-dimensional embeddings

[1] https://github.com/SU-NLP/W-NUT2020-Task-2

Proceedings of the 2020 EMNLP Workshop W-NUT: The Sixth Workshop on Noisy User-generated Text, pages 423–427
Online, Nov 19, 2020.

trained with 2 billion tweets [2], and the Word2Vec embeddings [3] which had been trained on 400 million tweets and has 400 dimensions.

After hyper-parameter tuning, the optimum setting for the CNN-based models has been found as the following: filter windows of 2, 3, 4, 5 with 300 feature map in each, Rectified Linear Unit (ReLu) activation function, a dropout rate of 20%. Using Adam optimizer for training with a learning rate of 0.001, and 16 as the batch size. All tweets are tokenized by white-spaces and then padded or cropped to a length of 128 tokens.

3.2 BERT

As our stronger baseline, we chose Bidirectional Encoder Representations (BERT) (Devlin et al., 2018). BERT models had been pre-trained over large document collections (Wikipedia and books) in unsupervised manner. Later on, they can be fine-tuned for a particular task by using labeled data of that corresponding task. Similarly we fine-tuned the base BERT model (12 layers, 12 attention heads, and 110 million parameters) for our classification task.

HuggingFace implementation of BERT [4] was used with the following optimized hyper-parameters: learning rate is 1e-5, training batch size is 8, and all tweets are cropped or padded to 128 tokens.

3.3 ALBERT

We also used A Lite BERT for Self-supervised Learning of Language Representations (ALBERT) (Lan et al., 2019). ALBERT had been trained on the same corpus as BERT. It is not only lighter, but it also surpassed BERT in many NLP tasks.

We used the HuggingFace implementation for ALBERT base model [5] and the same hyper-parameters used with BERT.

3.4 CT-BERT

Recent work, like BIOBERT (Lee et al., 2019) and SCIBERT (Beltagy et al., 2019), have shown that pre-training transformer models with specialized corporas can boost performances for domain-specific tasks. Similarly, CT-BERT, (Müller et al., 2020) which had been pre-trained on a corpus of 22.5 million tweets related to COVID-19, falls into the same domain of our task of tweets about Covid-19, and therefore was explored.

Pre-trained model weights are provided in the authors Github page [6]. Same hyper-parameter setting used in BERT and ALBERT, was also used in this setting.

3.5 Experiments

All classification models' performances on the validation set is reported in Table 1 .

Model	F1 Score
CNN Glove	0.8320
CNN W2V	0.8355
BERT	0.8848
ALBERT	0.8954
CT-BERT	0.9049

Table 1: The performances of the individual classification models on the validation set

According to Table 1, two CNN models with different pre-trained embeddings returned similar performances, but overall performed poorly compared to the BERT model. Even though BERT pre-training data did not contain either tweets or COVID-19 related content, it performed much better due to its better learning capacity.

Similar to the prior literature on text classification, ALBERT, which had been trained on the same corpora as BERT, outperformed BERT in this task as well. CT-BERT which had been trained over a much smaller but more domain-specific dataset (COVID-19 related tweets) has outperformed both BERT and ALBERT in this classification task. Once again, this shows the useful effects of using more specific data collections even in unsupervised pre-training.

3.6 Further Analysis

In order to see whether these models make the same mistakes or not, models' predictions on validation set instances were analyzed. Figure 1 contains examples from the validation set which were misclassified by any one of the models. In the figure, each row represents one instance, black signifies the Informative class examples and pink for the

[2] http://nlp.stanford.edu/data/wordvecs/glove.twitter.27B.zip

[3] https://github.com/FredericGodin/TwitterEmbeddings

[4] https://huggingface.co/bert-base-uncased

[5] https://huggingface.co/albert-base-v2

[6] https://github.com/digitalepidemiologylab/COVID-twitter-bert

Uninformative examples. Gold column is the true labels and other columns represents the individual classification models.

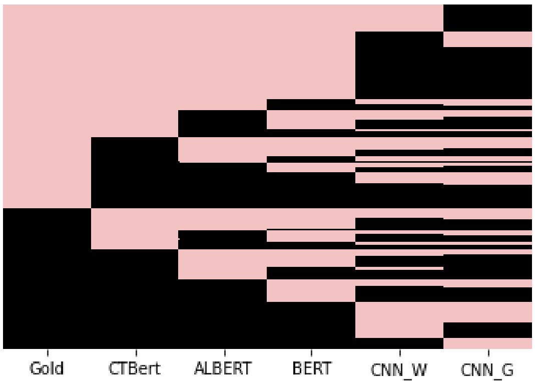

Figure 1: This figure contains all examples from the validation set that were misclassified by any of the listed models.

In Figure 1, one can see the similarity between the predictions made by BERT-based models: BERT, CT-BERT and ALBERT. CNN-based models (Word2Vec and Glove) also made similar mistakes which none of the BERT-based models did. We can also observe that CNN models, BERT and ALBERT have made the correct classification on some examples where our best model CT-BERT has failed. Combining strengths of these different models can improve the decision on these misclassified examples.

We also analyzed the possible performance variance in these models. BERT-base models which are fine-tuned on small data collections suffer from variance in performance. This is due to the randomly initialized weights in the final prediction layer built on top of BERT. Changing the random seed or slightly modifying the input can have an observable effect on the performance. To analyze this effect we trained 5 models with identical hyperparameters and different initial random seeds for all our BERT-based models. Performances of each model and each run on the validation set are reported in Table 2. We can see the difference between the best performing ALBERT model and the worst ALBERT model is more than 0.02 points. Slightly less but similar variance also exists for different BERT and CT-BERT models.

This variance between different versions of the same model, and different architectures having different strengths let us towards ensembling models.

	BERT	**ALBERT**	**CT-BERT**
	0.8833	0.8953	0.9115
	0.8789	0.8884	0.9083
	0.8742	0.8879	0.9070
	0.8722	0.8806	0.9050
	0.8669	0.8750	0.8990
Mean	0.8751	0.8854	0.9061

Table 2: The F1 Scores of 5 different randomly initialized BERT, ALBERT and CT-BERT models on the validation set.

4 Ensemble Methods

In this section we describe our two approaches to address the issue of variance in model performances and improve the overall classification accuracy. Our approaches focus on ensembling the probabilities of multiple classification models while making predictions.

4.1 SUM Ensemble

According to Figure 1, each individual model is predicting wrong labels for some instances, in which the other models are doing right. Even our best model CT-BERT is missing some cases which are correctly classified by other models. In order to decrease the amount of these easy and model specific misclassifications, we applied an ensemble approach to combine these different models.

In this ensembling strategy, each model is trained separately. When it is time for prediction, each model's prediction class probabilities are summed up. Then the final prediction label is chosen based on these combined scores. Such an ensemble strategy is especially useful for cases where some models are not certain of their predictions (probabilities around 0.5). In these cases other more certain models step up to make a more confident and hopefully correct prediction.

For this ensembling approach, in order to keep results more balanced 2 CNN-based models and 2 BERT-based models were used. BERT, which is the worst performing BERT-based model, was kept out in this ensembling, and only CT-BERT, ALBERT, CNN W2V and CNN GLOVE models were used.

4.2 SUM CT-BERT Ensemble

The motivation behind this approach is to reduce the variance in performance as shown in Table 2. As reported in Xu et al. (2020), fine-tuning multiple BERT models with different random seeds and ensembling their probability outputs can reduce variance in the overall classification performance. Our work here is a replication of the same work of Xu et al. (2020).

Our best performing model, CT-BERT, was used for this ensembling. 5 CT-BERT models trained with the same hyper-parameters but initialized with different random seeds were used. The individual performances of these models were already reported in Table 2. Similar to our SUM Ensemble approach, individual prediction probabilities were retrieved from each model, and then summed up. The final prediction was obtained from this total score.

4.3 Experiments

These two ensemble models' outputs were used as our system submissions. In Table 3, we report the performances of ensemble models on the validation and test sets.

Model	Val Score	Test Score
SUM Ensemble	0.9087	0.8790
SUM CT-BERT Ens	0.9106	0.8880

Table 3: The F1 Scores of ensemble models on validation and test set

Both ensemble models outperformed the best classifier model (CT-BERT) on validation set, and SUM CT-BERT Ensemble performed slightly better than SUM Ensemble on both validation and test set. In order to understand the specific errors these models made, Figure 1 was replicated, but this time including the two ensembles.

In Figure 2, we can clearly observe that SUM Ensemble is slightly better than CT-BERT. Considering the ensemble strategy, one conclusion we can arrive is that CT-BERT is probably making very confident predictions (probabilities close to 1 or 0) which makes it hard for the other three models to change CT-BERT's predicted label. This overconfidence of CT-BERT was helpful in some cases, but not all the times. This is something we will explore more in the future.

In the case of SUM CT-BERT, we see that ensembling multiple CT-BERT models has better performance than 4 out of the 5 CT-BERT models used in the ensembling process (reported in Table 2). The ensemble score is also better than the mean F1 score of these models (0.9062). Thus, reducing the variance caused by random initialization is generally useful.

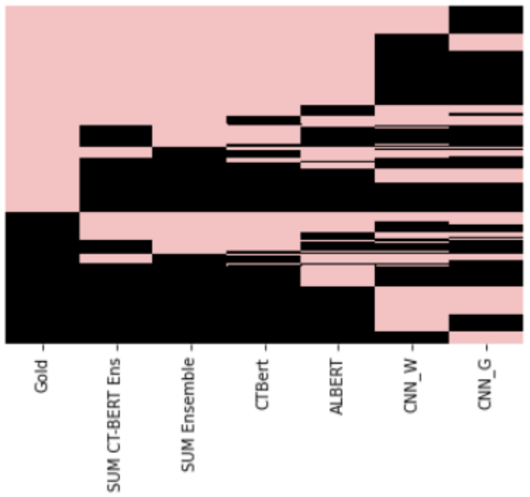

Figure 2: This figure contains all examples from the validation set that were misclassified by any of the listed models and ensembles.

One thing we would like to note is that all of our models were trained using the official training dataset (70% of the data) only. We did not use validation data (10% of the data) for training when submitting our predictions for the test set. This was an oversight on our part. We believe that training all models on validation set in addition to the original training set, would have yielded better results.

5 Conclusion

In this paper, we presented our work for W-NUT 2020 Task 2: Identification of Informative COVID-19 English Tweets. We experimented with BERT-based models and CNNs as our classification models. After analyzing the classification performance of these different models, we decided to use two different ensembling methods. Both of these ensemble strategies outperformed the individual models. More specifically, the ensemble strategy which addressed the variance in model performance caused by the random initialization, returned the best performance.

References

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Yoon Kim. 2014. Convolutional neural networks for sentence classification.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*.

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

Dat Quoc Nguyen, Thanh Vu, Afshin Rahimi, Mai Hoang Dao, Linh The Nguyen, and Long Doan. 2020. WNUT-2020 Task 2: Identification of Informative COVID-19 English Tweets. In *Proceedings of the 6th Workshop on Noisy User-generated Text*.

Yige Xu, Xipeng Qiu, Ligao Zhou, and Xuanjing Huang. 2020. Improving bert fine-tuning via self-ensemble and self-distillation.