LynyrdSkynyrd at WNUT-2020 Task 2: Semi-Supervised Learning for Identification of Informative COVID-19 English Tweets

Abhilasha Sancheti[*]
University of Maryland
College Park, USA
sancheti@cs.umd.edu

Kushal Chawla[*]
University of Southern California
Los Angeles, USA
kchawla@usc.edu

Gaurav Verma
Adobe Research
Bangalore, India
gaverma@adobe.com

Abstract

In this work, we describe our system for WNUT-2020 shared task on the identification of informative COVID-19 English tweets. Our system is an ensemble of various machine learning methods, leveraging both traditional feature-based classifiers as well as recent advances in pre-trained language models that help in capturing the syntactic, semantic, and contextual features from the tweets. We further employ pseudo-labelling to incorporate the unlabelled Twitter data released on the pandemic. Our best performing model achieves an F1-score of 0.9179 on the provided validation set and 0.8805 on the blind test-set.

1 Introduction

As of October 8, 2020, the novel coronavirus, SARS CoV-2, has infected 35.8 million people and led to over 1 million deaths [1]. With several stay-at-home orders and lockdowns in place, people have spent considerably more time than usual in disseminating and consuming information through social media. A subset of this publicly shared information relates to COVID-19, and ranges from information about new cases and deaths, lapses in policies made by local and state agencies, denied access to testing, etc. Since well-designed tools that monitor such information can help in quick identification and response from concerned agencies, recent research has focused on automatically detecting information and events related to COVID-19. For instance, Zong et al. (2020) aim to identify events on Twitter that relate to a new reported case or death, lapses in testing facilities, and information about cure and prevention, Lopez et al. (2020) analyze tweets to understand the perceptions of COVID-19 policies and how they change over time.

In the same vein as above, the 2020 edition of Workshop on Noisy User-generated Text (W-NUT 2020) hosted a shared task on 'Identification of Informative COVID-19 English Tweets'. The task involves automatically identifying whether an English Tweet related to COVID-19 is 'informative' or not. For a tweet to be considered informative in this context, it should provide information about recovered, suspected, confirmed, and death cases as well as location or travel history of the cases. The goal for developing such an automated system is to help track the development of the COVID-19 outbreak and to provide users the information related to the virus, e.g. any new suspicious/confirmed cases near/in the users' regions.

Aligned with the goals of this shared task, our paper details the use of state-of-the-art natural language processing techniques for identifying informative COVID-19 tweets. We experiment with a variety of methods, ranging from feature-based classifiers to leveraging recent advances in pre-trained neural architectures (Section 3). To further improve performance, we incorporate unlabelled tweets released on COVID-19 via masked language modelling and pseudo-labelling techniques. Our best performing model is an ensemble that uses Logistic Regression to combine the output probabilities of several base classifiers (Section 4). We further analyze the impact of pre-processing and semi-supervision through ablation studies. Through our qualitative adversarial analysis, we show how the predictions of BERT model are sensitive towards specific tokens such as 'confirmed case' or even locations and numerals, which also guides our data pre-processing steps.

2 Data Preprocessing

We conduct classification experiments on the COVID-19 Tweets dataset (Nguyen et al., 2020)

---

[*] Denotes equal contribution.
[1] https://covid19.who.int

provided for the shared task. The data split consists of 7000 tweets for training, 1000 in the validation set and 2000 in the blind test-set.

We start by lowercasing all the tweets, and replacing all urls with 'httpurl' and all usernames with '@user'. We then normalize all characters and pictograms. Additionally, we remove bad symbols (e.g. from a different language), any html tags, duplicated symbols or characters (like dots, question marks, special symbols, dashes or exclamations), and unnecessary underscores in the tokens. We also isolate punctuations, expand contractions (e.g. there'll to there will), and replace leet alphabets with their correct English versions using leet vocabulary[2]. We convert emojis into their text form [3] and finally normalize the different ways in which 'COVID-19' is written (e.g., covid-19, covid2019, covid19, or covid 2019) by replacing each occurrence with 'covid19'. These steps help to limit the vocabulary for better learning. On top of the above steps, we create different versions of the dataset as described below.

Cleaned: No further processing is done.

NUM-replaced: All the numerals in the dataset (except 19 in 'covid19') are replaced by NUM followed by the number of digits in the number.

LOC-replaced: We use the Python Spacy library[4] to get the named entity tags for each token in a tweet and replace the token having a 'GPE' (geopolitical entities i.e. countries, states, or cities) tag with 'LOC', so that the informativeness of a tweet is agnostic of a particular location.

NUM-LOC-replaced: We replace both numerals and tokens with 'GPE' tags in this version.

The motivation behind different versions of the dataset comes from our analysis (presented in Section 4) of the model trained on the **Cleaned** version of the dataset. We show how the model predictions are sensitive towards locations and numerals in a tweet, as a result of the patterns in the dataset. Hence, using NUM and LOC helps mitigate these issues. Following this pre-processing, we train a variety of machine learning models to classify tweets into two classes: informative and uninformative. We use the data split provided by the task organizers throughout to ensure consistency.

3 System Description

The system submitted for the shared-task is an ensemble of a variety of machine learning models which we discuss below in detail.

3.1 Fine-tuning pre-trained models

Motivated by their state-of-the-art performance on several downstream natural language processing tasks, we use pre-trained models like BERT (Devlin et al., 2018) for our classification task. BERT model learns a contextual representation of an input text which helps in representing the semantic information contained in an input. Borrowing notations from Sun et al. (2019), we leverage the pre-trained model in the following ways.

BERT-FiT: Following the standard fine-tuning pipeline (Devlin et al., 2018), the model is initialized with a pre-trained architecture and is further **Fi**ne-**T**uned on the provided labelled dataset.

BERT-ITPT-FiT: Inspired by the improvements using with**In T**ask **P**re-**T**raining (Sun et al., 2019), we further train the out-of-the-box pre-trained model on our training dataset using the masked language modeling (MLM) objective for 3 epochs. Thereafter, the model is fine-tuned for classification using the associated ground-truth labels.

BERT-IDPT-FiT: Extending the above approach, we extract a total of 27,388 tweets related to the pandemic using the Twitter API[5] for **I**n-**D**omain **P**re-**T**raining. We use the tweet IDs released with the Covid-19 tweets dataset (Lamsal, 2020) and the WNUT shared task on Event Extraction[6]. The extracted tweets are pre-processed in the same manner as described in Section 2 and augmented with the training set to further train the pre-trained model on the MLM loss for 5 epochs. Once trained, we then fine-tune the model on the classification task using the provided labelled training set. As we show later (Table 1), leveraging additional in-domain tweets further helps in improving the performance on our task.

We primarily use 'bert-base-uncased' architecture for all our experiments and train separate models[7] for each version of the dataset as well as each variant of the BERT model. All the models are fine-tuned for 20 epochs with the maximum input length set to 100, batch size to 8, and dropout to 0.3. The model having best F1-score for the 'Informative' class on the validation dataset is used for evaluation.

We also experimented with other pre-trained models: BERT-Large (Devlin et al., 2018), RoBERTa (Liu et al., 2019), XLNet (Yang et al., 2019), and Covid-BERT (Müller et al., 2020) but did not find any visible difference in performance. Hence, we restricted our experiments to BERT-base owing to its smaller size and consequently, lesser consumption of computational resources.

3.2 Classification using fastText (fastText)

Joulin et al. (2017) proposed fastText – a simple and efficient baseline for text classification that uses bag of n-gram features to capture partial information about the local word order. We use the fastText library to train a classifier for our task. We train the model for 10 epochs with a learning rate initialized at 0.1 and the maximum length of n-grams is set to 3.

3.3 Most-frequent N-gram features (ngram)

We also build our own implementation of n-gram features. Using the **NUM-LOC-replaced** version of the dataset, we extract the most frequent 5000 unigrams, bigrams, and trigrams for each class and use the presence or absence of these n-grams in a given tweet, as features. These features are used as input to a feed-forward neural network, with two hidden layers of size 64, dropout of 0.1, and activation as ReLU. Although the feature vector is highly sparse, we find this approach to perform reasonably well on our task.

3.4 Leveraging the Universal Sentence Encoder (USE)

We leverage the Universal Sentence Encoder (Cer et al., 2018) to get sentence embeddings of the tweets. These 512-dimensional embeddings are used as input features to a feed-forward neural network identical to the one described above for

n-gram features. Sentence embedding based classifiers also take into account the contextual (short-range as well as long-range) information of each word as well as ordering of the words, unlike the n-gram based models, performing well in practice.

3.5 Hand-crafted features (HCF)

Next, leveraging the advances in affective computing, we build a total of 150 hand-crafted features, comprising both syntactic and affect features. Syntactic features include statistics from the tweet based on the counts of punctuation marks along with NER and POS tags. We also include text readability metrics from the textstat toolkit[8]. On the other hand, affect features are based on existing lexicons in the literature. Specifically, we use Warriner VAD wordlists (Warriner et al., 2013), formality word-lists (Brooke et al., 2010), PERMA model (Seligman, 2012), Temporal word lists (Park et al., 2015), EmoLex (Mohammad et al., 2013), and lastly, LIWC (Pennebaker et al., 2001). All the features are concatenated to form a feature vector for each tweet which is used as an input to the feed-forward network described previously.

3.6 Employing Pseudo-Labelling (PL)

In order to leverage the unlabelled data for feature-based methods, we resort to pseudo-labelling. The models are trained for at most 50 epochs in the following manner: After every 10 epochs, we select the checkpoint with the best F1-score for the 'Informative' class saved till now and use it to predict labels on the collected unlabelled tweets (same as described in Section 3.1). We randomly pick 1000 tweets on which the prediction confidence is above 0.99 and consider the predicted labels for these tweets as pseudo ground-truth labels. Hence, these 1000 tweets are removed from the set of unlabelled data and instead added to the training dataset for future epochs. The training continues from the best checkpoint found till now.

In Section 4, we show improvements due to the use of pseudo-labelling across all the methods based on fixed feature vectors. We also tried pseudo labelling with BERT. However, using MLM to incorporate unlabelled tweets (**BERT-IDPT-FiT**), performs better than pseudo-labelling in that case.

---

[7]We use huggingface library available at `https://github.com/huggingface/transformers` for fine-tuning of pre-trained models.

[8]`https://pypi.org/project/textstat/`

Model	Accuracy↑	Precision↑	Recall↑	F1-score↑
Majority	0.5280	0.0000	0.0000	0.0000
HCF-PL	0.7460	0.7410	0.7104	0.7252
USE-PL	0.7756	0.7196	0.8601	0.7834
fastText	0.8083	0.8372	0.7373	0.7840
ngram-PL	0.8090	0.7932	0.8072	0.7997
BERT-FiT	0.9010	0.8768	0.9201	0.8977
BERT-ITPT-FiT	0.9010	0.8716	0.9272	0.8984
BERT-IDPT-FiT	0.9070	0.8736	**0.9392**	0.9050
Ensemble	**0.9209**	**0.9034**	0.9322	0.9176
Ensemble-TT	**0.9209**	0.9002	0.9364	**0.9179**

Table 1: Performance of various models on Precision, Recall and F1-score for Informative class and model accuracy on the validation dataset. The numbers are averaged over 3 runs. BERT-based and fastText models were trained on different pre-processed versions of the dataset. We report the scores for the corresponding best performing configuration above.

3.7 Ensemble Model

Our final model is an ensemble of the base classifiers described above. To build an ensemble, we train a logistic regression model using the output probabilities from a subset of the base classifiers, as features. Further, we tune the threshold above which the model would output the class as 'Informative', and otherwise output 'Uninformative'. The ensemble model without any threshold tuning is referred to as **Ensemble**. The model with threshold tuning is referred to as **Ensemble-TT**, that uses a threshold value of 0.5168 instead of 0.5.
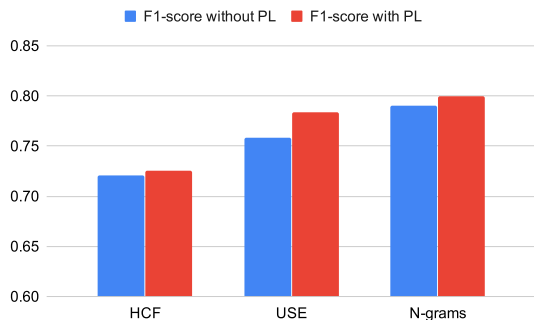
4 Results and Analysis

Figure 1: Effect of pseudo-labelling on F1-score for the Informative class. Pseudo-labelling improves the performance for all three methods.

The primary evaluation metric for the shared task is F1-score for the 'Informative' class. The other metrics used are the Precision and Recall for Informative class and the overall accuracy of the model.

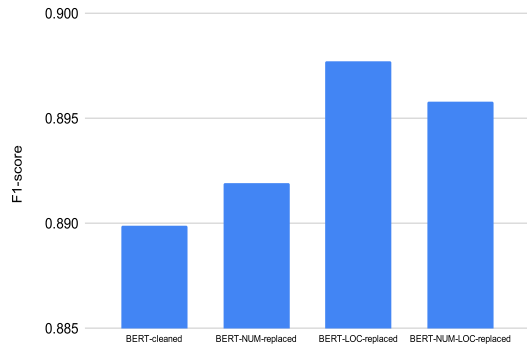We first illustrate the benefit of employing

Figure 2: Impact of different ways of data pre-processing on F1-score using the **BERT-FiT** model.

pseudo-labelling in Figure 1. All three methods, namely **HCF**, **USE** and **ngrams** achieve better F1-scores by leveraging the unlabelled data, along with the provided training dataset. Hence, we next compare these models with other baselines and BERT-based pre-trained methods in Table 1. All models easily beat the majority baseline, attesting that the models are learning useful patterns from the data. While feature-based methods perform reasonably well with **ngram-PL** achieving an F1-score close to 0.8, they are outperformed by BERT-based methods with approximately a 10% gain. High performance of **BERT-IDPT-FiT** shows that incorporating additional in-domain data using the MLM objective improves the performance. Finally, an ensemble using logistic regression achieves the best precision, resulting from a reduction in the number of false positives, while still maintaining the same recall. Our best performing model, **Ensemble-TT** achieves an F1-score of 0.9179 on the validation dataset and 0.8805 on the blind test-set.

Input: the taslee palm city estate in maitama , abuja , has alerted ...	Prediction
... its residents.	0
... its residents of a case .	1
... its residents of a confirmed case .	1
... its residents of coronavirus.	0
... its residents of a case of coronavirus .	1
... its residents of a confirmed case of coronavirus .	1
... its residents of malaria.	0
... its residents of a case of malaria .	1
... its residents of a confirmed case of malaria .	1

Table 2: **BERT-FiT** predictions on artificially-created examples, inspired from an instance in the validation dataset. 0 (1) refers to Uninformative (Informative) class.

Qualitative Adversarial Analysis: In Table 2, we show the sensitivity of **BERT-FiT** model towards specific tokens such as 'confirmed' and 'case' in

Input	Prediction
his family members got infected in santa clara .	1
his family members got infected in rohini .	0
5 family members got infected in rohini .	1
5 family members are healthy in rohini .	1

Table 3: **BERT-FiT** predictions on artificially-created examples, depicting bias towards numerals and locations. 0 (1) refers to Uninformative (Informative) class.

the tweets. Using these tokens governs the output predictions, regardless of whether the tweet is talking about covid-19 or an arbitrarily chosen disease, malaria. This is justified since the dataset mostly contains the tweets related to the pandemic, but suggests to exercise caution while using such models for downstream monitoring applications. Further, in Table 3, we show sensitivity towards numeric and location tokens in the tweets. Mere change of location to a less frequent one in the dataset or use of numerals inverts the model predictions, regardless of whether the tweet is actually informative or not. This observation infact inspires our data-pre-processing stages, where we mask the numeric and location tokens from the dataset. We investigate this further in Figure 2, which establishes that effective pre-processing can help mitigate these biases, while keeping the performance at-par or better on our classification task.

5 Conclusion

In this paper, we describe our system to identify informative COVID-19 English tweets. We find that an ensemble model which uses a logistic regression to combine the predictions of a variety of feature-based to neural methods achieves the best performance on the shared task. Our analysis shows that incorporating unlabelled tweets results in consistent performance gains. We show how the trained model can be sensitive to specific tokens in the tweets, and hence, advice for exercising caution while deploying machine learning models for downstream monitoring applications.

References

Julian Brooke, Tong Wang, and Graeme Hirst. 2010. Automatic acquisition of lexical formality. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Armand Joulin, Édouard Grave, Piotr Bojanowski, and Tomáš Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*.

Rabindra Lamsal. 2020. Coronavirus (covid-19) tweets dataset.

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. *arXiv preprint arXiv:1907.11692*.

Christian E Lopez, Malolan Vasu, and Caleb Gallemore. 2020. Understanding the perception of covid-19 policies by mining a multilanguage twitter dataset. *arXiv preprint arXiv:2003.10359*.

Saif Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*.

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

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*.

Gregory Park, H Andrew Schwartz, Maarten Sap, Margaret L Kern, Evan Weingarten, Johannes C Eichstaedt, Jonah Berger, David J Stillwell, Michal Kosinski, Lyle H Ungar, et al. 2015. Living in the past, present, and future: Measuring temporal orientation with language. *Journal of personality*.

James W Pennebaker, Martha E Francis, and Roger J Booth. 2001. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*.

Martin EP Seligman. 2012. *Flourish: A visionary new understanding of happiness and well-being*. Simon and Schuster.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*. Springer.

Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. 2013. Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior research methods*.

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

Shi Zong, Ashutosh Baheti, Wei Xu, and Alan Ritter. 2020. Extracting covid-19 events from twitter. *arXiv preprint arXiv:2006.02567*.