IRLab@IITBHU at WNUT-2020 Task 2: Identification of informative COVID-19 English Tweets using BERT

Supriya Chanda
IIT BHU
INDIA
supriyachanda.rs.cse18
@itbhu.ac.in

Eshita Nandy
Gauhati University
INDIA
nandyeshita4
@gmail.com

Sukomal Pal
IIT BHU
INDIA
spal.cse
@itbhu.ac.in

Abstract

This paper reports our submission to the shared Task 2: Identification of informative COVID-19 English tweets at W-NUT 2020. We attempted a few techniques, and we briefly explain here two models that showed promising results in tweet classification tasks: DistilBERT and FastText. DistilBERT achieves a F_1 score of 0.7508 on the test set, which is the best of our submissions.

1 Introduction

The WWW and then Web 2.0 enabled people to express their views and opinions through blog posts, online forums, product review websites, and social media. Millions of people use social network sites like Facebook, Twitter, LinkedIn, and Google Plus to express their emotions, opinions, and share views on different issues that matter in their lives. The present pandemic situation has severely reduced physical interaction and forced us to use virtual platforms where people inform and influence others. Social media generates a large volume of sentiment-rich data every day through tweets, status updates, blog posts, comments, reviews and so on. We often depend upon such user-generated content online to a great extent for decision making.

As the pandemic is spreading worldwide, online media is flooded with uncontrolled user-generated content, lot of which are not at all moderated or quality-checked. It, therefore, becomes more important to understand this spread, get aware of the differences between informative and uninformative news, and act upon it wisely.

Coronavirus causes illness, which can vary from common cold and cough to sometimes more severe disease. SARS-CoV-2 (n-coronavirus) is the new virus of this family, which has led to more than 1 million deaths worldwide[1].

Regularly, we find thousands of tweets by users worldwide, expressing their sentiments on COVID-19. Although these tweets often carry valuable information, they are unstructured and, are, therefore, challenging to process. So, analyzing these tweets is quite essential to understand whether they are informative ones or not.

We used a few machine learning approaches to classify test tweets as either informative or non-informative. Before the classification, we cleaned the tweets, constructed a representation of tweets with different word embedding techniques, and then built the classification model.

2 System Description

2.1 Datasets

The W-NUT shared task[2] organizers provided a dataset (Nguyen et al., 2020) that consists of 10K COVID-19 English tweets[3], that included 4719 tweets labeled as informative and 5281 tweets, labeled as uninformative. The statistics of training, development, and test data corpus collection and class distribution are shown in Table 1. Here, each tweet is annotated by three independent annotators, and an inter-annotator agreement score of Fleiss' Kappa at 0.818 is obtained. Some tweet examples from the training dataset are shown in Table 2.

2.2 Data Pre-processing

The Twitter dataset used in this work is already labeled into two classes: informative and uninformative. Before feeding them into classifier, tweets were pre-processed using the following steps:

[1] https://www.worldometers.info/coronavirus/

[2] http://noisy-text.github.io/2020/covid19tweet-task.html

[3] https://github.com/VinAIResearch/COVID19Tweet

Proceedings of the 2020 EMNLP Workshop W-NUT: The Sixth Workshop on Noisy User-generated Text, pages 399–403
Online, Nov 19, 2020.

Data	INF	UNI	TOTAL
Training	3303	3697	7000
Validation	472	528	1000
Test	944	1056	2000

Table 1: Training, Validation and Test Data set Collection and Class Distribution (informative as INF and uninformative as UNI)

- removal of the hashtag symbol (#) and all the user mentions (@USER)
- removal of stopwords using NLTK[4] library
- removal of Non-ASCII characters
- removal of all the emoticons, symbols, numbers, special characters.

2.3 Word Embedding

Word embedding is arguably the most widely known technique in the recent history of NLP. It captures semantic property of a word. We have used 200-dimension pre-trained GloVe[5] (Pennington et al., 2014) model, Word2Vec (Mikolov et al., 2013), and 300-dimension pre-trained FastText[6] (Mikolov et al., 2018) model for computing the word embedding. Also we have used `bert-base-uncased` and `distilbert-base-uncased` pre-trained models[7] to get a vector as an embedding for the sentence that we can use for classification.

- **Word2Vec:** Training tweets are used to train the model. Each word in the input text is characterized using N-dimensional vector. The dimension of the vector can be configured depending on the complexity of the text data. We used 200 dimensions. Tokens of each tweet are converted into a numerical vector of 200-dimensions before passing them to the model.
- **Glove:** GloVe (Global Vectors for Word Representation) is an extension to word2vec for efficiently learning word vectors. Unlike Word2vec, Glove does not depend on local statistics, i.e. local context information of words, but contain global statistics like word co-occurrences.
- **FastText:** FastText, developed by Facebook, combines certain concepts introduced by the NLP and ML communities, representing sentences with a bag-of-words and n-grams using subword information and sharing them across classes through a hidden representation.
- **BERT:** Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) is a technique for NLP pretraining developed by Google. BERT is pretrained on a large corpus of unlabelled text, including the entire Wikipedia (that is 2,500 million words!) and Book Corpus (800 million words). BERT-Base uncased have 12 layers (transformer blocks), 12 attention heads, and 110 million parameters.
- **DistilBERT:** DistilBERT (Sanh et al., 2019) is a smaller version of BERT developed and open-sourced by the team at HuggingFace. It is a lighter and faster version of BERT that roughly matches its performance. DistilBERT also compares surprisingly well to BERT on downstream tasks while having respectively about half and one third the number of parameters.

2.4 Classifiers

After pre-processing our data and transforming all the tweets into proper representation form, we implement our classification algorithms and construct our training models. We tried a few hand-picked algorithms, for instance, Logistic Regression, Support Vector Machine, XGBoost, Bidirectional Long Short-Term Memory (Bi-LSTM), and Gated recurrent units (GRU).

- **Support Vector Machine:** SVM, initially designed for binary classification, gives an excellent result for text categorization tasks such as sentiment analysis. Here we consider binary SVM for simplicity. SVM performs classification by finding an optimal hyper-plane that separates two classes. The optimal hyperplane has a maximum margin (the distance between the nearest data point and hyperplane is called a margin). The datapoint that lies nearest to the hyper-plane is called the support vector.

[4] https://www.nltk.org

[5] http://nlp.stanford.edu/data/glove.6B.zip

[6] https://fasttext.cc/docs/en/english-vectors.html

[7] https://huggingface.co/transformers/pretrained_models.html

Sample tweets from dataset	Label
Democrats somehow managed to fight ebola without calling it ”the African virus.” A cluster of COVID-19 cases has emerged in New York CIty’s Hassidic neighborhood, so it’s only a matter of time before the local Trump Klux Klan starts talking about ”the Jew virus.”	UNINFORMATIVE
@USER @USER 1 week ago today (March 14), there were only 115 cases of CoVid in FL. Today, last count was at 763... that is a 563% increase in 1 week due to your “slave to Trump” incompetence. Imagine next Saturday if there is another 563% increase-we will be at 9300+. Resign	INFORMATIVE

Table 2: Example tweets from the COVID dataset for both labels

- **Gradient Boosting:** XGBoost is a scalable machine learning approach that has proved to be successful in a lot of data mining and machine learning challenges.

- **Bidirectional Long-Short Term Memory:** BiLSTM combines bidirectional recurrent neural network models and LSTM units to capture the context information. The BiLSTM model treats all inputs equally. For the task of sentiment analysis, the sentiment polarity of the text largely depends on the words with sentiment information. Firstly, the weighted word vectors are used as inputs of the BiLSTM model. Then outputs of the BiLSTM model are used as the representations of the comment texts.

- **Gated Recurrent Unit:** The Gated Recurrent Units (GRU) is the newer generation of Recurrent Neural networks, similar to LSTM. GRU does not have any cell state and used the hidden state to transfer information. It has only two gates, a reset gate and an update gate. GRU is preferred over LSTM because it performs well in terms of time consumption and memory utilization. In this model, we have obtained better accuracy with the GRU layer.

- **Logistic Regression:** Logistic Regression is a classification algorithm, particularly suitable for binary classification, as it provides us a baseline model. In our case, it produces a high-accuracy result.

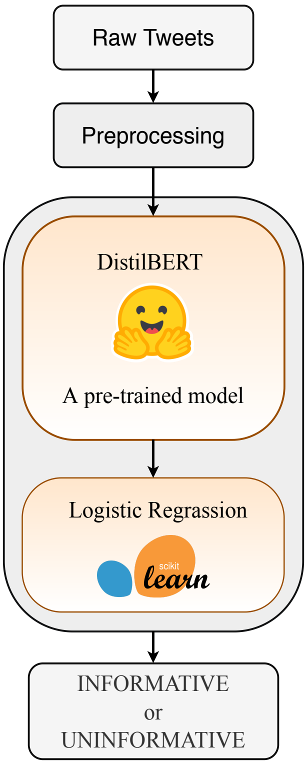

Figure 1: System 2 architecture

2.5 Hyper-parameter Settings

We did stopword removal for all the models except DistilBERT and BERT. For Glove+GRU model, the dimension of word embeddings was 300, the hidden units of GRU was 128, `Optimizer` = adam, `loss function` = binary cross entropy, `Dropout` = 0.5, `activation function` = sigmoid. For Glove+BiLSTM model, the dimension of word embeddings was 300, 3 dense layer `Optimizer` = adam, `loss function` = binary cross entropy, `Dropout` = 0.2, for the first two dense layer the was `activation function` relu and last dense layer it was sigmoid. For FastText+BiLSTM model, the dimension of word embeddings was 300, 3 dense layer `Optimizer` = adam, `loss function` = binary cross entropy, `Dropout` = 0.2, for the first two dense layer the was `activation function` relu and last dense layer it was sigmoid.

3 Results and Analysis

We have used `scikit-learn`[8] machine learning package for the implementation. Table 4 reports our results on validation dataset for both classes

[8] http://scikit-learn.org

(INFORMATIVE and UNINFORMATIVE). We have selected two models that performed the best F_1 score during the validation phase and submitted it for the final prediction on the test dataset.

We have observed that DistilBERT and LR (Fig 1) gave better F_1 score than others as shown in Table 3. As mentioned in the organizer's evaluation criteria, the three evaluation matrices are calculated only for the INFORMATIVE class on test data. On validation data, we have calculated all matrices for both of the classes and put it on Table 4. That is why the evaluation results shown in Table 4 on validation data are slightly deviating from the evaluation result shown in Table 3 on test data provided by organizers. In the training data, there are some ambiguous keywords like *HTTPURL*, *coronavirus*. These words were present in both classes, so the models learned on this ambiguous data and got confused on validation data.

System(#)	F_1	Pre	Rec	Acc
FastText + GRU (1)	.7361	.7908	.6886	.7670
DistilBERT + LR (2)	.7508	.7904	.7150	.7760

Table 3: Evaluation results on test set

The experimental results show that DistilBERT outperforms on this task over BERT and others embedding techniques. We compare two network architectures, BiLSTM and GRU. The results show that the GRU model performs better than the BiLSTM model on this task. FastText outperforms over word2vec and Glove because FastText treats each word as a composed of character n-grams. If words are rare their character n grams are still shared with other words - hence the embeddings can still be good. where word2vec and Glove both treat words as the smallest unit to train on.

4 Conclusion

This study reports our system for shared task 2: Identification of informative COVID-19 English tweets in W-NUT 2020. We performed a comparative analysis of Machine learning models: Logistic Regression (LR), Support Vector Machine (SVM), and XG-Boost (XGB), Deep learning models: BiLSTM, GRU with few embedding techniques: Word2Vec, GloVe, FastText, BERT, and DistilBERT. Based on a series of experiments, we find that DistilBERT outperforms on this task. However, there is room for improvement. In the future, we plan to use other pre-trained models with some fine-tuning.

References

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.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

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*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.

	Word2Vec + SVM				**Word2Vec + XGB**			
	Precision	Recall	F_1-score	support	Precision	Recall	F_1-score	support
UNINFORMATIVE	0.78	0.85	0.81	528	0.76	0.84	0.80	528
INFORMATIVE	0.81	0.83	0.77	472	0.80	0.70	0.74	472
macro avg	0.79	0.79	0.79	1000	0.78	0.77	0.77	1000
weighted avg	0.79	0.79	0.79	1000	0.78	0.77	0.77	1000
Accuracy	0.79				0.77			
	Glove + Bi-LSTM				**Glove + GRU**			
UNINFORMATIVE	0.79	0.86	0.82	528	0.87	0.73	0.79	528
INFORMATIVE	0.82	0.74	0.78	472	0.75	0.88	0.81	472
macro avg	0.80	0.80	0.80	1000	0.81	0.81	0.80	1000
weighted avg	0.80	0.80	0.80	1000	0.81	0.80	0.80	1000
Accuracy	0.80				0.80			
	FastText + Bi-LSTM				**FastText + GRU**			
UNINFORMATIVE	0.80	0.84	0.82	528	0.84	0.78	0.81	528
INFORMATIVE	0.81	0.76	0.78	472	0.77	0.83	0.80	472
macro avg	0.80	0.80	0.80	1000	0.81	0.81	0.80	1000
weighted avg	0.80	0.80	0.80	1000	0.81	0.81	**0.81**	1000
Accuracy	0.80				0.81			
	DistilBERT + LR				**DistilBERT + XGB**			
UNINFORMATIVE	0.80	0.85	0.83	528	0.75	0.82	0.78	528
INFORMATIVE	0.82	0.77	0.79	472	0.77	0.69	0.73	472
macro avg	0.81	0.81	0.81	1000	0.76	0.75	0.76	1000
weighted avg	0.81	0.81	**0.81**	1000	0.76	0.76	0.76	1000
Accuracy	0.81				0.76			
	BERT + LR				**BERT + XGB**			
UNINFORMATIVE	0.77	0.82	0.79	528	0.75	0.79	0.76	528
INFORMATIVE	0.78	0.72	0.75	472	0.74	0.70	0.72	472
macro avg	0.77	0.77	0.77	1000	0.74	0.74	0.74	1000
weighted avg	0.77	0.77	0.77	1000	0.74	0.74	0.74	1000
Accuracy	0.77				0.74			

Table 4: Precision, recall, F_1-scores, and support for all experiments on validation dataset