Pre-trained Model and Bi-LSTM Performance on Disaster Tweet Detection

Huijun Hao
hhao004@ucla.edu
UID: 105863846

Jingwei Huang
j284huan@ucla.edu
UID: 805525466

Syed Nawshad
snawshad@ucla.edu
UID: 805871517

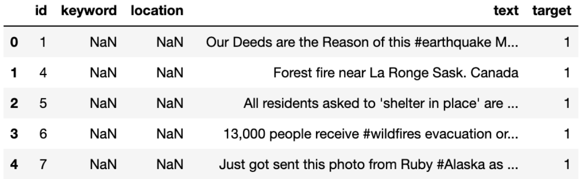
Yuefu Liu
liuyf@ucla.edu
UID: 405711899

Abstract

Abstract-In this paper we investigate the effect of LSTMs and pre-trained models such as BERT, RoBERTa and a couple of variations of those models on the effect of disaster tweet detection. The performance of the models used are documented based on F1 scores, a measure of accuracy in binary classification. Our paper finds that BERTweet obtained the best model performance of 85.9%.

1. Introduction

The data set we will be using to test model performance is from Kaggle's NLP disaster tweet classification competition. An example of some training data entries are shown below in **Table I.**



	id	keyword	location	text	target
0	1	NaN	NaN	Our Deeds are the Reason of this #earthquake M...	1
1	4	NaN	NaN	Forest fire near La Ronge Sask. Canada	1
2	5	NaN	NaN	All residents asked to 'shelter in place' are ...	1
3	6	NaN	NaN	13,000 people receive #wildfires evacuation or...	1
4	7	NaN	NaN	Just got sent this photo from Ruby #Alaska as ...	1

Table I. Training Data Entries

The input data consists of texts, keywords and locations. The output consists of a target 1 or 0, which corresponds to whether a tweet is a disaster tweet or not.

The pre-trained models are developed in two training steps: a semi-supervised training step, where the model is pre-trained with a large corpus with some words masked and a supervised training step, where the model is fine-tuned for a specific task.

1.1. BERT

One of the pre-trained models utilized in this paper is BERT (Bi-directional encoder representations with transformers). The BERT model consists of stacked Transformer encoders that process tokens in full context before and after.

1.2. RoBERTa

RoBERTa (Robustly optimized BERT approach), a variant of BERT, is trained longer, with bigger batches over more data, on longer sequences and dynamically changing the masking pattern. It also has the next sentence prediction objective removed from BERT. RoBERTa is shown to outperform BERT-large on GLUE and SQuAD tasks.

1.3. BERTweet

BERTweet is a pre-trained large-scaled language model for English Tweets. It uses the same architecture as BERT and is trained based on RoBERTa pre-training procedure. This model is designed to specialize in Tweet NLP tasks including POS tagging, NER, and text classification.

1.4. Bi-LSTM

Bi-LSTM (Bi-directional long short-term memory), built on the LSTM model, contains sequences of data with backward (future to past) and forward (past to future). The model used on tagging sequence type data, such as part of speech tagging task in natural language processing: given a sentence, the sentence is a natural sequence of words, tagging words for a certain purpose. The LSTM hidden layer states of the two layers propagate forward and backward respectively. Combined with the output of two-layer LSTM, the final annotation result is obtained.

2. Experiments

2.1. Preprocessing Performed

Some of our models used additional pre-processing. The BERT model used additional emoji translations. A dictionary was created specifically for each emoji and the meaning of the emojis was used for the BERT model to train.

We have trained the RoBERTa model with and without preprocessing such as removing the URL, html-style text and emojis in the training data. However, the performance has changed only insignificantly. The model performs generally well even without any preprocessing, hence the results presented in this report is without additional preprocessing. We also find that BERTweet (based on RoBERTa) performs decent preprocessing, so we can compare the performance between vanilla RoBERTa and this specific one.

BERTweet model provides an internal normalizer for pre-processing. By enabling the normalization argument of its tokenizer, it translates the user mentions and web/url links into special tokens @USER and HTTPURL respectively. It also converts emotion icons into strings using the emoji package and normalizes the word tokens. Therefore, we don't apply pre-processing procedures for this model since it's already embedded in the tokenizer.

LSTM used multiple preprocessing methods. In order to capture more useful information, for the data cleaning part I removed HTML, emoji and punctuation from the training data set. Normalization method applied on text training to improve the accuracy. According to the comparison, with data-preprocessing model accuracy is 83.93% and without data-preprocessing model accuracy is 92.4% which indicated that data cleaning might remove some useful information from the training set.

2.2. Hyperparameters

In order to fine tune the model for our task and achieve a better performance, we have tried different combinations of hyperparameters.

Both RoBERTa and BERTweet model's performance benefit from the increase in batch size. Due to the memory limit and training time, we choose the batch size as 32 for RoBERTa and 80 for BERTweet. BERT was run for 4 epochs, RoBERTa was run on 3 epochs, and BERTweet was run for 5 epochs.

We also explored different learning rate and weight decay parameters. BERT, RoBERTa and BERTweet both use the 10^{-5} learning rate and 0.01 weight decay.

For the LSTM model both bidirectional LSTM and GRU are used with global max pooling. The vanilla hidden layer applied to reduce the variance loss also avoids overfitting. Since the label target is 0 or 1, the activation function is relu. Rmsprop was chosen as the optimizer because cumulative gradient is calculated by exponentially weighted moving average method to discard the distant gradient history information (the weight of the reduced learning rate of the gradient farther from the current is smaller). The value of dropout is 0.5, learning rate is 0.01 and batch size is 256.

3. Results

The summary of our results are shown in **Table II**. The BERTweet model achieved the highest F1 score. RoBERTa's peak performance varied from 83% to about 85% and peaked with 3 epochs.

Preprocessing on emoji tokens can improve the model performance. BERT small obtained about 81% F1 score and improved about 1-2% on average with emoji processing. BERTweet is trained based on RoBERTa procedures with preprocessing for emoji tokens as well and appears to have similarly increased in performance by roughly 2%. Bi-LSTM and Bi-GRU model with 4 epochs has binary classification accuracy 83.93%. The preprocesses part removed valid information to avoid a model to capture sequence of data.

Model	F1 score
BERT	81.2280%
BERT+emoji processing	82.2081%
RoBERTa	83.2392%
BERTweet	85.8607%
Bi-LSTM	83.93% (binary classify accuracy)

Table II. Model Performance Results

The correlation between F1 score and epochs for the best-performed model, BERTweet was shown in **Fig 1**. The model converges quickly and F1 score achieves about 85% in the first epoch. The fluctuation of F1 score might be due to the overfitting problem.

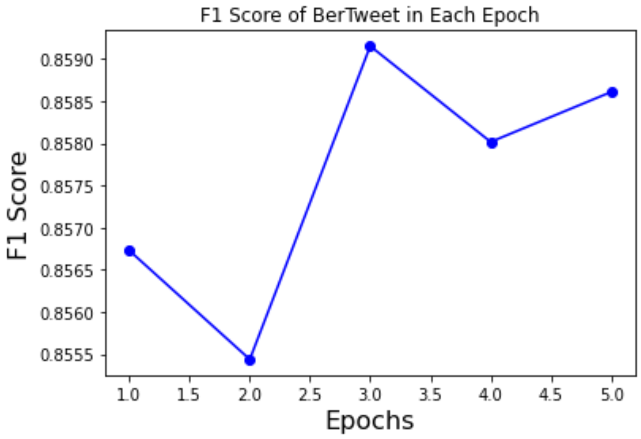


Fig 1. Correlation between F1 score and epochs

4. Conclusion

The BERTweet model outperformed all other models tested in this paper. The result is reasonable since BERTweet is a fine-tuned model specially designed for tweet dataset. It outperformed the RoBERTa as expected, which matched the results in Nguyen's work[3]. This is most likely due to

BERTweet's pretrained model being specialized for tweets and taking into account URLs and emojis.

BERT didn't improve a significant amount from the use of emoji processing. This is most likely due to the fact that BERT's pre-trained model is more generally fitted to any transfer learning that occurs afterwards for fine-tuning. As a result, adding in emoji processing does not add any significant improvements.

Bi-LSTM did better without preprocessing. The issue is the data clearing part removed multiple parts like: punctuation, emoji, html, URL etc. Some of the data patterns might be included there. Removed all others except for text would lack information capture which affects model performance.

5. Reference

[1] BERT+emoji processing: Disaster Tweet Submission. https://www.kaggle.com/code/ritheshsreenivasan/disastertweetv1/notebook?scriptVersionId=50824869. Sreenivasan, Rithesh. 2020.

[2] RoBERTa: Liu Y, Ott M, Goyal N, et al. Roberta: A robustly optimized Bert pretraining approach. arXiv.org. https://arxiv.org/abs/1907.11692. Published July 26, 2019. Accessed May 16, 2022.

[3] BERTweet: A pre-trained language model for English Tweets, Nguyen et al., EMNLP 2020

[4] Bi-LSTM with and without propcecessing https://www.kaggle.com/code/boshili/textvectorization-bi-lstm