Machine Learning Engineer Nanodegree

Capstone Proposal

Topic: Real or Not? NLP with Disaster Tweets[1]

Faraz Alam Ansari
January 20, 2020

Proposal

Domain Background

Twitter is undoubtably one of the most popular social media platforms. It is used globally not only to express opinions, ideas and excitement but also as a means of effective communication. There is an increasing use of Twitter these days as a means of communication during emergency situations like natural calamities or other disasters. High accessibility to smartphones almost round the clock enables and encourages people to share or announce the state of emergency they are observing in real time. As a result, many disaster relief agencies are eager to programmatically monitor Twitter so as to get alerts in a real time during disaster scenarios. That's why an efficient algorithm to pick disaster tweets from the mountains of tweets every day is definitely a necessity in order to detect and identify disaster outbreak in no time, which could definitely reduce losses to human life or property by several folds.

Problem Statement

The objective of this project is to develop an algorithm to predict which tweets are about real disasters and which ones are not.

When provided with a new unseen tweet the algorithm should be able to correctly identify if it as a real Disaster tweet or not. This algorithm should ideally be able to provide a score or probability of the tweet being a real disaster tweet. This is basically a Natural Language Processing (NLP) problem and is picked from a current Kaggle competition[1].

Datasets and Inputs

The dataset I will be using is the same as suggested in the Kaggle competition. It's a dataset of 10,000 tweets that were hand classified. This dataset was created by the company **figure-eight** and is publicly available at their 'Data for everyone' website[2].

As per the data-set details provided at the website – the data contributors looked at over 10,000 tweets culled with a variety of searches like "*ablaze*", "*quarantine*", and "*pandemonium*" and then noted whether the tweet referred to an actual disaster event (as opposed to a joke with the word or a movie review or something non-disastrous). We will be using this labelled data for this project to develop a model to efficiently classify tweets as disaster tweets or otherwise.

Solution Statement

I plan to use the powerful new language classification model – **BERT** (**B**idirectional **E**ncoder **R**epresentations from **T**ransformers)[3] which has recently gained much popularity in the area of Natural Language Processing to solve our Disaster tweets classification problem. BERT is an open source model provided by Google. A pre-trained BERT model can be finetuned with just one additional output layer to create state-of-the-art models for a wide range of tasks without the need of substantial task specific architecture modifications. I will be using the PyTorch version of BERT provided by ***huggingface*** at https://github.com/huggingface/transformers.

Benchmark Model

A similar type of work has been conducted and published by ***Guoqin Ma*** at Stanford [3], though he used a different dataset altogether for his study. He is using a combination of BERT and several Neural networks (LSTM, CNN, etc) and have obtained different accuracies and other metric scores with different combinations. The paper describes 5 different algorithms with differing evaluation metric scores. I plan to use the algorithms outlined in this paper as benchmark model for my project.

Evaluation Metric

For evaluating my models, I plan to use 4 metrics that are widely used in machine learning classification models, namely - accuracy, precision, recall and F1-score. Mathematical representation for these metrices are as below.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Where TP = True Positive, TN = True Negatives, FP = False Positives and FN = False Negative

Similarly, precision and recall can be calculated as

$$\textbf{Precision} = \frac{\text{True Positive}}{\text{Actual Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive + False Positive}}$$

$$\textbf{Recall} = \frac{\text{True Positive}}{\text{Predicted Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive + False Negative}}$$

F-1 score, on the other hand can be calculated from precision and recall as below:

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Project Design

A very high-level description of the workflow I plan for approaching this NLP binary text classification problem is as below.

- Doing some exploratory data analysis (EDA) with the provided dataset. Identifying any outliers if present and removing them.
- Getting BERT downloaded and set up.
- Converting a dataset in the **.csv** format to the **.tsv** format that BERT better supports.
- Loading the **.tsv** files into a notebook and dividing it into ***train*** and ***test*** sets.
- converting the text representations to a feature representation that the BERT model can work with.
- Setting up a pretrained BERT model for fine-tuning.
- Fine-tuning a BERT model.
- Evaluating the performance of the BERT model.

References

[1] https://www.kaggle.com/c/nlp-getting-started.

[2] https://www.figure-eight.com/data-for-everyone/

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proc. of NAACL, 2018

[4] Guoquin Ma. Tweets Classification with BERT in the Field of Disaster Management. Stanford Univ. https://web.stanford.edu/class/cs224n/reports/custom/15785631.pdf