Machine Learning Engineer Nanodegree

Capstone Proposal - Disaster Tweets

Dimitrios Papadimas
March 30, 2020

Proposal

This project is based on Kaggle's competition **Disaster Tweets** (`https://www.kaggle.com/c/nlp-getting-started`). Its main objective is the use of Natural Language Processing [2] to identify short pieces of text (tweets) that contain information relative to disastrous events.

1 Domain Background

Natural language processing (NLP) is a sub field of computer science and artificial intelligence that refers to the interaction of a computer with one or more human languages. This interaction could consist of language comprehension, reproduction or, simply, analysis. An input for a computer system that uses NLP technology could be text (in its digital form) or spoken language.

The present capstone project will focus on the task of language comprehension from text and more specifically text classification from English text. An important technique that will be utilized is the representation of different words with word embeddings [4].

Word Embeddings is a word representation technique through vectors. These vectors capture syntactic and semantic relationships between words by taking into account the context around which they appear. A very common use of word embedding is to calculate the similarity between two words. The term "banana" for instance, is semantically closer to the term "apple", since they are both fruits and appear in the same context, compared to the term "car".

2 Problem Statement

The problem statement is based on the aforementioned Kaggle's competition description

The famous social media platform Twitter (`https://www.twitter.com`) has become an important communication channel in times of emergency. The ubiquitousness of smartphones enables people to announce an emergency they're observing in real-time. As a result, there is an increase desire from agencies (i.e. disaster relief organizations and news agencies) to programatically monitor Twitter for such events.

However, it is not always clear whether a person's words are actually announcing a disaster or are being used metaphorically. For example, in a tweet such as:

”The sky last night was ablaze!”
the use of the word “ABLAZE” is clearly metaphorically for a ”human reader”. On the other hand, a machine lacking human reasoning or context understanding, cannot make this distinction with equal ease.
Based on the above, this capstone’s challenge will be the development of one or more models that can accept as input a short text (size of a regular tweet) and identify whether its content addresses a real catastrophic event/emergency, or not. For the evaluation of these models, the accuracy metric will be used (See 5). In addition, the accuracy from the Kaggle competition’s testing set will be also considered.

3 Datasets and Inputs

For the development and validation of the above mentioned models, two datasets, provided by the Kaggle’s platform, will be used.
From these, the main dataset is the training set. It contains three fields related to tweets:

- id: the tweet’s unique id in respect to the dataset
- text: the textual content of the tweet
- target: the label defining whether the tweet refers to a real disaster (1) or not (0)

The second dataset, the testing set, will consist only of the first two fields and will be used for predicting the target labels of the tweets and uploading them to Kaggle for validation.
Finally, the final model will accept as input a tweet’s textual content, or a list of tweets along with their contents, and will label it/them as disastrous or not.

4 Solution Statement

The task of developing a text classifier given a large set of labeled data (tweets that have been already classified as disastrous or not from humans) can be addressed by utilizing machine learning algorithms. There are numerous approaches for developing classifiers, but for the scope of this capstone the following are proposed:

- Naive Bayes classifier (as a baseline model) (See 6)
- Dense Neural Network classifier
- LSTM classifier [3]

As far as the input of the classifier is concerned, the textual data must be transformed into a machine readable format; vectors. For this task, the following approaches will be used:

- Representation of a tweet based on its tf-idf of each word

- Representation of each tweet using the Universal Sentence Encoder model [1]
- Representation of each word using word embeddings (See 1)

5 Evaluation Metrics

Since the main goal of this capstone is the development of a binary classifier (whether or not a tweet is related to a disaster), the metric that will be used for the validation of the developed models will be accuracy. More precisely, given a set of labeled tweets, it calculates how many of them are classified correctly by the classifier.

Regarding the benchmark model, it will be trained using a subset of the training set. The remaining dataset will be used for the benchmark calculation. However, in order to calculate the accuracy of the developed models, a k-fold algorithm will be utilized, in order to obtain more precise results. The ratio between the training and validation data will be 4:1.

Last but not least, the performance of the models on the testing set provided by Kaggle will also be considered.

6 Benchmark model

As mentioned above, a Naive Bayes classifier will be used as a baseline model. Based on various analysis (i.e. `https://sebastianraschka.com/Articles/2014_naive_bayes_1.html`) Naive Bayes are often used for text classification. A Naive Bayes classifier, which is based on Bayes' theorem of conditional probability, treats each feature as independent from the others, which can be a naive assumption, but is observed to work out well on text data. Thus, in order to create a baseline score, the training set(See 3) is splitted into two different sets, a training and a validation set. The training set is used to train the model and the validation set to calculate the accuracy metric. For the representation of the tweets, the tf-idf score of each of their words is being used.

The accuracy of the model in the validation set (unseen data) is **68.7 %**. In addition, using the test dataset provided from Kaggle, the accuracy of the baseline model, as reported from Kaggle, is **69.7 %**.

7 Project Design

In this section, a high level summary of the data workflow of the final product will be presented. The developed classifier will get tweets as input (either a single tweet or a list of tweets) and predict whether they refer to a disastrous event or not.

The first step of the workflow is the extraction and pre-processing of the textual data of the tweet. The pre-processing might consist of tokenizing the text, removing noise (such as characters that are not alpha-numerical etc), removing stop-words etc. Next, the textual data are transformed to their vector representations. This step might include tf-idf representation vectors or word

embeddings. Finally, the vectorized textual data are fed to the model and predictions are produced.
The final output which is presented to the user, is a set of predictions for the tweets that he/she enters as input.

References

[1] https://tfhub.dev/google/universal-sentence-encoder/4.

[2] James F. Allen. *Natural Language Processing*, page 1218–1222. John Wiley and Sons Ltd., GBR, 2003.

[3] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[4] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality, 2013.