Capstone Project Proposal

Classification of Disaster Tweets

Bikram Dutta
bikramdutta@outlook.com
(July 15, 2020)

DOMAIN BACKGROUND

In simple words, Natural Language Processing is nothing but a branch of Machine Learning, which teaches machines to recognize, understand and react to the languages that humans use. Since it is associated with human languages, it is termed as Natural Language Processing. It is essentially taking data in natural language as input and processing it in an understandable or processable format for the machine. The ultimate objective of NLP is to read, decipher, understand, and make sense of the human languages in a manner that is valuable.[2] Most NLP techniques rely on machine learning to derive meaning from human languages.

Twitter is an American microblogging and social networking service on which users post and interact with messages known as "tweets". Registered users can post, like, and retweet tweets, but unregistered users can only read them. Users access Twitter through its website interface, through Short Message Service (SMS) or its mobile apps.[1] In today's world Internet is a luxury no more. Twitter has high visibility to all bodies, big and small. There are no barriers or filters in the flow of information. It is all transparent. The 'direct mention' feature has revolutionized the phrase – 'dropping the curtains'. In other words, it is all out in the open and everybody is watching.

Access to this kind of real time data has many collateral benefits. These benefits depend on the nature of the information. Processing this information using machine learning and having automated systems to react to that information is technically the way forward for all of us. One such extremely useful application is getting the twitter real time information related to disasters, processing it and enabling the particular task force to aid the situation.

Every activity in disaster management such as managing evacuation plan, and running rescue missions demands accurate and up-to-date information to allow a quick, easy, and cost-effective response to reduce the possible loss of lives and properties. It is a challenging and complex task to acquire information from different regions of a disaster-affected area in a timely fashion. The extensive spread and reach of social media and networks allow people to share information in real-time. However, the processing of social media data and gathering of valuable information require a series of operations such as processing each specific tweet for a text classification, possible location determination of people needing help based on tweets, and priority calculations of rescue tasks based on the classification of tweets. These are three primary challenges in developing an effective rescue scheduling operation using social media data. [4]

Diving into technicality, these tweets are nothing but data and information in human languages i.e. natural language data. Thus, our primary focus is processing this natural data, using NLP, and deriving inferences from it.

PROBLEM STATEMENT

In the times of emergencies and disasters, people and groups tweet the details of disaster, their current positions, an SOS signal, a danger signal, and other valuable information. These tweets are nothing but words and sentences i.e. natural language data.

The ultimate goal is to get the tweets and identify whether the tweets are truly related to a disaster or not. A set of tweets is provided, for which the accurate classification is already done. It is required to build a machine learning model that predicts which Tweets are about real disasters and which ones aren't.[5] The problem statement is part of a Kaggle Competition.[5]

DATASETS AND INPUTS

This problem statement is part of a challenge in a Kaggle Competition. The link to the Kaggle Competition- https://www.kaggle.com/c/nlp-getting-started/

A dataset of 10,000 tweets are provided. These 10,000 tweets are divided into 2 sets of training and test data.

Following files are provided in the Kaggle Dataset: -

Sr. No.	Data	Description
1.	train.csv	A csv file containing tweet texts and their classification.
2.	test.csv	A csv file containing tweet texts, for which classification has to be done.

SOLUTION STATEMENT

It is a straightforward binary classification, whether a tweet is truly a disaster tweet or not.

For this particular problem, the initial steps of cleaning and segregating the data are of utmost importance. If I don't clean the data and extract relevant textual data then our trained model can turn out to be erroneous and irrelevant.

After that, I will explore the data to summarize the main characteristics of the data at hand. Based on the summary, I will decide whether or not any other treatment is to be done on the data.

Since classification is to be done, I will have to observe the accuracy against a few classifiers viz. Logistic Regression Classifier, KNN Classifier, Naïve Bayes Classifier, Decision Tree

Classifier, Random Forest Classifier, SVM Classifier, etc. and select the best model after comparison. In addition to the basic classification techniques, I intend to use the benchmark model that I have stated in the following section. I also intend to use the Keras BERT (Bidirectional Encoder Representations from Transformers) model [3] because this approach is also conceptually rich and fascinating. After that, I will have to choose the best classification model according to accuracy and speed.

Finally, I will make use of the selected model to classify the tweets in the test dataset and submit the final predictions.

BENCHMARK MODEL

Since this is a binary classification post NLP, the initial steps tend to be conventional with custom modifications. The classification later on depends on the classifier and the way the classifier is tuned.

There are many approaches for this set of problem, but one model has used conceptually rich and basic approach for the requirement: -

TFIDF based Feature Words Extraction and Topic Modelling combined with a basic Logistic Regression Classifier [6]

After pre-process of raw text, distinct terms extracted compose the Bag of Words. And then TFIDF is used in order to re-weight the count features for reducing the influence of more frequent yet less valuable terms and enhancing the influence of rarer yet more valuable terms. Topic modelling based on Term Frequency times In-verse Document Frequency (TFIDF) and Latent Dirichlet Allocation (LDA) is achieved.[7]

sklearn.feature_extraction.text.TfidfVectorizer[8] is used to transform the texts. This is inputted to a basic Logistic Regression model and the model is trained. Cross validation is performed and the model and parameters with best accuracy is chosen.

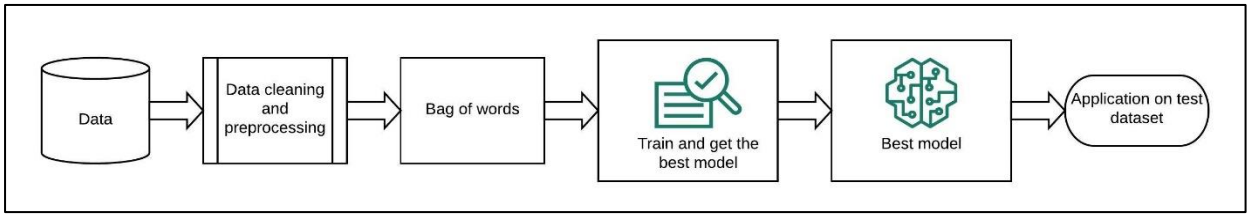
EVALUTATION METRICS

4 metrics are involved: -

1. **Accuracy** measures the fraction of the total sample that is correctly identified
2. **Precision** measures that out of all the examples predicted as positive, how many are actually positive
3. **Recall** measures that out of all the actual positives, how many examples were correctly classified as positive by the model
4. **F1 Score** is the harmonic mean of the **Precision** and **Recall**

Classification Report of sklearn.metrics.classification_report, computes those metrics for the given training and validation set.

PROJECT DESIGN



The solution would require me to indulge in the following operations: -

1. **Obtain data**
   Gathering the required data from the appropriate data source. The details of the dataset are mentioned in the DATASET AND INPUTS section.

2. **Pre-processing and data cleaning**
   Natural language data can contain many unwanted characters and irrelevant data. Therefore, data cleaning and pre-processing is required.

3. **Exploratory Data Analysis**
   It gives a summary of the nature of the data. Based on this summary I will decide whether or not any further treatment is to be done on the data.

4. **Getting a bag of words**
   A bag-of-words is a representation of text that describes the occurrence of words within a document.[11] It involves two things:
   a. A vocabulary of known words.
   b. A measure of the presence of known words.

5. **Use and test different models and approaches**
   a. Train the classification model according to the approach
   b. Cross Validation
   c. Verify and modify the model according to the accuracy
   d. Get the best of the model

6. **Select the best approach and model**
   I intend to select the approach and model with the highest accuracy.

7. **Apply the selected model on test set.**

8. **Submit the final classification report.**

REFERENCES

1. Wikipedia contributors. (2020, July 14). Natural language processing. In *Wikipedia, The Free Encyclopaedia*. Retrieved 03:02, July 15, 2020, from https://en.wikipedia.org/w/index.php?title=Natural_language_processing&oldid=967567487
2. 'A simple introduction to Natural Language Processing,' *Medium* by Dr. Michael J. Garbade. (2018, October 15) n.p. https://becominghuman.ai/a-simple-introduction-to-natural-language-processing
3. 'Bert – Explained state of art language model for NLP', *Medium* by Rani Horev (2018, November 10) n.p. https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270
4. 'A Deep Learning Approach for Tweet Classification and Rescue Scheduling for Effective Disaster Management (Industrial)' by Md. Yasin Kabir and Sanjay Madria, SIGSPATIAL , November 2019, Chicago, Illinois, USA https://arxiv.org/pdf/1908.01456.pdf
5. Kaggle Competition: Real or Not? https://www.kaggle.com/c/nlp-getting-started/overview/description
6. 'Disaster Tweets EDA, Basic Model,' *Kaggle Kernel* by Ratan Rohith https://www.kaggle.com/ratan123/start-from-here-disaster-tweets-eda-basic-model
7. Zhao, Guifen & Liu, Yanjun & Zhang, Wei & Wang, Yiou. (2018). TFIDF based Feature Words Extraction and Topic Modeling for Short Text. 188-191. 10.1145/3180374.3181354. https://www.researchgate.net/publication/324465038_TFIDF_based_Feature_Words_Extraction_and_Topic_Modeling_for_Short_Text
8. Library description: - https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
9. Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.
10. API design for machine learning software: experiences from the scikit-learn project, Buitinck *et al.*, 2013.
11. 'A Gentle Introduction to the Bag-of-Words Model,' *Machine Learning Mastery* by Jason Brownlee. (2017, October 9) from https://machinelearningmastery.com/gentle-introduction-bag-words-model/