MINI PROJECT REPORT

Disaster Control through Big Data Analysis of Tweets

Submitted to the

Pune Institute of Computer Technology, Pune.

In partial fulfilment for the award of the Degree of

Bachelor of Engineering

In

Information Technology

By

Raghav Utpat

Sanya Varghese

Saniya Shah

Neelanjney Pilarisetty

Under the guidance of

Prof. D. D. Londhe

Department Of Information Technology

Pune Institute of Computer Technology College of Engineering

Sr. No 27, Pune-Satara Road, Dhankawadi, Pune -

CERTIFICATE

This is to certify that the project report entitled

Disaster Control through Big Data Analysis of Tweets

Submitted by

Raghav Utpat

Sanya Varghese

Saniya Shah

Neelanjney Pilarisetty

is a bonafide work carried out by them under the supervision of Prof. D. D. Londhe and it is approved for the partial fulfilment of the requirement of Software Laboratory Course-2015 for the award of the Degree of Bachelor of Engineering (Information Technology)

Prof. D. D. Londhe

Internal Guide

Dept. Information Technology

Prof. A. M. Bagade

External Guide

Dept. Information Technology

Prof. D. D. Londhe

Internal Guide

Date:

Place:

ACKNOWLEDGEMENT

We thank everyone who has helped and provided valuable suggestions for successfully creating a wonderful project.

We are very grateful to our guide, Prof. D. D. Londhe, Head of Department Dr A. M. Bagade and our principal Dr P. T. Kulkarni. They have been very supportive and have ensured that all facilities remained available for the smooth progress of the project.

We would like to thank our professor and Prof. D. D. Londhe for providing very valuable and timely suggestions and help. We would also like the entire project staff team for providing valuable reviews and suggestions from time to time.

We would like to thank our entire department and college staff for the very valuable help and co-ordination throughout the duration of the project.

We would also like to thank our families and all our friends for the valuable support they provided throughout the duration of the project.

Raghav Utpat

Sanya Varghese

Saniya Shah

Neelanjney Pilarisetty

ABSTRACT

Twitter 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. Because of this, more agencies are interested in programmatically monitoring Twitter (i.e. disaster relief organizations and news agencies).

But, it's not always clear whether a person's words are actually announcing a disaster. There might be the usage of words like, "ABLAZE". However, the author may mean it metaphorically. This is clear to a human right away, especially with the visual aid. But it's less clear to a machine.

We've built a machine learning model that predicts which Tweets are about real disasters and which ones aren't. We have access to a dataset of 10,000 tweets that were hand classified.

During times of crisis and emergencies, Twitter's live, open and public features have been leveraged by NGOs, citizens, government agencies and the media to share and exchange information. This has to lead to unprecedented collaboration by NGOs, citizens, government agencies and the media on the platform.

In South Asia, the usefulness of Twitter during disaster relief came to light during the Kashmir floods of 2014, as citizen groups organized themselves entirely on Twitter to help agencies on the ground. This effort was replicated in 2015 when Chennai was hit with a flood that lasted several weeks.

In July 2016, @TwitterIndia worked with NGOs, other private sector and citizen participants to work towards a focused strategy for disaster relief operations. Our next steps have included onboarding relief agencies on Twitter to amplify their message, plan capacity-building workshops for NGOs, and increase implementation of best practices.

Index

LIST OF FIGURES

ABBREVIATIONS

1	EDA	-	Exploratory Data Analysis
2	BERT	-	Bidirectional Encoder Representations from Transformers
3	CLS	-	Classification token
4	SEP	-	Separator token
5	MLM	-	Masked Language Modeling
6	NSP	-	Next Sentence Prediction
7	GLUE	-	General Language Understanding Evaluation
8	MNLI	-	Multi-Genre Natural Language Inference
9	QNLI	-	Question Natural Language Inference
10	SQuAD	-	Stanford Question Answering Dataset
11	MRPC	-	MicRosoft Paraphrase Corpus

INTRODUCTION

How TWITTER helps in a Disaster?

The evidence continues to pile up that Twitter is a news service, not a social network. Of course, Twitter only works like a news service because its news is routed according to social connections–and that's the secret to the service's ability to endlessly issue, digest and re-synthesize news into actionable 140-character memoranda. This is true even–or perhaps especially–in an emergency.

It is observed that many of the tweets during a disaster were issued by local and national news media, but a surprising number originated with disaster-specific Twitter accounts that arose for the purpose of updating others with useful information. Local and national news organizations, especially, engaged in synthetic and derivative tweeting during the disaster, while 80% of the original, "citizen-reported" tweets came from locals who were living the disaster.

This complicated interplay between original reporting by locals and synthesis by both traditional news media outlets and flood-specific twitterers led researchers to conclude that Twitter is not simply a platform for broadcasting information, but one of informational interaction. Navigation of this unwieldy space is difficult. Many of these conventions have evolved to aid this navigation, directing other users to valuable information, placing virtual signposts within a complex information space."

"Through retweeting, Twitterers both self-organize and create the need for more self-organization, as they generate even more noise that gives rise to the need for more directing and focusing behaviours. Derivative information production is, therefore, a user-driven cycle of shaping and re-shaping a shared interaction and information space."

A natural question in disaster tweeting is whether or not the information pouring out of Twitter on a particular event can be processed quickly enough - either by Twitter users themselves or some outside body - to allow decision-makers to act.

MOTIVATION

With the advent of new technologies, our ability to communicate with one another has evolved significantly. No longer are societies solely dependent on traditional media outlets, newspapers, radio, and TV for the news. With rapidly evolving smartphone technologies, societies are just an 'app' away from being able to deliver or receive information within milliseconds. Popular social media platforms such as Twitter, Facebook, and YouTube, have supplanted the traditional media outlets for accessing and responding to information. Every day millions of users worldwide are connected and receive their news via online social networks warranting researchers to study the mechanisms behind human interactions.[1]

What has not been explored adequately is whether these new platforms can be recognized as vital resources to improve response to the disaster and crisis events by targeting a specific geographic location that is susceptible to a disaster. The inability to reach geographically targeted populations remain some of the main reason for inadequacies in disaster response, especially where critical information needs to reach and be disseminated rapidly to the most 'at-risk and vulnerable populations'

APPLICATION

Twitter has shown to have the potential to increase survival during Tornado-related disasters. Social media's technology platforms allow for multidirectional network communication which can aid officials during disasters. This provides public and mental health value to the population affected by connecting vital services and resources.

This project would help Disaster managers to discover ways to mitigate morbidity and mortality throughout the entire disaster cycle: prevention, preparedness, response and recovery. This capacity is largely dependent on community empowerment and community level mobilization and health promotion efforts. People who are at the bottom of the pyramid of preparedness for disasters are empowered and are given the opportunity to be "made partners in the process of preparedness with usable knowledge, social awakening, practical training and guidance for use of local resources indicatively to respond with developed skills". Twitter has become an immensely valuable tool worldwide, particularly in disseminating and conversing about issues in everyday life.

Despite a limited 140 character maximum and the consequential pithy form of text used, it remains an extremely popular means of communication. Hence, our project would help read tweets and understand the severity of a disaster that might occur.

LITERATURE SURVEY

INTRODUCTION

The paper: “Tweets Classification with BERT in the Field of Disaster Management” by Guoqin Ma,Department of Civil Engineering Stanford University talks about how The rise of social media over the past 15 years has marked a shift in the potential of how information is collected and disseminated during natural disasters. Researchers have leveraged social media to fulfill many tasks in disaster management, including but not limited to outbreak detection, information retrieval, sentiment analysis, evacuation behavior study, hazard assessment, and damage assessment.

The labelling system used in this paper, is a classification criterion called informativeness-based. This labeling could offer us more insights into the local situation when a disaster occurs.

APPROACH

In this paper “Tweets Classification with BERT in the Field of Disaster Management”, BERT models are built based on the pytorch-pretrained-BERT repository on github. All the BERT models are builted upon BERT base uncased model.

Here, for text preprocessing, texts are lowercased. Non-ascii letters, urls, @RT:[NAME], @[NAME] are removed Texts with length less than 4 are thrown away. No lemmatization is performed and no punctuation mark is removed since pre-trained embeddings are always used. No stop-word is removed for fluency purpose.

DATA

The classification criterion is informativeness-based. The datasets used are CrisisLexT26 and some datasets on CrisisNLP. Minor difference in labeling may exist between these datasets, but generally the labels are: affected individuals, infrastructure and utilities damage, caution and advice, donation and volunteering, sympathy and emotional support, other useful information, not related or not informative, etc. These labels could offer us insights into the local situation when a disaster occurs. In this paper, the aforementioned labeled datasets are compiled into a single large one. The number of labeled Tweets was 75800. Most Tweets are in English, with some sparse exceptions

Label Name	Label Count
Not Realted/Not Informative	25785
Other useful Information	18877
Donations/Voluntering	8925
Affected Indivisuals	8009
Sympathy/Emotional Support	5020
Infrastucture/Utilities Damage	4559

Caution/Advise	3171

Disaster Type	Disaster Count
Hurricane	30860
Earthquake	20540
Floods	20540
Wildfire	3620
Landslides	2598
Traffic Crash	2385
Terrorism	1977
Building Collapse	945
Meteor	915
Explosion	907
Haze	706

Volcano	211

Experimentation

Here, 5 metrics, namely accuracy, Matthews correlation coefficient, precision, recall, F1-score, are considered when evaluating a model. Accuracy, Matthews correlation coefficient, macro precision, macro recall, macro F-1 score are calculated from all the classes, while precision, recall, and F1-score, are calculated for each class. Adam optimizer is used for all models' training. Validation dataset and test dataset are of size 5000 respectively. The rest 64,346 samples are used for training. Samples are shuffled between epochs during training.

Table 4: Evaluation metrics

Model	Accuracy	Mathews Coef.	Macro Precision	Macro Recall	Macro F-1
Baseline	0.64	0.56	58.00	68.43	60.71
Default BERT	0.67	0.59	60.43	71.14	64.00
BERT+NL	0.67	0.59	60.43	71.14	64.00
BERT+LSTM	0.67	0.59	60.57	68.00	63.14
BERT+CNN	0.67	0.59	60.86	69.29	63.43

RESULTS

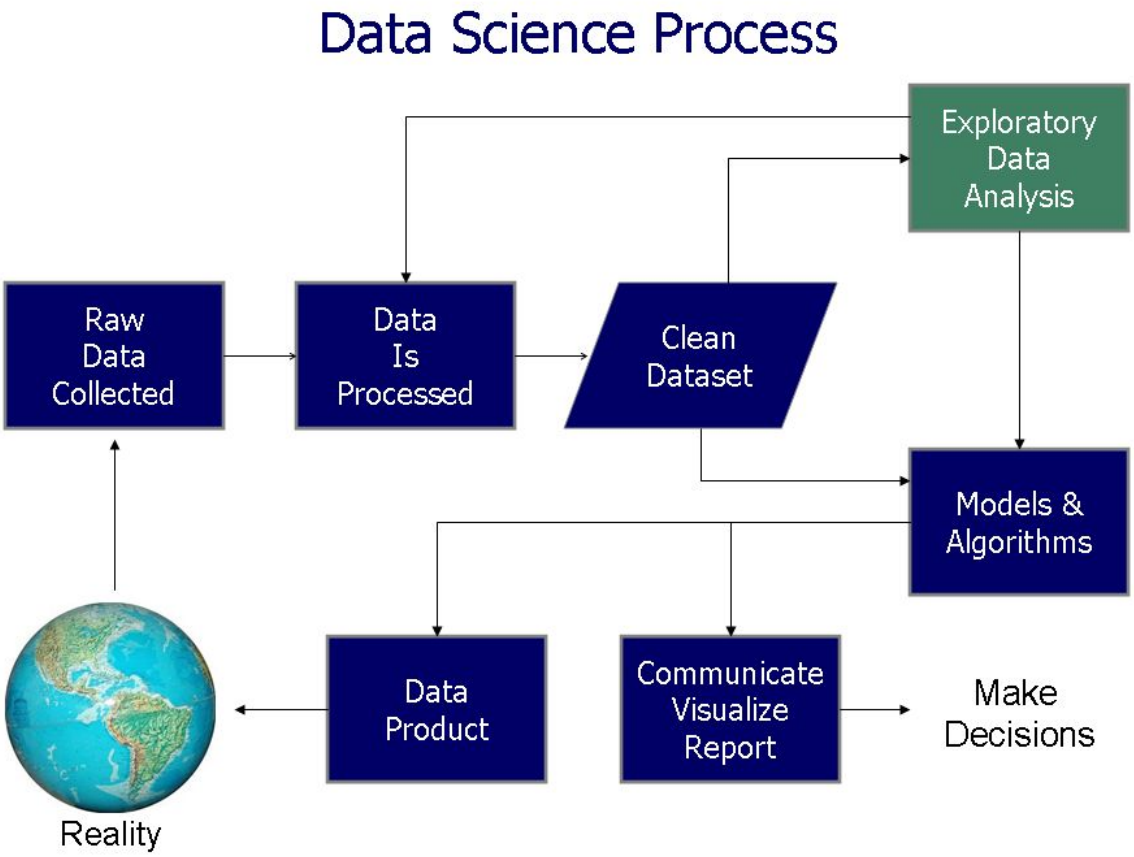
The accuracy score and Matthews of all BERT models are higher than the baseline for around 3%. Averagely, default BERT, BERT+NL, BERT+LSTM, BERT+CNN outperform baseline model by 2.4(%), 2.6, 3.3, 2.9 in terms of precision; The 3 customized BERT models have better performance in precision but worse performance in recall than the default BERT.

FINDINGS

BERT-based classifiers could attain better performance compared with the bi-LSTM baseline model. Some labels are better predictable than others. Accurate message classification is a necessary requirement to make decisions from the abundant but noisy user-generated data. Ambiguity and subjectivity are a great obstacle to boost up performance of classifier. The quality of the data needs improvement so as to be able to better classify tweets.

Design and Implementation

The Data Science Process



Wikimedia : Data Visualisation Process

Phases of Implementation

1. Exploratory Data Analysis :

Wikipedia defines **exploratory data analysis** (**EDA**) as “an approach to analyzing data sets to summarize their main characteristics, often with visual methods”. It is used to see what the data can tell us without any formal modeling. It focuses narrowly on checking assumptions required for model fitting and hypothesis testing .

2. Data Cleaning :

As is said often in Computing in general and Data science in particular :

Garbage in , garbage out

Incorrect or inconsistent data leads to false conclusions . A lot of experts even go as far as to say that ‘quality data beats fancy algorithms’ . A simple algorithm can outweigh a complex one just because it was given enough high quality data.

3. Model selection :

It is the phase of choosing the right kind of models which can best fit the data at hand and also any novel data that it hasn’t seen before. This requires statistical and mathematical expertise. Choosing the right kind of the model ensures maximum efficiency and performance. It ensures that the right meaning is extracted from the data.

4. Model training :

In this phase we fit the parameters of our chosen model to the data we have by incrementally improving its performance according to a particular metric using optimization models such as gradient descent . Data is often divided into batches and input to the model .

5. Model Testing :

It is the phase where the model trained in the training phase is checked for performance using a set of data with known values of dependent variables/target

variables . This is used to ensure that the model can generalize to data outside the training set i.e. it has a low variance.

6. Result Analysis :

The performance of the model on both the training and test data is analysed . Often through visualisations , the performance of the algorithm is examined for different cases and different independent variables . The results are compared to the baseline model to get an idea of relative superiority.

Algorithms and Other Terms

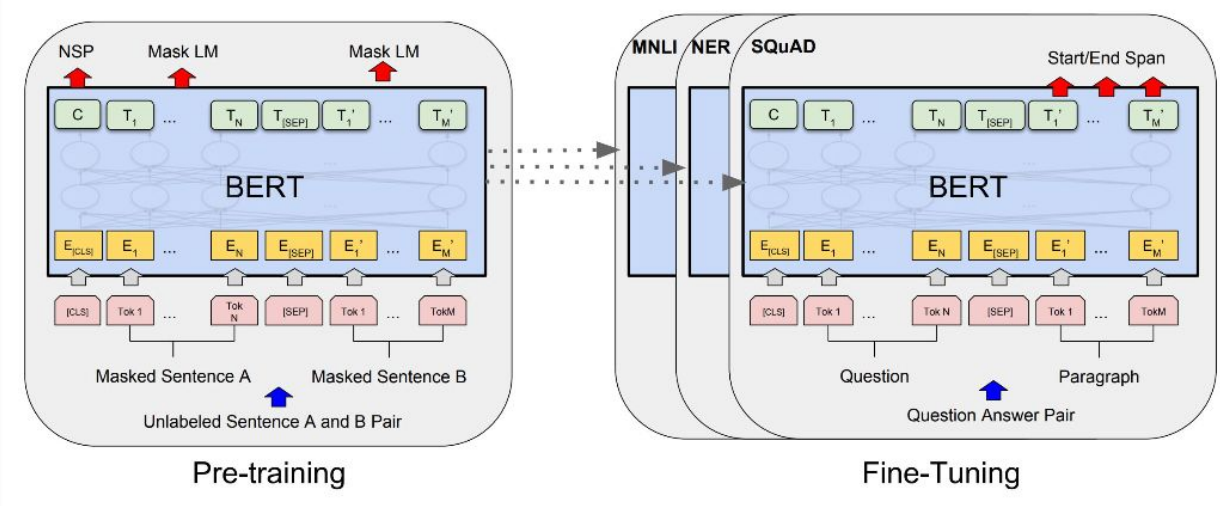
1. BERT

Bidirectional **E**ncoder **R**epresentations from **T**ransformers , is a bi-directional, unsupervised language representation, trained using a plain text corpus.

Pre-trained representations can either be context-free or contextual, and contextual representations can further be unidirectional or bidirectional . Context-free models generate a single word embedding for every input word . Contextual models instead generate a representation for each different occurrence i.e. each context of every word.

As opposed to directional models, which read the text input sequentially (left-to-right or right-to-left), the Transformer encoder reads the entire sequence of words at once. Therefore it is considered bidirectional. This characteristic allows the model to learn the context of a word based on all of its surroundings (left and right of the word).

There are two steps in our framework : pre-training and fine-tuning



Credit : Google AI

Pre Training BERT -

The model is pre-trained using the following 2 unsupervised tasks on the BooksCorpus and Wikipedia corpus -

- Masked Language Modeling(MLM) : A critical issue in bi-directional models is that it would allow a word to , in a sense, “see itself” and models can trivially predict the word. A solution to this problem is Masked Language Modeling . We simply mask some percentage of the input tokens at random, and then predict those masked tokens.
- Next Sentence Prediction(NSP) : In order to train a model that understands sentence relationships, we pre-train for a binarized next sentence prediction task that can be trivially generated from any monolingual corpus. Specifically,when choosing the sentences A and B for each pre-training example, 50% of the timeBis the actual next sentence that follows A(labeled as IsNext) , and 50% of the time it is a random sentence from the corpus (labeled asNotNext).

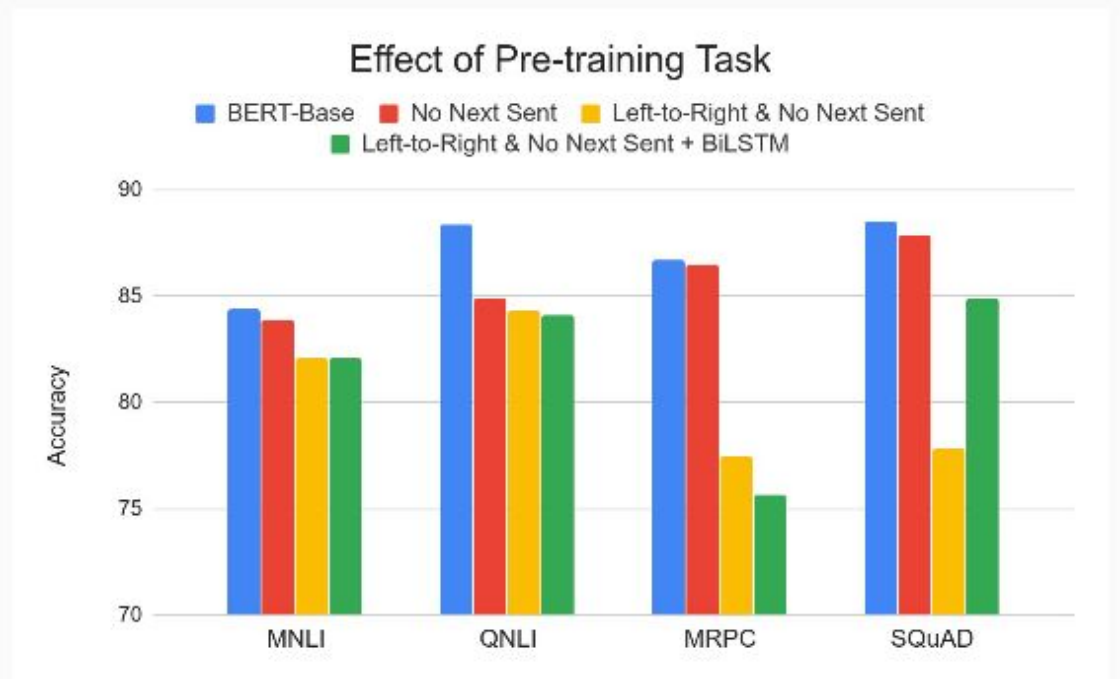
For Wikipedia we extract only the text passages and ignore lists, tables, and headers. It is criti-cal to use a document-level corpus rather than a shuffled sentence-level corpus such as the BillionWord Benchmark in order to extract long contiguous sequences.

Fine Tuning BERT -

For each task, we simply plug in the task-specific inputs and outputs into BERT and fine-tune all the parameters end-to-end.

For fine-tuning, the BERT model is first initialized with the pre-trained parameters, and all of the param-eters are fine-tuned using labeled data from the downstream tasks. Each downstream task has sep-arate fine-tuned models, even though they are ini-tialized with the same pre-trained parameters.

A distinctive feature of BERT is its unified ar-chitecture across different tasks. There is minimal difference between the pre-trained architec-ture and the final downstream architecture.



Credit : Google AI

A graph showing the accuracy of models on different tasks from the GLUE benchmark. All tasks are single sentence or sentence pair classification. The models are different combinations of i) is or isn't bidirectional ii) has next sentence prediction(NSP)

2. Regularization

According to Wikipedia :

"In mathematics, statistics, and computer science, particularly in machine learning and inverse problems, regularization is the process of adding information in order to solve an ill-posed problem or to prevent overfitting."

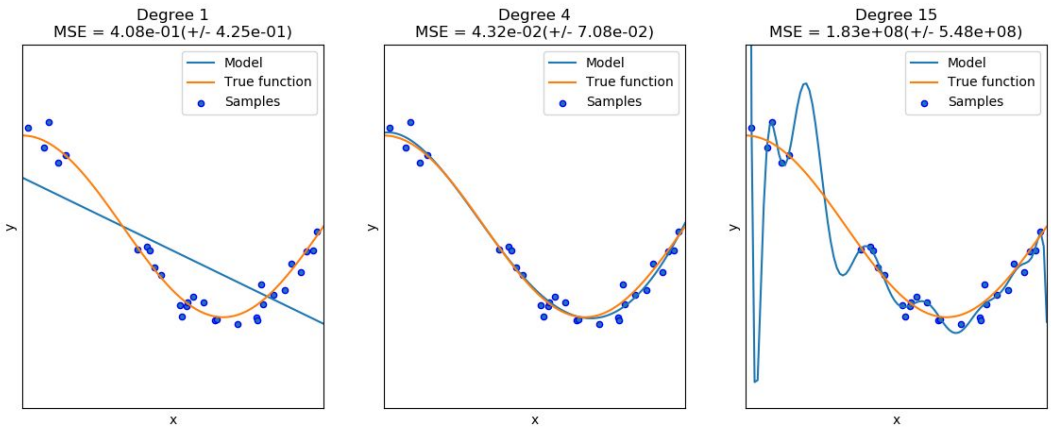
The concepts of bias,variance and the balance between them are critical to understand the relevance of regularization :

The bias-variance tradeoff :- The bias–variance tradeoff is the property of a set of predictive models whereby models with a lower bias in parameter estimation have a higher variance of the parameter estimates across samples, and vice versa. The bias–variance dilemma or bias–variance problem is the conflict in trying to

simultaneously minimize these two sources of error.

According to Wikipedia :

- The bias error is an error from erroneous assumptions in the learning algorithm. High bias can cause an algorithm to miss the relevant relations between features and target outputs (underfitting).
- The variance is an error from sensitivity to small fluctuations in the training set. High variance can cause an algorithm to model the random noise in the training data, rather than the intended outputs (overfitting).



Credit : scikit-learn.org

Image showing a polynomial model being fit to the given data. From the left

a) Degree 1 polynomial has high bias low variance (underfitting)

b) Degree 4 polynomial almost perfectly fits the true function

c) Degree 15 polynomial has very low bias and high variance (overfitting)

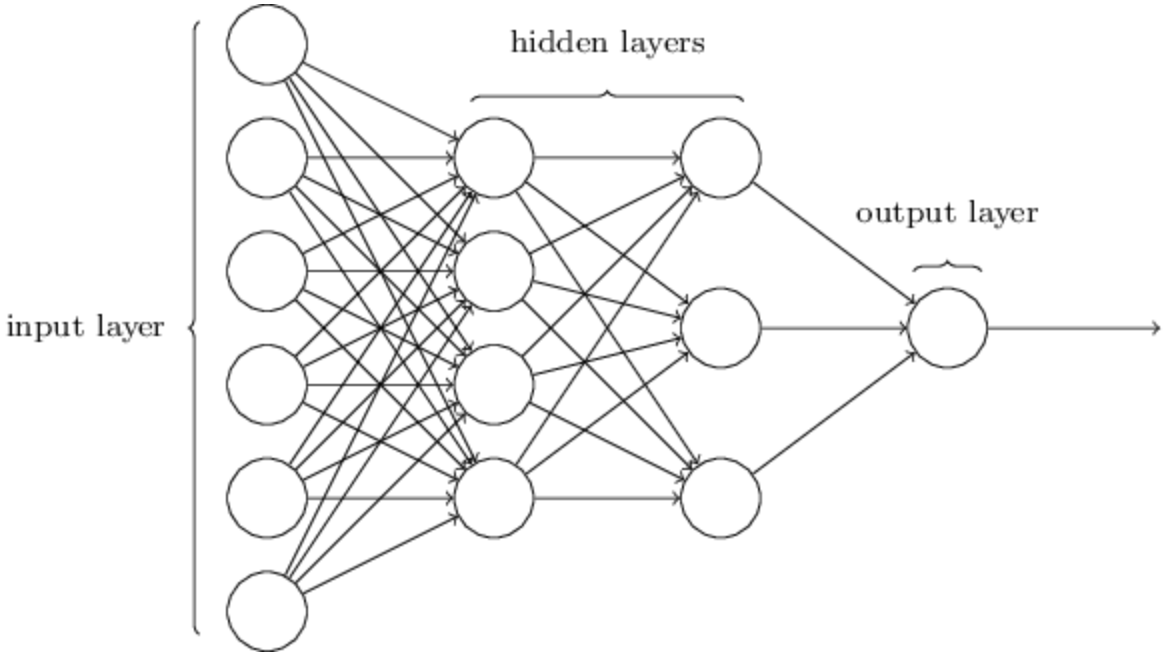
Regularization is a technique used for tuning the function by adding an additional penalty term in the error function. The additional term controls the excessively fluctuating function such that the coefficients don't take extreme values.

3. Neural Network

Modeled loosely on the human brain, a neural net consists of thousands or even millions of simple processing nodes that are densely interconnected. Most of

today's neural nets are organized into layers of nodes, and they're "feed-forward," meaning that data moves through them in only one direction. An individual node might be connected to several nodes in the layer beneath it, from which it receives data, and several nodes in the layer above it, to which it sends data.

To each of its incoming connections, a node will assign a number known as a "weight." When the network is active, the node receives a different data item — a different number — over each of its connections and multiplies it by the associated weight. It then adds the resulting products together, yielding a single number. If that number is below a threshold value, the node passes no data to the next layer. If the number exceeds the threshold value, the node "fires," which in today's neural nets generally means sending the number — the sum of the weighted inputs — along all its outgoing connections.



When a neural net is being trained, all of its weights and thresholds are initially set to random values. Training data is fed to the bottom layer — the input layer — and it passes through the succeeding layers, getting multiplied and added together in complex ways, until it finally arrives, radically transformed, at the output layer. During training, the weights and thresholds are continually adjusted until training data with the same labels consistently yield similar outputs.

Tools Used

1. Anaconda
2. Jupyter Notebook
3. Collab
4. Python 3.6
5. Tensorflow
6. Keras
7. Pandas
8. Numpy
9. Matplotlib
10. Scikit learn

DATASET DESCRIPTION

The dataset used for this project was made openly available by Figure Eight, a machine intelligence company, on their website here - https://www.figure-eight.com/data-for-everyone/

The data is available as a manually labeled collection of ~10,000 tweets that have keywords linked to disasters.

This data is divided into train and test sets in a 70:30 ratio after randomly shuffling it.

The data description :-

1. ID – unique token for each tweet
2. Keyword – The word in the tweet that signals a disaster
3. Location – location of user if available
4. Text – the contents of the tweet
5. Target – whether the tweet actually is about a disaster or not

Visualization

Visualizing data plays a vital role in this project. Following are the representations used:-

1) A simple bar graph to compare the number of real and fake tweets
2) A word cloud is shown to depict the most frequently used words
3) Confusion matrices are used to give the final result processed by both the models individually.
4) A bar plot is used to represent the most frequently used words.

References

- Tweets Classification with BERT in the Field of Disaster Management
- https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/poster/15785631.pdf
- TensorFlow Hub
- concrete_NLP_tutorial/NLP_notebook.ipynb at master · hundredblocks/concrete_NLP_tutorial
- Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”, 2018

Future Works

We aim to analyse more datasets and wish to be able to increase the accuracy along with the speed of detection of the disaster because we believe that time plays a very crucial role in a system like ours where lives may be lost.

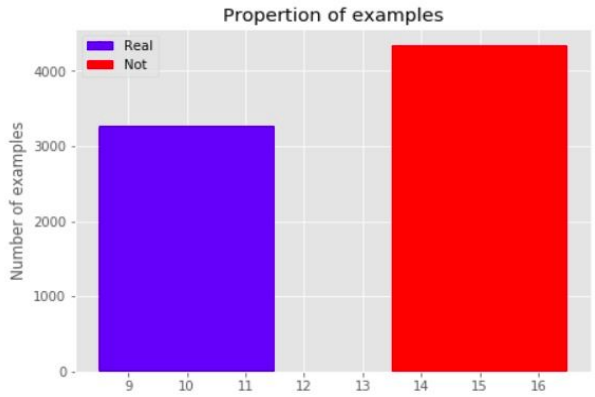
A bi-directional LSTM can be stacked on top of a customized BERT layer for improved results . A CNN in place of the LSTM might further increase performance.

Appendix

Screenshots from the project

1) A Bar plot to show number real and fake tweets

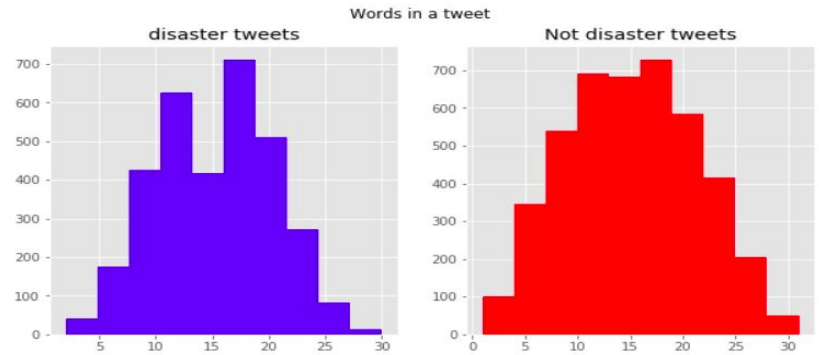
```
# bar plot of the 3 classes
plt.rcParams['figure.figsize'] = (7, 5)
plt.bar(10,Real_len,3, label="Real", color='blue')
plt.bar(15,Not_len,3, label="Not", color='red')
plt.legend()
plt.ylabel('Number of examples')
plt.title('Propertion of examples')
plt.show()
```



2) Comparison graphs for words in tweets

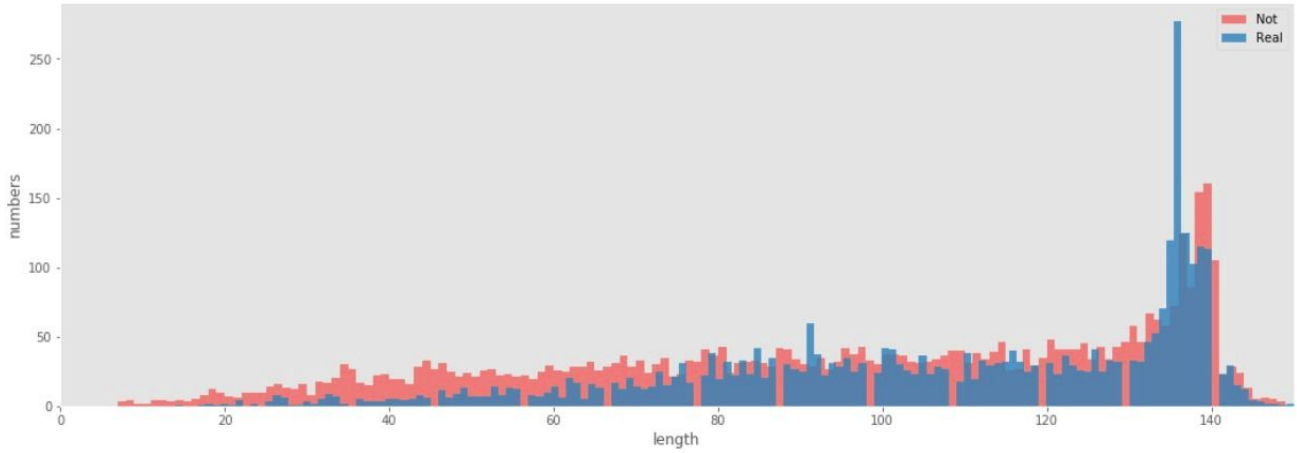
Number of words in a tweet

```
fig,(ax1,ax2)=plt.subplots(1,2,figsize=(10,5))
tweet_len=tweet[tweet['target']==1]['text'].str.split().map(lambda x: len(x))
ax1.hist(tweet_len,color='blue')
ax1.set_title('disaster tweets')
tweet_len=tweet[tweet['target']==0]['text'].str.split().map(lambda x: len(x))
ax2.hist(tweet_len,color='red')
ax2.set_title('Not disaster tweets')
fig.suptitle('Words in a tweet')
plt.show()
```



3) A histogram of number of characters in tweets

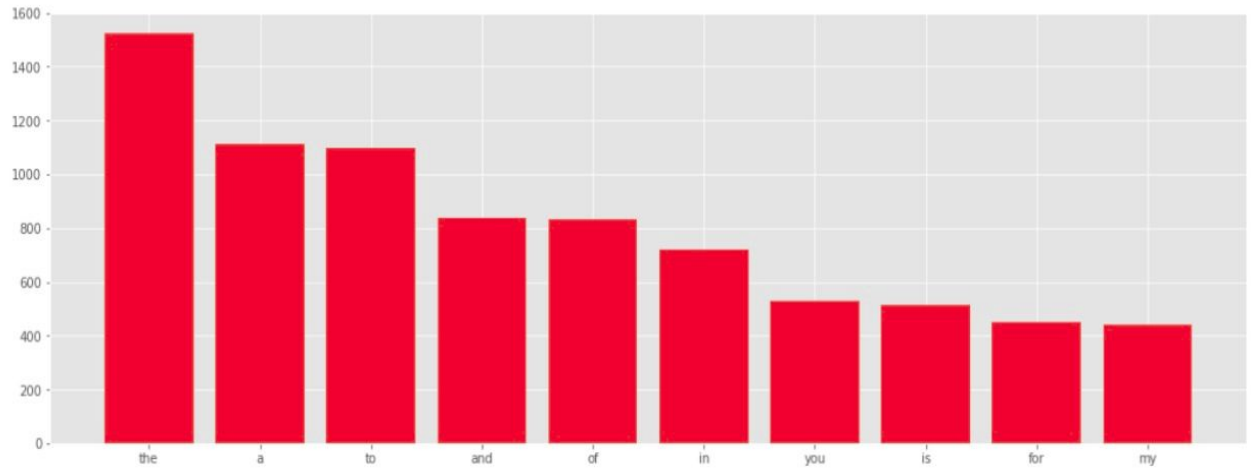
```
plt.rcParams['figure.figsize'] = (18.0, 6.0)
bins = 150
plt.hist(tweet[tweet['target'] == 0]['length'], alpha = 0.6, bins=bins, label='Not')
plt.hist(tweet[tweet['target'] == 1]['length'], alpha = 0.8, bins=bins, label='Real')
plt.xlabel('length')
plt.ylabel('numbers')
plt.legend(loc='upper right')
plt.xlim(0,150)
plt.grid()
plt.show()
```



4) A continuous Bar Container of the most common stopwords

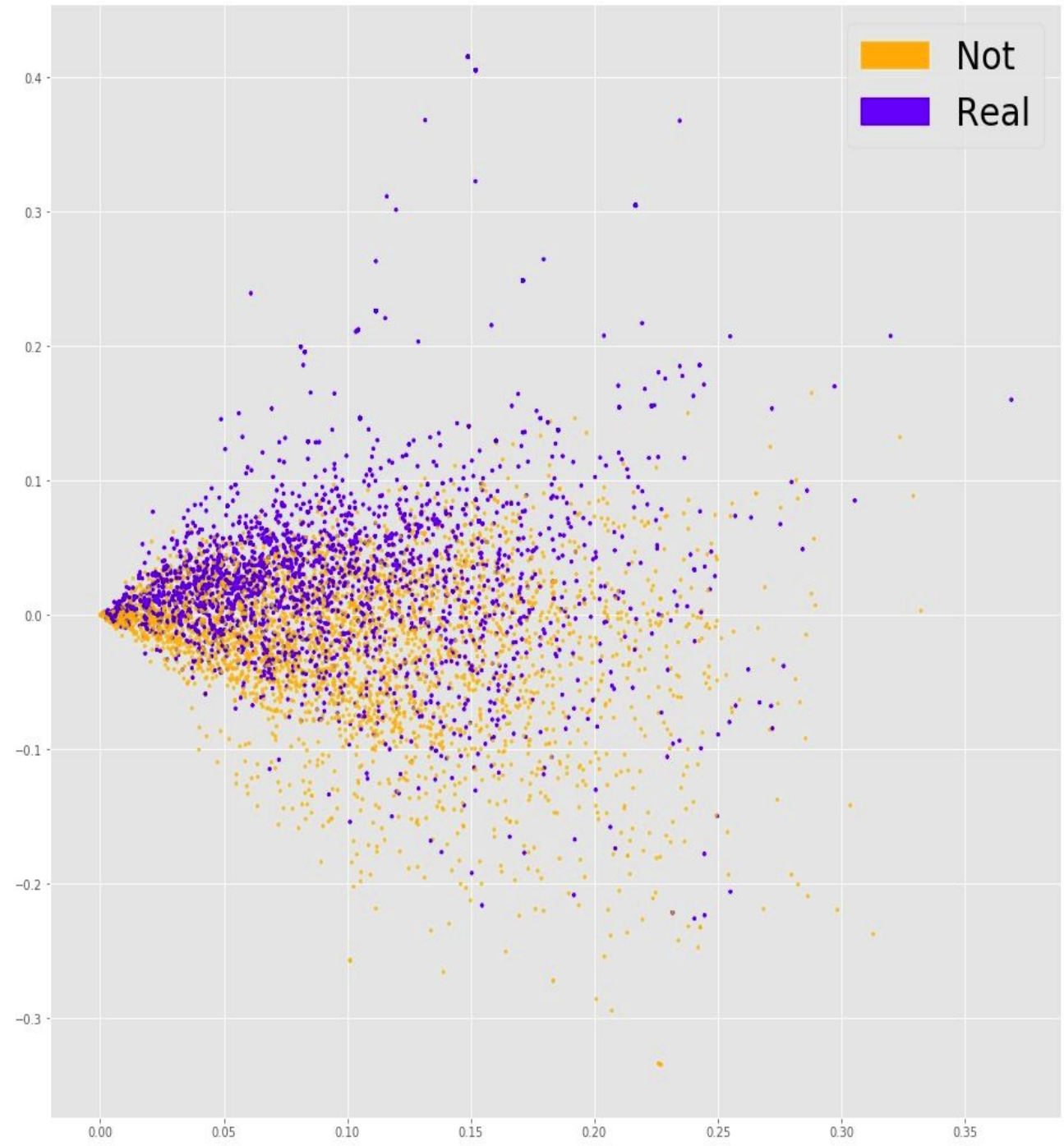
```
plt.rcParams['figure.figsize'] = (18.0, 6.0)
x,y=zip(*top)
plt.bar(x,y)
```

```
<BarContainer object of 10 artists>
```



5) A Scatterplot for topic modelling

```
fig = plt.figure(figsize=(16, 16))
plot_LSA(X_train_tfidf, y_train)
plt.show()
```



6) Wordcloud to display the set of words in a real tweet

```
# Generating the wordcloud with the values under the category dataframe
plt.figure(figsize=(12,8))
word_cloud = WordCloud(
                          background_color='black',
                          max_font_size = 80
                         ).generate(" ".join(corpus_new1[:50]))
plt.imshow(word_cloud)
plt.axis('off')
plt.show()
```