Social Media Content Analyst Tool: An NLP-based System to Identify Informative Tweets

1st Hardik Sonetta
School Of Computer Science
University of Windsor
Windsor, Canada
sonetta@uwindsor.ca

2nd Jalpa Patel
School Of Computer Science
University of Windsor
Windsor, Canada
patel2fu@uwindsor.ca

***Abstract*—Information is cascaded on social media platforms and it has been proven to be the best medium for raising situational awareness for various causes and events. It helps us to disseminate information regarding current incidents including natural disasters. The information diffused on social networks during the time of disasters can greatly help in locating the exact source of the disaster and aid in efficiently allocating resources to bring relief to affected individuals. But, at the same time, a substantial amount of irrelevant or less informative data is generated which may hinder in analyzing the disastrous situation [1]. This motivated us in identifying and eliminating the false information which may be of no use to the social network. Our proposed work uses NLP techniques to generate embeddings for our supervised learning models. The proposed work comprises of system implemented using LSTM and BERT models on which comparative study is performed using precision, recall, accuracy and F1 score evaluation metrics. We use deep learning approach to determine if the given Tweet gives any useful information about the disaster or not. This is a binary classification task where we classify Tweets as 'Real' or 'Non-Real' disaster Tweets. Our experiments indicate that BERT state-of-art method performed better than the LSTM model which incorporated GloVe word embeddings.**

***Index Terms*—Text Classification, GloVe, BERT, Twitter**

I. Introduction

Social media are large complex networks, which are generally used to describe human relations. Complexity of these networks is increasing with the advent of social media platforms. Twitter is a social network, which is a widely popular platform for information diffusion. Recent studies show that users disseminate information on Twitter about events like disasters, election process, terrorism, etc. There is a lot of information that is diffused at the times of such events and there are studies being conducted to analyze the sentiments of the information from microblogs like Twitter and these are used to further address the needs of users by humanitarian organizations and news agencies. There lies a bigger question that needs to be addressed before sentiment learning which is if all the diffused information is reliable or not. A study was conducted on why fake news was a harder problem to solve. And the researchers studied that fake news travelled faster, farther and in greater depth than truth [1]. Further, the study showed that it took truth six times as long as false information to travel. False information also travels greater depth as compared to the truth. Also, considering the recent crisis about COVID-19, news agencies reported that Twitter along with agencies is working together to remove misinformation about COVID-19 pandemic and the information that could cause harm to people. Hence, it is quite important to address the problem of the reliability of information diffused on the occurrence of events through social media networks. Our work would aid in identifying such information and will lead to cascading of the data that has some social value and awareness, about occurrence of events. Addressing that issue would also help in reducing the manual classification of truth or fake content done by organizations and agencies. In our study, we have introduced semi-supervised models that use NLP techniques. We create GloVe vector word embeddings to input in our models. To further enhance our study, we make a comparative analysis between LSTM and BERT models. The analysis is performed on the metrics of accuracy, precision, recall and F1-score

II. Literature Review

There is only limited work done to identify the reliability of information produced on microblogs like Twitter. There are resaerch work done to determine the sentiments of the tweets to analyze the emotions after the disaster has occurred. Research is also performed to determine the most affected areas during a disaster. The studies performed by various researchers are event-specific like floods, earthquakes and fires. Data is collected based on event-specific hashtags and keywords to perform analysis. Research conducted by Barnwal et. al uses classification techniques for identifying fact checkable microblogs [2]. They use classification and ranking techniques and evaluate their models by precision on the top 100 tweets. Another study conducted by Sit et. al describes that LSTM model performs better for identifying disaster-related tweets by using word semantics. In research performed by Reem and Simon, they use word embeddings to classify tweets for crisis response and they achieved a result of 62.04% of F1 Score [3]. Based on the work being performed in similar fields, we propose to leverage Word Embeddings with supervised learning techniques to achieve greater results in determining if the tweet is reliable or not.

III. Project Details and Methodology

The methodology section consists of definitions followed by architecture of the proposed system, which are as follows:

A. Definitions

Global Vectors: GloVe is a learning algorithm commonly used to obtain vectors for word representations. Unlike other embeddings GloVe uses semantics from both local and global corpus. It computes co-occurrence matrix X where the total number of co-occurrences of i with j in the corpus is computed as:

$$X_{i,j} = \sum_{a}^{b} X_{i,a}$$

BERT (Bidirectional Encoder Representations from Transformers) : BERT is a bidirectional, unsupervised language representation which is pre-trained on a large corpus like Wikipedia. Traditional embedding models generate directional word embedding which is generated as a combination of token and position embeddings as shown in, Fig. 1. But BERT is designed to have bi-directional embedding, which represents both previous and next embedding making it bidirectional. It uses the Masking model (Mask Language Model - MLM) to mask some words from input and then generate a multi-layered context.

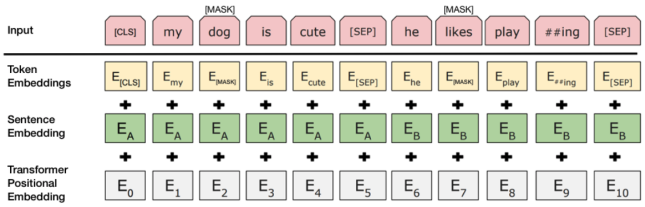

Fig. 1. Embedded input representation of BERT

B. Architecture

The proposed architecture can be simplified into three steps of Data Gathering, Data Pre-Processing, Data Analysis and System Modelling as shown in Fig 2.

1) Data Gathering: We collected tweets using the Tweepy API and web crawler script. We generated 1000 tweets where the challenge was with manually labelling them. The real-time tweets were collected in the span of 12th – 18th March. Hence, we have collected data from a Kaggle competition which consists of 7361 labelled data. The online corpus was created in a span of six days i.e., from 26th – 31st August 2015.

2) Data Preprocessing: Data preprocessing is a very important step while dealing with data sets for Machine Learning tasks. Because it leads to better data sets, that are cleaner and are more manageable and will lead to improved performance of our models. We do so by creating a corpus of data by eliminating stop words and punctuation which are key tasks in data preprocessing. Furthermore, we perform

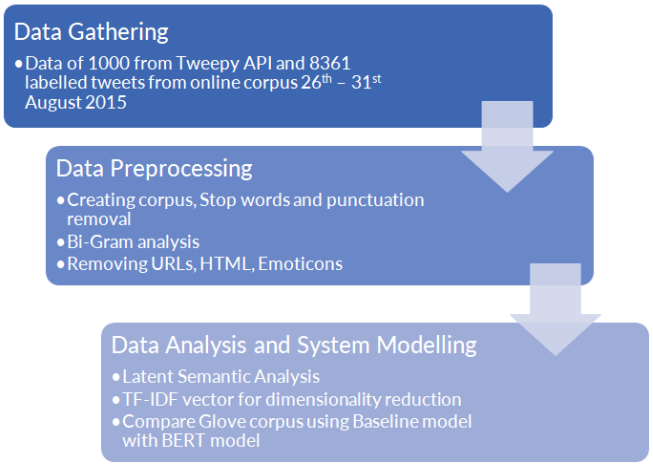

Fig. 2. Proposed Architecture

Bi-Gram analysis to eliminate unnecessary pairs of words with high frequency. The Data preprocessing steps are explained in detail:

a) Stop word removal: The twitter posts usually consist of many stop words, because of the usage of the online messaging/chatting language. Hence, this data needs a lot of preprocessing as we need to make meaningful insights. Fig 3. shows the high frequency of stop words in the tweets.

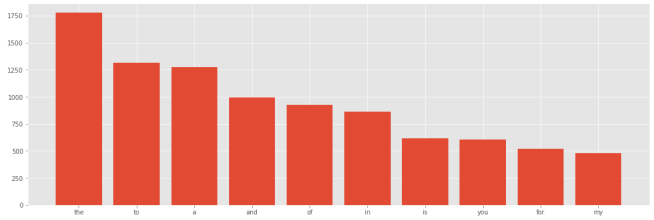

Fig. 3. Frequency of stop words in the tweets

b) Punctuation removal: As we have already noticed high frequency for stop words, there can be a high possibility for high frequency of punctuation usage while writing tweets. We generated a visualization for occurrence of punctuation in the tweets as shown in Fig. 4.

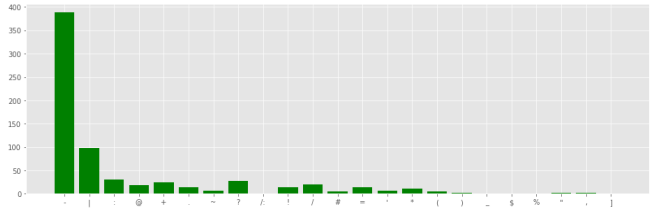

Fig. 4. Frequency of punctuation in Tweets

c) BiGram Analysis: Bigram is a sequence of two adjacent elements from a string of tokens, which are typically letters, syllables, or words. A bigram is an n-gram for n=2

[7]. The frequency distribution of every bigram in a string is commonly used for simple statistical analysis of text in many applications like text classification in our case. We performed bigram analysis on our dataset and Fig. 5 shows the high frequency bigram in our corpus of data.

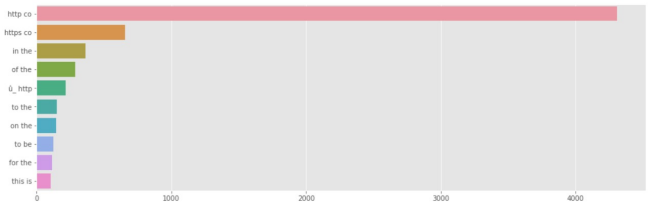

Fig. 5. BiGram Analysis

d) URL Removal: Utilizing the bigram analysis done in step 'C', by observing the highly frequent bigrams it was evident that there were mere URLs and Links present in our tweet dataset. As these links and URLs don't give out much valuable information so we eliminated them using regex expressions. Regex expressions are pattern matching techniques that find a pattern of 'https','www' and '.com' in the texts and eliminate them. After this step, the resulting data consisted of clean tweet text without any URLs.

e) Word cloud analysis: Finally, we generate word cloud for tweets belonging to both categories to generate a visual representation of words with their frequencies. In word cloud the more frequently a word appears the larger is its size compared to other words of the word cloud. The figure shows the two-word clouds for disaster and non-disaster tweets in Fig 6 & 7, respectively. In Fig. 7, we can see that there are usage of words which may not be necessarily related to disaster while words in Fig.6 (Disaster Tweets) give us a better gist of the disaster situations.

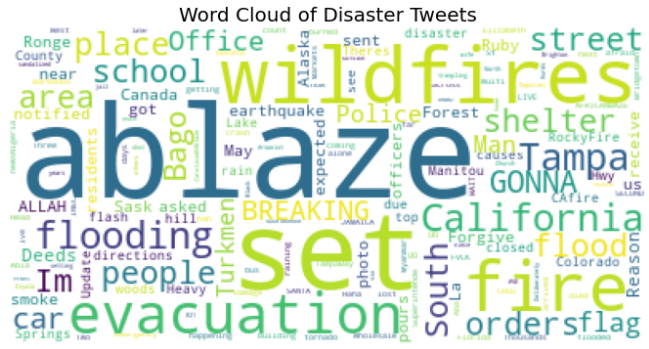

Fig. 6.

3) Data Analysis and System Modelling: We analyze the tweets by using a famous NLP technique, Latent Semantic Analysis to determine the relationships amongst words. We also generate a TF-IDF vector for input to our models. Dimensionality Reduction is performed before LSA and TF-IDF

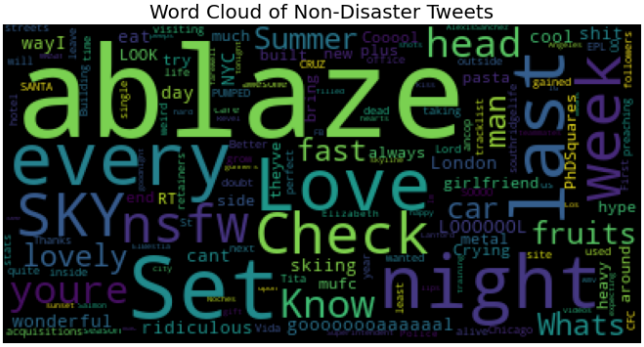

Fig. 7.

vectors. We utilize two models, LSTM and BERT models. For the GloVe model, we create GloVe embeddings and then use the output vector for the LSTM model. For the BERT model, we create an input corpus and train corpus. Input corpus is the vocabulary file that is a predefined English corpus. The training corpus is tokenized and used to input in the BERT layer of the model.

IV. Experimental Setup

This sections discuss more about the implementation aspect of the proposed system and the performance achieved by implementing our approach and the challenges we faced while working on the project.

A. Implementation Details

In this section, we present the implementation details for the Tweet analysis system developed for this study. The tweet analysis system is a binary classification module that consists of generating embedding vectors (numerical representation) of text and training the GloVe and BERT model to predict the category of tweet. There are two categories of the tweets we are trying to classify as mentioned above: Real tweets and Non-Real tweets about the disaster. The implementation details for the above experiment is described further. The system is implemented using python programming language. Additionally, a python script was written to scrape data from a widely known social media platform i.e. Twitter. For accessing Twitter via a python script 'Tweepy 3.8.0' API was configured and utilized. Tweepy is a great tool for simple automation and creating twitter bots that can extract data from Twitter consisting tweets, username, timestamp and Geo-location [8]. The data scraped from the Twitter platform was consolidated with an open source labelled dataset which consisted of 7361 tweets using Pandas library which a python data wrangling library. After consolidation of the datasets manual labelling was done to the 1000 tweets extracted from the social media platform. Once the dataset was consolidated and labelled, preprocessing of the dataset was done using NLTK library to remove URLs, stopwords and emoticons from tweets. Natural language toolkit (NLTK) is a suite of libraries and programs for symbolic and statistical natural language processing for English language [9]. Data analysis and pre-processing was done

TABLE I

Train-Test Ratio	Train set	Test set
GloVe	80%	20%
BERT	70%	30%

using Scikit Learn library. Scikit learn is a machine learning library for python programming language which offers various important features for machine learning such as classification, regression and clustering algorithms and data manipulation operations like preprocessing, dimensionality reduction and transformation of data [10]. After preprocessing the data, word embeddings were generated for the clean text using GloVe and BERT embedding modules. The embeddings were generated using the tokenization process which is implemented using GloVe and BERT libraries.

Experiments are conducted with the available implementation of GloVe and BERT model in python Keras library [11] using 5 & 10 epochs. Before the training process, the training data is split in the ratio mentioned in Table 1.

Model performance during the training process are visualized by plotting the accuracy and loss plots. Data visualization were done using python's Matplotlib and Seaborn library. Matplotlib provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+ [12].

Lastly, the training process was hosted on Google Colab which is a google cloud platform that provides python environment to train Machine Learning model [13]. The cloud instance is initiated with 12GB of memory with GPU enabled core.

B. Performance Evaluation and Findings

This sections describes the results attained from our proposed approach and detailed discussion of the findings from Accuracy and Loss plots of the models, Confusion matrix and from the Classification reports.

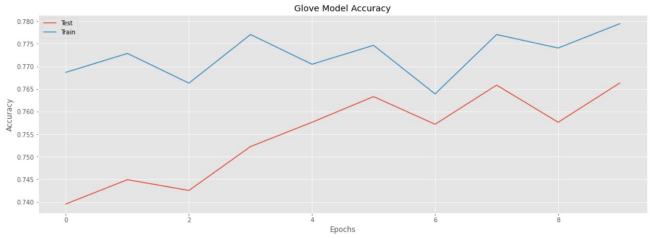

Fig. 8. Glove Model Accuracy

a) Accuracy and Loss plots: The plots give insights into the training process of the two models, the x-axis of the plots corresponds to the number of epochs model have been run and y-axis corresponds to the accuracy and loss measurements, respectively. The fig. 8 shows the accuracy plot of the GloVe model, as we can see there is an increasing trend in the line plot at the end which we can interpret that model can be further trained for more epochs but only with more training data otherwise the model will be stuck in the local minima. While, fig. 9 shows the loss plot of GloVe model, the loss values are quite close, which shows the model has comparable performance on train and test dataset and the model doesn't overfit or underfit our data.

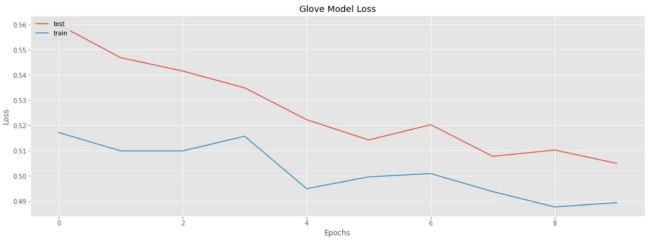

Fig. 9. Glove Model Loss

Fig. 10 shows the accuracy plot of BERT model, as we can see the we have decent test set accuracy of 85% but also the train accuracy surpasses the test accuracy, this shows that the model is overfitting. The model can be further trained by adding dropout to the layers of our model. Dropout is the technique where the data doesn't pass through few of the neurons chosen at random during the training phase. These neuron units are not considered during a particular forward or backward pass. This process will aid in generalizing our model better on unseen data and prevent the over-fitting of the model on training set.

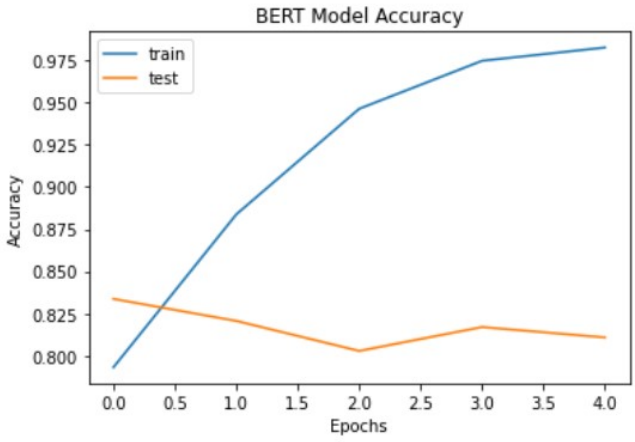

Fig. 10. BERT Model Accuracy

b) Confusion matrix: The two figures 11 & 12 corresponds to the confusion matrix of GloVe and BERT model. In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an error matrix, [14] is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning. Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class. It is a special kind of contingency table, with two dimensions ("actual" and "predicted"), and identical sets of "classes" in both dimensions [14]. As the

name says Confusion Matrix, it helps to evaluate if the model is confusing the classification of the two classes and leading to misclassification. Fig. 11 shows the confusion matrix for BERT model, the faded blue shades shows the wrongly predicted classes i.e 'False Negative', 'False Positive'. The model falsely classifies 297 samples as positive when they were negative, while 225 samples are classified as false negative when they were actually positive. The darker shades in the confusion matrix shows true classification were 1259 and 979 samples are True Positive True Negative, respectively. Lower the number of False Negative and False Positive better is the performance of the model.

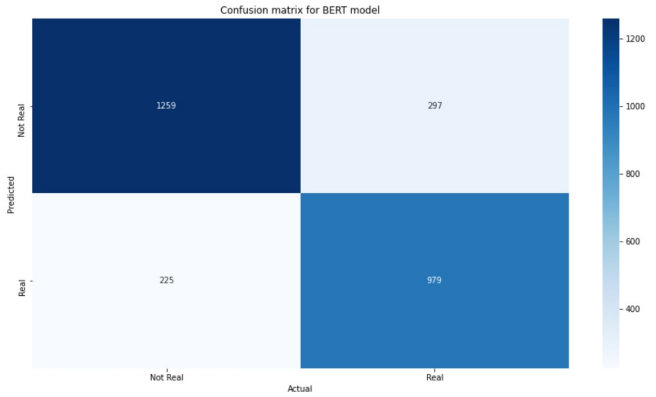

Fig. 11. Confusion Matrix BERT Model

Fig. 12 corresponds to the confusion matrix of GloVe model. The faded red shades of the matrix show the misclassification by the model. The model misclassifies 93 samples as falsely negative when they are actually positive and 276 samples as falsely negative when they are actually positive. The darker shades show accurate classification of tweets where 833 samples are correctly classified as True positive and they are actually positive and 471 samples are classified as True Negative and they are actually Negative.

c) Classification report: The classification report is used to measure the quality of predictions from a classification algorithm. More specifically, True Positives, False Positives, True negatives and False Negatives generated from confusion matrix are used to predict the metrics of a classification report as shown above. The report shows the main classification metrics precision, recall and f1-score on a per-class basis [15]. The metrics are calculated by using true and false positives, true and false negatives. Positive and Negative in this case are generic names for the predicted classes which is 'Real' 'Not Real' disaster tweets in our task. The classification report reflects all the evaluation metrics on which our model is evaluated. A detailed description of each metric follows; Precision - Precision is the ability of a classifier not to label an instance positive that is actually negative. For each class it is defined as the ratio of true positives to the sum of true and false positives. Recall - Recall is the ability of a classifier to find all positive instances. For each class it is defined as

TABLE II

Glove	Precision	Recall	f1-score	Accuracy
Class 0	0.75	0.90	0.82	0.78
Class 1	0.84	0.63	0.72	

the ratio of true positives to the sum of true positives and false negatives. Recall - The F1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. Generally, F1 scores are lower than accuracy measures as they embed precision and recall into their computation. As a rule of thumb, the weighted average of F1 should be used to compare classifier models, not global accuracy. Lastly, Accuracy – It is the ratio of truly classified samples to the total samples in the dataset. Selection and the importance given to each metric to evaluate the performance of the model depends upon the application of the system. Medical applications give more importance to the Precision metrics as it shows how precise is the trained model. For e.g., precision shows how many patients the model has classified as sick when they are actually sick. In our study, we have given importance to the precision and accuracy metrics, we did not consider the recall and F1-score as they play importance when there is high imbalance in the two classes we are trying to predict.

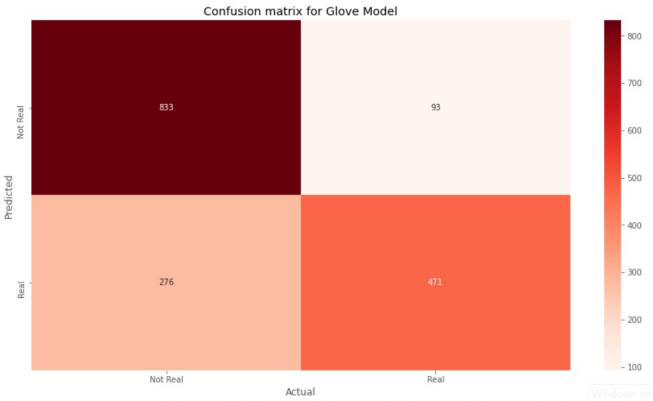

Fig. 12. Confusion Matrix GloVe Model

Table. II shows the classification report for GloVe model, as we can see that the model has achieved an accuracy of 78% which is pretty decent. But as discussed before the accuracy can further be increased if more training data is available. The model achieves a precision score of 75% for 'Class 0' and 84% for 'Class 1'. The model has lower confidence in predicting 'Class 0' which is 'Real' disaster tweets than 'Non-real' disaster tweets. This aspect of the metrics shows that model does not perform well in predicting class 'Real' tweets which is of importance to us.

Table. III shows the evaluation metrics for the BERT model. The model is trained on 70% of the dataset and achieved an accuracy of 82%, which is comparatively better than the GloVe model. The model achieved a precision of 77% on 'Class 1' i.e. Not-Real tweets, while for 'Class 0' i.e. Real tweets it achieved a precision of 85%. As the BERT model predicts

TABLE III

BERT	Precision	Recall	f1-score	Accuracy
Class 0	0.85	0.81	0.83	0.82
Class 1	0.77	0.81	0.79	

class 0 with a higher precision than the GloVe model, BERT gives a better performance in terms of classifying the tweets.

C. Challenges

The biggest challenge that we faced was extracting the useful data from the Twitter platform using Tweepy API. This was a keyword-based data extraction approach, where a keyword was given to the API and it would return only 100 Tweets at a time. So, to tackle this challenge a script was written which would autonomously hit Twitter in periodic intervals to extract the data and would append it at the end of the offline CSV file. Furthermore, the system compromises of a deep neural network model at core and training these models require a lot of training data. To work with such huge dataset and achieve a good accuracy on the model it takes a lot of computation power. The system that we were working on did not had an up-to the mark specifications and it took 5 hours to train a single model on our local system. This training process had to be repeated if the model parameters were fine tuned. So, we leveraged the Google cloud platform as the computation resource. We created an instance of 12GB memory and GPU enabled configuration. The training time for a single model on the Cloud platform took only half hour which saved a lot of computation time.

V. Conclusion

We have proposed a system to classify tweets into two categories (Real and Non-Real Tweets) by using the tweets extracted from Twitter and open source dataset and training deep learning models on top of it. With this system, the end-users can classify the tweets and determine if the sample tweets give any information about the real-world crisis events or not. The end-user can be an individual or an emergency relief organization. The emergency relief organization can benefit from our system by determining how many individuals are actually affected by the disaster and can effectively map the relief resources to them.

Experimental results showed that, in the comparative study been done, the GloVe model performs decently but the BERT model performs significantly better with the task of binary classification of tweets. By the utilization of the descriptive features generated from tweets using GloVe and BERT embeddings, our system provides a baseline to classify new tweets with a better accuracy. However, data with lost of noise can may degrade the performance of the proposed approach; hence noise removal techniques like elimination of stopwords, links, emoticons are necessary in such cases.

As a future work, we are planning to add more training data and train our model further. We believe that this would result in higher precision and accuracy to classify tweets in future. As the new training data would be added there won't be a need to manually label this dataset as we did before. The model is now trained to an accuracy of 82% so we can explicitly give this new data to our model and it will generate the labels for us as part of the prediction task. This predicted data can again be given to our model for training and to improve the accuracy. This will be a much more mature and incremental approach on improving the performance of the model.

Acknowledgment

We would like to thank Dr. Pooya Moradian Zadeh for providing us with all the resources and reference materials to help understand the concepts of Social Networks. Also, for giving valuable feedback on project progress during class project presentations. This project helped us to understand more about social networks and analyze the data generated by the users on Twitter platform.

References

[1] Vosoughi, Soroush Roy, Deb Aral, Sinan. (2018). The spread of true and false news online. Science. 359. 1146-1151. 10.1126/science.aap9559.

[2] Barnwal, D., Ghelani, S., Krishna, R., Basu, M., Ghosh, S. (2019, January). Identifying fact-checkable microblogs during disasters: a classification-ranking approach. In Proceedings of the 20th International Conference on Distributed Computing and Networking (pp. 389-392).

[3] Sit, M. A., Koylu, C., Demir, I. (2019). Identifying disaster-related tweets and their semantic, spatial and temporal context using deep learning, natural language processing and spatial analysis: a case study of Hurricane Irma. International Journal of Digital Earth, 12(11), 1205-1229.

[4] ALRashdi, R., O'Keefe, S. (2019). Deep Learning and Word Embeddings for Tweet Classification for Crisis Response. arXiv preprint arXiv:1903.11024.

[5] Naili, M., Chaibi, A. H., Ghezala, H. H. B. (2017). Comparative study of word embedding methods in topic segmentation. Procedia computer science, 112, 340-349.

[6] Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[7] https://en.wikipedia.org/wiki/Bigram

[8] https://medium.com/@jasonrigden/tweept-a-python-library-for-the-twitter-api-9d0537dcebd4

[9] https://en.wikipedia.org/wiki/Natural_Language_Toolkit

[10] https://medium.com/@deepanshugaur1998/scikit-learn-part-1-introduction-fa05b19b76f1

[11] https://keras.io/examples/pretrained_word_embeddings/

[12] https://en.wikipedia.org/wiki/Matplotlib

[13] https://colab.research.google.com/notebooks/intro.ipynbrecent=true

[14] https://en.wikipedia.org/wiki/Confusion_matrix

[15] https://muthu.co/understanding-the-classification-report-in-sklearn/