SOCIAL WEB FOR DISASTER MANAGEMENT

Dissertation submitted in partial fulfillment of the requirement for the degree of

BACHELOR OF ENGINEERING IN COMPUTER ENGINEERING

By

Nikita Salkar	**Seat No. : 0306**
Shravan Manerikar	**Seat No. : 0293**
Mugdha Khatavkar	**Seat No. : 0285**
Delstan D’souza	**Seat No. : 0259**

Project Guide: Dr. Kavita Asnani

Assistant Professor

Computer Engineering Department

Goa College of Engineering

GOA COLLEGE OF ENGINEERING

(GOVERNMENT OF GOA)

FARMAGUDI, PONDA, GOA – 403401

GOA UNIVERSITY

(2018-19)

GOA COLLEGE OF ENGINEERING

(GOVERNMENT OF GOA)

FARMAGUDI, PONDA, GOA - 403401

SOCIAL WEB FOR DISASTER MANAGEMENT

Bona fide record of work done by

Nikita Salkar	Seat No.: 0306	Shravan Manerikar	Seat No.: 0293
Mugdha Khatavkar	Seat No.: 0285	Delstan D’souza	Seat No.: 0259

Dissertation submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF ENGINEERING
IN
COMPUTER ENGINEERING

Of **GOA UNIVERSITY**

JUNE 2019

...………………………

Dr. Kavita Asnani

Faculty guide

…………..……………….

Dr. J. A. Laxminarayana

Head of the Department

………………………..

Dr. Krupashankar M. S.

Goa College of Engineering

Certified that the candidate was examined in the viva-voce examination held on
2nd day of July, 2019

….……………………..

(Internal Examiner)

…………………………..

(External Examiner)

DECLARATION

This research project titled “***Social Web for Disaster Management***” is a presentation of our original research work. Wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature, and acknowledgement of collaborative research and discussions.

The project has been done under the able guidance of Dr. Kavita Asnani, during the year 2018-19 at the Goa College of Engineering by

Nikita Salkar	Seat No. : 0306
Shravan Manerikar	Seat No. : 0293
Mugdha Khatavkar	Seat No. : 0285
Delstan D’Souza	Seat No. : 0259

In my capacity as the guide of this team, I certify that the above statements are true to the best of my knowledge.

Dr. Kavita Asnani
Project Guide
Computer Engineering Department
Goa College of Engineering

ACKNOWLEDGMENT

Hard work and perseverance are the corner stones of a successful venture.

Great things never come from comfort zones.

We would like to extend our heartfelt gratitude towards all those who have helped us learn, grow and accomplish this endeavor. Without their able guidance, help and encouragement, this project would not have been possible.

We express our sincere appreciation to our Principal, Dr. Krupashankar M. S. for providing us with all the facilities. We are indebted to our Professor, Dr. J. A. Laxminarayana, Head of Computer Engineering Department, for bolstering our confidence and providing his full-fledged support. We are profoundly thankful to our Professor, our project guide, Dr. Kavita Asnani, who has been a buttress throughout our journey and whose overwhelming encouragement and guidance helped us throughout this project. Also, this project would not have been possible without the help of IIT Kharagpur. We would, hence, like to extend our sincere thanks and gratitude to the Indian Institute of Technology, Kharagpur for their generous help. We also thank our teachers and other staff members of the Computer Engineering Department for their help and assistance. We deeply thank our families for their moral and economic support throughout this journey. Last but not least, we thank all our friends and acquaintances for making this voyage possible.

ABSTRACT

This project, ***Social Web for Disaster Management***, was conceived with the aim of exploring the possibilities of extending and amalgamating the benefits of technological advancement in the field of machine learning with disaster management.

Our project exploits the social web of twitter to investigate and verify the possibility of using machine learning techniques to assist in linking requirements and resources. The primary objective is to assay the output of machine learning techniques of clustering and deep learning applied to the data collected during crisis.

Disasters caused by natural hazards witness active communication on social media platforms such as twitter that then witnesses a deluge of tweets that are germane to the disaster. These tweets contain substantial information about situational awareness and the emergency of resources, classified as need tweets. Several humanitarian organizations and volunteers offer help and resources in the form of donations etc. to the victims, classified as availability tweets.

The main thrust, here, is to find whether an automatic link can be established for assisting in enabling all the resources to reach the victims well in time, using the machine learning techniques of clustering and deep learning. Using technical know-how we propose to study the problem to bridge this gap between victims and volunteers.

For this study we use the dataset of the 2015 Nepal Earthquake. This dataset contained 66,000 tweets out of which 44,921 tweets were retrieved using the Twitter API. This dataset of 44,921 tweets is then first pre-processed to filter out unwanted data. Then the cleaned information needs to be segregated and categorized into need tweets and availability tweets. We explore unsupervised clustering techniques and deep learning methods for this purpose.

Under clustering, we implement the K-means algorithm and the DBSCAN algorithm on the dataset. It was observed that the k means algorithm divided the 44,921 vectors into 2 clusters. The DBSCAN algorithm identified 13,091 of the total 44,921 tweets as outliers while the remaining 31,830 tweets of 44,921 were divided into 167 clusters.

Manual annotation was also performed on the 31,830 tweets obtained after removing the 13,091 outliers from the 44,921 tweets to reveal only 207 need tweets and just 374 availability tweets

indicating that a larger amount of substantial data was required to be detected distinctly and accurately by clustering algorithms for the purpose of optimal clustering.

From this, we learnt that the performance of unsupervised learning technique of clustering lends itself to exploitation for the purpose of grouping tweets into need tweets and availability tweets, marginally, as compared to word embedding based techniques as undertaken on prior occasions.

Deep learning techniques were also explored. Recurrent Neural Network was implemented in order to classify the tweets into the topical classes, such that output of this process will be served as an input for need-availability matching model. From the RNN architecture we used Long Short Term Memory or LSTM Algorithm for classification of data. It was found that the classification of need and availability was restricted to a precision 0.772 and 0.337 due to the small size of the substantial data in the dataset. Although it was a restriction, the method used by us yielded better results than that of the previous, which will be compared in the following chapters. It was learnt that had there been more training data, then the accuracy of the model would have increased drastically.

As part of future work, we intend to map the need tweets to that of availability, using the same approach used above for deep learning by using Manhattan LSTM.

We believe that much research still needs to be done on the project, after which, if implemented successfully, this project will become significantly relevant in the field of disaster management.

TABLE OF CONTENTS

Contents

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

1. API - Application Program Interface
2. ASCII - American Standard Code for Information Interchange
3. URL - Uniform Resource locator
4. t-SNE - T-distributed Stochastic Neighbor Embedding
5. DBSCAN - Density based Spatial Clustering of Applications with Noise
6. RNN - Recurrent Neural Network
7. LSTM - Long Short Term Memory
8. Sklearn – Scikit Learn
9. MATLAB - Matrix Laboratory
10. sgdm – Stochastic Gradient Descent with Momentum
11. RMSprop – Root Mean Square Propagation
12. adam – Adaptive Moment Estimation

Chapter 1
INTRODUCTION

1.1 Overview

This project, ***Social Web for Disaster Management***, studies advances in technology in the field of machine learning approaches and amalgamating the same with disaster management. Our project exploits the social web of twitter to investigate and verify the possibility of using machine learning techniques to assist in linking requirements and resources. The primary objective is to assay the output of machine learning techniques of clustering and deep learning applied to the data collected during crisis.

Natural and man-made disasters are unforeseen contingencies that warrant and witness urgent requirements of various resources in the disaster stricken areas. Over the past years the world has experienced several disasters such as bomb-blasts, earthquakes, hurricanes, cyclones, typhoons, floods etc.

When a disaster strikes, the overriding objective is to realize the resources that are required by the people struck by the disaster. Resources such as medical facilities, basic necessities i.e. food, water, shelter and infrastructural resources are imperative for survival. During these critical times it is paramount that these resources be made available to the victims as early as possible to contain the number of human casualties and to arrest the aftereffects. Faster dispatch of resources to victims is possible only after determining the urgently required resources and the resources that are available for dispatch.

Social media in today’s digital era has a profound and an impactful influence in various domains. Online social media covers a large geographical area and removes boundaries while diffusing information and so can also be exploited to help expedite the relief efforts when a disaster strikes.

Information and details of a disaster are promulgated across the globe via social media. News about casualties, the scale of the devastation, immediate requirements etc. are shared by social media users to spread awareness which enables the government, organizations and people willing to help, gain cognizance of the situation and pool in resources and donations in whatever form, to offer succor to the disaster-hit victims.

Social media sites such as twitter witness a deluge of tweets and become active reservoirs of information when a calamity strikes. For example, according to statistics, during the recent Kerala floods, more than 2.62 million tweets were shared across India as well as the globe [1]. Then again the hurricane Sandy that made landfall on 29th October, 2012 saw more than 20 million tweets within the 1st week itself. According to a research carried out by Pew Research Center, information, photos and videos made up more than half the twitter conversation [2]. Also during the 2011 Great East Japan Earthquake, around 55 million tweets were collected pertaining to the disaster.

Organizations and other individual users also take to twitter to share and spread information about various disaster relief camps, donations, volunteering programs and monetary and medical resources that are available for aid. Furthermore, tweets also contain data regarding the specific resources that are needed on-site. This information, pertinent to the ongoing disaster contains useful facts that can be used to accelerate the process of supplying resources. This information can be categorized into 2 groups, namely, need tweets and availability tweets.

- **Need tweets**: These are tweets that emphasize on the needs or requirements of specific resources of the affected people at the location of the disaster such as medical facilities, food, water, clothing, shelter etc.
- **Availability tweets**: These tweets highlight the resources at hand and potential humanitarian help that are intended to be offered by organizations and volunteers such as food and water packets, medical kits, donations, transportation facilities, blood donation camps etc.

Thus, using the need tweets and the availability tweets we can expedite the efforts of sending out the exact required appropriate resources to the exact locations that need those respective resources by matching the identified need-tweets with the availability-tweets. This helps the

Government bring order from chaos, provide basic necessities and to rehabilitate the affected regions with the aim of bringing back the state of disaster-struck areas to stability.

Our research aims at segregating the need-tweets and the availability-tweets. For the purpose of this project the dataset used is that of the 2015 Nepal earthquake. On 25th April, 2015, Nepal was struck by a devastating earthquake of magnitude 7.8 Mw.

Table 1 and Table 2 depict examples of tweets that are classified into need-tweets and availability-tweets:-

Need-Tweets
1) There is shortage of Blood as well as oxygen cylinders
2) Sir, in incidents like earthquake there is need for more orthopedicians appeal to States & center to send Ortho team to Nepal
3) You can donate water, old cloths, medicines, bandages, anything helpful for the Earthquake victims. Location :... http://t.co/MYNhGQ8Svu
4) RT @InfoMumbai: Urgent need of analgesic, antibiotics, betadiene, swabs in kathmandu!! Call for help 9851166044 #earthquake

Table 1: Need Tweets

Availability-Tweets
1) RT @ndtv: Nepal earthquake: DSGMC, SGPC to send 25,000 food packets every day to Kathmandu http://t.co/KMVuvHL8Lh http://t.co/w9w5EypllB
2) "@tajinderbagga: All 7 Gorkha Regiments of IndianArmy to send Nepali Gorkhajawans 2 #Nepal with medical officers 2 assist operations."
3) "@Komal_Indian: For blood requirements in Kathmandu Contact Mr. Adhikari 00977-9862005225"
4)Indian power minster Mr. @PiyushGoyal has offered to send engineers and equipment to restore the power grids in Nepal #EarthquakeNepal

TABLE 2: AVAILABILITY TWEETS

We take a look at methods for identifying tweets into tweets informing needs for resources (need tweets) and tweets and informing availability of resources (availability tweets). We initially discuss baseline methodologies followed by our proposed present work.

Prior work:

1) Purohit et al.: To classify tweets, into tweets informing about needs and tweets informing about availability of resources, prior works done by Purohit et al. [28] employed a set of 18 regular expressions for classifying tweets that ask for resources to be donated, or inform about donated resources.

2) Different from the prior works of Purohit et al., to classify tweets, again, into tweets informing about needs and tweets informing about availability of resources prior works have also included pattern matching methods [24] [25] and also word embedding based retrieval techniques like word2vec [26] for capturing the semantics of need and availability tweets for the purpose of tweet retrieval. For this purpose a dataset of 50,068 tweets pertaining to the

Nepal earthquake was used.
The study revealed that pattern matching methods could not identify the required need and availability tweets because the tweets seldom contained intuitively complementary terms such as “need”, “availability” or “distribute”.

On the other hand, the contextual word2vec based methods successfully identified need tweets and availability tweets. Thus, the performance of word2vec based methods is much superior compared to pattern matching method establishing the efficiency of contextual matching. The accuracy was found to be 18% and 45% for need and availability respectively.

Present work:

This study, in contrast to the above methods, employs unsupervised clustering and deep learning techniques for the same purpose of classifying tweets into need tweets and availability tweets.

The steps taken to accomplish the required tasks were as follows:

a) Extracting thousands of tweets that were relevant to the disaster.
b) Preprocessing the tweets, removing emoticons, abbreviations etc., multi-lingual tweets while keeping only English.
c) Transforming the tweets into vector representation.
d) Implementing clustering techniques to cluster into need tweets and availability tweets.
e) Deep learning.

In order to retrieve the tweets, the Twitter API is used to gain access to the tweets. Analysis of the information retrieved from the dataset requires categorization, cleaning and understanding before it is put to use. The data retrieved from the social media for the purpose of disaster relief first needs to be pre-processed to filter out unwanted data. Preprocessing techniques are used to clean and make the tweets, written in social media jargons, appropriate for use. The preprocessed tweets need to be segregated into need-tweets and availability-tweets. We then explore clustering techniques and deep learning methods with a focus on facilitating the same.

Thus, the principal objective of this study is classifying the extracted tweets during an on-going crisis into need-tweets and availability-tweets with a focus on linking the two. This linking of tweets will help the Government formulate a formative solution and to bring order from chaos. It will also give organizations direction in which to proceed speedily and help save time to extend their help and will channelize the coordination between emergency requests and relief offers.

1.2 Motivation

Time and speed are of essence. This is true especially during a crisis. This project seeks to assist in providing humanitarian succor to those affected by unforeseen crises. There is a lack of communication between the two sides due to which, sometimes, there may be delays and resources that are available may not reach the victims in crucial times. The tweets shared on twitter can be utilized to create a link between the two parties.

We were motivated to undertake this project by the remarkable study undertaken by the Indian Institute of Technology, Kharagpur. The study dealing with the exploitation of social media for emergency and faster disaster relief was brought to our notice by our guide. Their efforts served as an inspiration for us since this is a socially relevant issue pertaining to the assistance that can be provided to the victims of natural disasters. This project helps the Government formulate a formative solution and bring order from chaos. It will also give organizations direction in which to proceed speedily and help save time to extend their help and will channelize the coordination between emergency requests and relief offers.

More importantly, what mainly drew us towards this project is that this project helps us put our technical knowledge to use for a very noble cause.

Chapter 2
LITERATURE SURVEY

2.1 Dataset

The dataset of the Nepal Earthquake was obtained by putting out a request to IIT, Kharagpur. The dataset contains 66 thousand tweets posted during the course of Nepal Earthquake, April 2015. The tweets in the dataset are multilingual (comprising English, Hindi, Nepalese). The statistics of the dataset are as follows:

- **Task I: Need - Availability - Retrieval**
  - **Training set**
    - Contains 20 thousand tweet ids, i.e., identifiers of tweets posted on Twitter during the disaster.
  - **Test set**
    - Contains 46 thousand tweet ids, i.e., identifiers of tweets posted on twitter during the disaster.

- **Task II: Need - Availability – Matching**
  - **Training set**
    - Contains 200 correct matching of need-tweets and availability-tweets corresponding to the training set.
  - **Test set**
    - Contains 427 correct matching of need-tweets and availability-tweets corresponding to the test set.

* For the above two files, format of each line is *<Need-tweet-id>:<Availability-tweet-id1>,<Availability-tweet-id2>,..,<Availability-tweet-idN>*where the availability-tweets mentioned on a line are all correct matching for the need-tweet whose id is mentioned at the beginning of the same line.

We have retrieved:

- Training tweets:- 16834 valid tweets out of 20K.
    - Training need tweets:- 194 valid tweets out of 211.
    - Training availability tweets:- 624 valid tweets out of 719.
- Testing tweets:- 40772 valid tweets out of 46K.
    - Testing need tweets:- 368 valid tweets out of 427.
    - Testing availability tweets:- 813 tweets out of 980.

3166 training and 5228 testing tweets could not be retrieved since they were deleted or users that tweeted had deleted their accounts and were no longer twitter users.

The python code used to retrieve the tweets is as follows:

```
import tweepy
import sys
ACCESS_TOKEN = “XXXXXX”
ACCESS_TOKEN_SECRET = “XXXXXX”
CONSUMER_KEY = “XXXXXX”
CONSUMER_SECRET = “XXXXXX”
auth = tweepy.OAuthHandler(CONSUMER_KEY , CONSUMER_SECRET)
auth.set_access_token(ACCESS_TOKEN , ACCESS_TOKEN_SECRET)
api = tweepy.API(auth)
arr = [592632778451132417,592683588258308098,592549938073980928,
592279426030641153,592012112500666370,592542239563378688        ]
i=1
non_bmp_map = dict.fromkeys(range(0x10000, sys.maxunicode + 1), 0xfffd)
for id_of_tweet in arr:
try:
tweet = api.get_status(id=id_of_tweet)
except ConnectionError as c:
print(c)
exit
except tweepy.error.TweepError as e:
continue
print (i,':- ',tweet.text.translate(non_bmp_map),'\n')
i+=1
```

2.2 Challenges

1) Complex Segregation of tweets into situational and non-situational

During any natural or man-made disaster twitter proves to be, without doubt, the most prominent and the most significant social media tool used to spread situational awareness. Twitter witnesses millions of tweets being posted in a very short span of time. A very small number of these tweets are found to create situational awareness but a very large number of them are found to express opinions or sentiments of people or even spread rumors.

So, segregating the tweets spreading rumors or reflecting sentiments or opinions of people from tweets creating situational awareness is extremely arduous but never-the-less paramount and indispensably relevant for the mapping of availability tweets to need tweets. Research studies [22, 23] show how segregating these tweets are a crucial challenge.

2) Irrelevant data

The informational data collected from the social media sites is large but mostly has conversational content that is irrelevant. The tweets collected simply suggest *sadness, condolences, hope etc.* towards the people affected. Thus most of the data is noisy and so is rendered useless. From the data we retrieved, only 4.86% of data was relevant from training set. Such tweets need to be removed for optimum results.

3) Small size of tweets

The size of relevant tweets is very small. Tweets are restricted to word limit of 140 characters per tweet. The word limit on twitter restricts the capacity of people to exchange detailed information.

4) Unrecognizable twitter jargons

It is mostly seen that the conversational content in the tweets is large. Many times the tweets are informally written using a large number of abbreviations, slang and emoticons. The frequent use of abbreviations, slangs and emoticons again is a challenge when it comes to machine understanding and renders the data useless.
SlangSD consisting of over 90,000 slang words/phrases can be used to deal with the problem of abbreviations.

5) Diversity in nomenclature of resources

The need-availability mapping can be done only if both the need tweet and the availability tweet address the same resource. Need-availability pairs addressing same resources are easy to match than pairs addressing different resources or addressing same resources using different terms are difficult to match.

6) Limitations of supervised learning approach

The tweets collected are unlabeled and therefore supervised learning methods can be onerous to implement and also time consuming.

Thus the segregation of information collected into relevant and non-relevant and matching of the need tweet and the availability tweet is by and large challenging and hence needs further research.

2.3 Preprocessing

Tweets have a stringent word limit, and users often make use of innovative abbreviations which are difficult to handle for retrieval systems. They are mostly informal and may involve the use of multiple languages in the same tweet (called code mixing), or even multiple scripts in a tweet. It is also difficult to make sense of emoticons, and informal short-hands especially innovative ones made up by users.

Preprocessing involves performing all the clean-up jobs on the provided tweets.

From the retrieved tweets, only alphabets will considered and converted to lowercase. Stop-words will be removed [7].

Hashtags and usernames starting with # and @ respectively will be pruned from every tweet [7]. URLs i.e. words starting with HTTP or http present in the tweets will also be removed. The duplicate tweets and the retweets available in the set are excluded. Punctuations will also have to be removed.

Preprocessing is done to facilitate word embedding or other techniques that will help in classifying tweets and in turn matching need tweets with availability tweets [10].

Text preprocessing operations include:

- Removing non-ASCII characters.
- Removing all the stop-words (also for Hindi and Nepali).
- Normalizing all the text of tweets into lowercase.
- ASCII transliterations of Unicode text (Tweet).
- Separation of sentence into tokens (words).
- Stemming of social media tweets using the standard Porter stemmer.
- Translate tweets using Google Translate.
- Pruning words starting with hashtag or username starting with @.
- Removing URLs.
- Removing duplicate tweets and retweets.
- Removal of punctuation.

An example of preprocessing is given in Example 1.

Example 1:

Original tweet: 11 :- we are deprived of homes
&amp; are on street and basecamps
many are injured
no foods
no water
#prayfornepal earthquake http://t.co/wBTI6M2VCD

No special entitites: 11 :- we are deprived of homes
are on street and basecamps
many are injured
no foods
no water
#prayfornepal earthquake http://t.co/wBTI6M2VCD

No hyperlinks: 11 :- we are deprived of homes
are on street and basecamps
many are injured
no foods
no water
#prayfornepal earthquake

No hashtags: 11 :- we are deprived of homes
are on street and basecamps
many are injured
no foods
no water
earthquake

```
No punctuation: 11     we are deprived of homes
  are on street and basecamps
 many are injured
 no foods
 no water
  earthquake
No small words:       are deprived  homes
  are  street and basecamps
 many are injured
  foods
  water
  earthquake
No whitespace: are deprived homes are street and basecamps many are injured foods water earthquake
No emojis: are deprived homes are street and basecamps many are injured foods water earthquake
Tweet tokenize: ['are', 'deprived', 'homes', 'are', 'street', 'and', 'basecamps', 'many', 'are', 'injured', 'foods', 'water', 'earthquake']
No stop words: ['deprived', 'homes', 'street', 'basecamps', 'many', 'injured', 'foods', 'water', 'earthquake']
Final tweet: deprived homes street basecamps many injured foods water earthquake
```

As observed in the dataset, tweets are multilingual and code mixed. The word count of the tweets in the dataset is given in Table 3.

	English(Approx)	Hindi(Approx)	Nepali(Approx)
Training Availability	12400	693	57
Training Need	3690	82	68
Testing Availability	13000	1106	694
Testing Need	6700	374	236

TABLE 3:WORD COUNT

FIGURE 1: STATISTICS

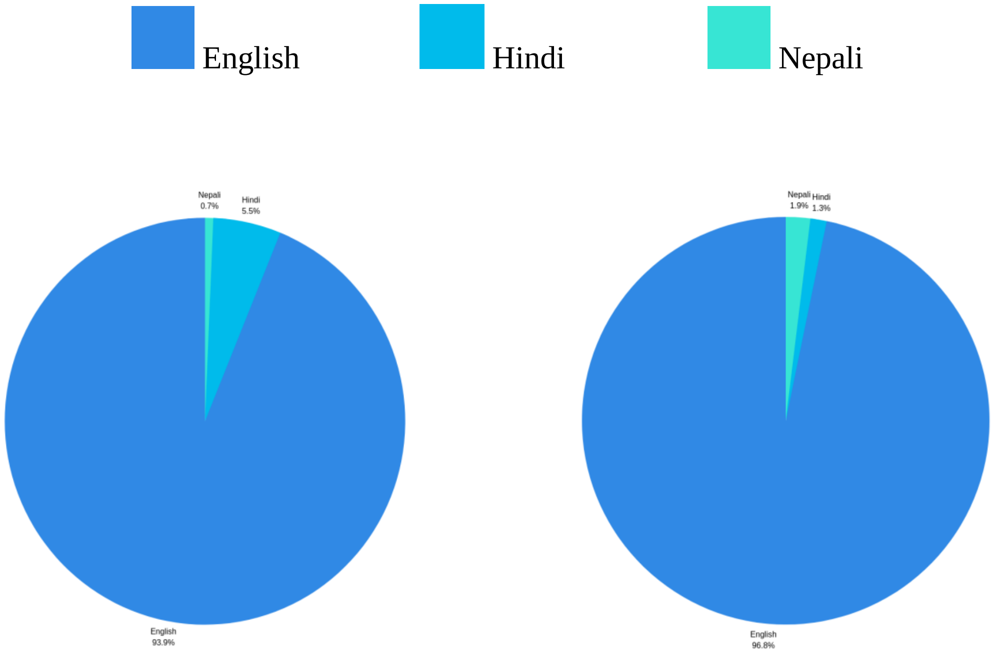


FIGURE 1.1: TRAINING AVAILABILITY

FIGURE 1.2: TRAINING NEED

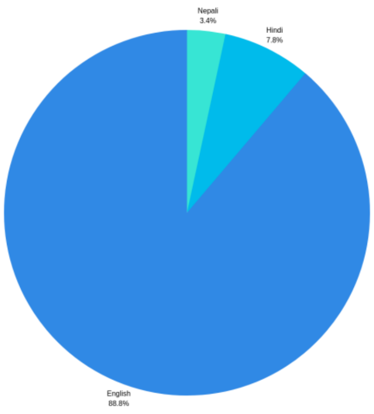


FIGURE 1.3: TESTING AVAILABILITY

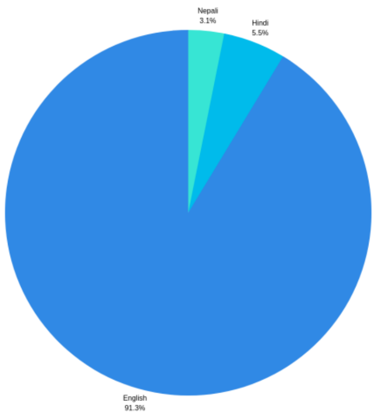


FIGURE 1.4: TESTING NEED

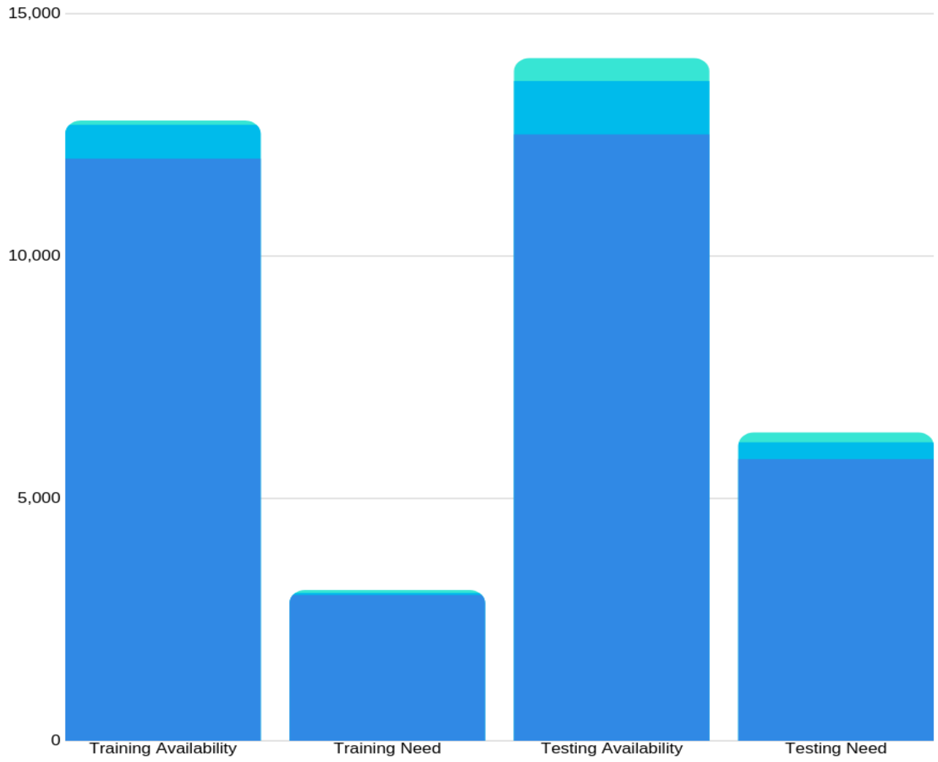


FIGURE 1.5: GRAPH

Figure 1.1 and Figure 1.2 shows the word count of need tweets and availability tweets respectively from the classified training set.

Figure 1.3 and Figure 1.4 shows the word count of the classified testing set.

Figure 1 shows that English tweets provide more information compared to tweets in other languages or scripts, but Non-English tweets also give important information about a situation and cannot be ignored. Hence, we tried using Google Translate to translate Non-English tweets. We observed that the translations were not hundred percent accurate.

We have preprocessed the entire dataset. The classified training set includes a total of 766 english tweets and the non-classified training set consists of 10900 English tweets. The testing set consists of 34,021 english tweets.

Original Data:

	Training	**Testing**
Monolingual	Tweets: 11001 Words: 218000	Tweets: 34190 Words: 668100
Mixed	Tweets: 9000 Words: 170000	Tweets: 5023 Words: 99200

Table 4:Tweet And Word Count Of Entire Dataset

	Training		**Testing**	
	Need	**Availability**	**Need**	**Availability**
Monolingual	Tweets: 186 Words: 3690	Tweets: 580 Words: 12400	Tweets: 326 Words: 6700	Tweets: 690 Words: 13000
Mixed	Tweets: 8 Words: 150	Tweets: 34 Words: 750	Tweets: 38 Words: 610	Tweets: 116 Words: 1800

Table 5: Tweet And Word Count Of Segregated Data

After Preprocessing:

Training	Testing
Tweets: 10900 Words: 45000	Tweets:34021 Words: 289400

TABLE 6:TWEET AND WORD COUNT AFTER PREPROCESSING(ENTIRE DATASET)

Training		Testing	
Need	Availability	Need	Availability
Tweets: 186 Words: 1950	Tweets: 580 Words: 6400	Tweets: 322 Words: 3200	Tweets: 690 Words: 6860

TABLE 7:TWEET AND WORD COUNT AFTER PREPROCESSING(SEGREGATED DATA)

The code for preprocessing is as follows:

```
import re
import codecs
from string import punctuation
from nltk.corpus import stopwords
from nltk.tokenize import TweetTokenizer
punctuation += '′′’…“”——―»«' # string.punctuation misses these.
cache_english_stopwords = stopwords.words('english')
def tweet_clean(tweet,f1):
tweet_no_special_entities = re.sub(r'\&\w*;', '', tweet)
                tweet_no_tickers = re.sub(r'\$\w*', '', tweet_no_special_entities)
```

```
tweet_no_small_words = re.sub(r'\b\w{1,2}\b', ", tweet_no_punctuation)

tweet_no_whitespace = re.sub(r'\s\s+', ' ', tweet_no_small_words)

tweet_no_whitespace = tweet_no_whitespace.lstrip(' ') # Remove single space remaining at the front of the tweet.

tweet_no_emojis = ".join(c for c in tweet_no_whitespace if c <= '\uFFFF') # Apart from emojis (plane 1), this also removes historic scripts and mathematical alphanumerics (also plane 1), ideographs (plane 2) and more.

tknzr = TweetTokenizer(preserve_case=False, reduce_len=True, strip_handles=True) # reduce_len changes, for example, waaaaaayyyy to waaayyy.

list_no_stopwords = [i for i in tw_list if i not in cache_english_stopwords]

tweet_filtered =' '.join(list_no_stopwords)

print(tweet_filtered)

f1.write(tweet_filtered + '\n')

tweet_no_hyperlinks = re.sub(r'https?:\/\/.*\/\w*', ", tweet_no_tickers)

tweet_no_hashtags = re.sub(r'#\w*', ", tweet_no_hyperlinks)

tweet_no_punctuation = re.sub(r'[' + punctuation.replace('@', ") + ']+', ' ', tweet_no_hashtags)
```

2.4 Data Representation

Our first approach to separate the need-tweets and availability-tweets is to implement various clustering algorithms. The dataset that needs to be clustered consists of textual tweets. In order to make the data suitable for clustering we need to convert the tweets into vector representation. Vectorization is essential since machine learning techniques can work only with numerical data and not on textual tweets.

The first method we implemented to accomplish vectorization is the Word2vec model.

2.4.1 Word2vec

Word2vec is a double layered neural network model that, given a corpus, generates vector representations of the words in the corpus. Textual data is converted into numerical data that can be used for clustering.

The utility of word2vec is seen from the fact that, given a vector space, word2vec can identify the words that are mathematically similar to one another. Word2vec can capture the meaning of words semantically. It thus, places homogenous words closer in the vector space and heterogeneous words farther apart.

Word2vec thus converts each word in the tweets in our dataset, into vectors that have multiple dimensions. However in order to visualize the vector space which is in high dimensions, dimensionality reduction needs to be performed which is done using t-SNE.

T-distributed Stochastic Neighbor Embedding (t-SNE)

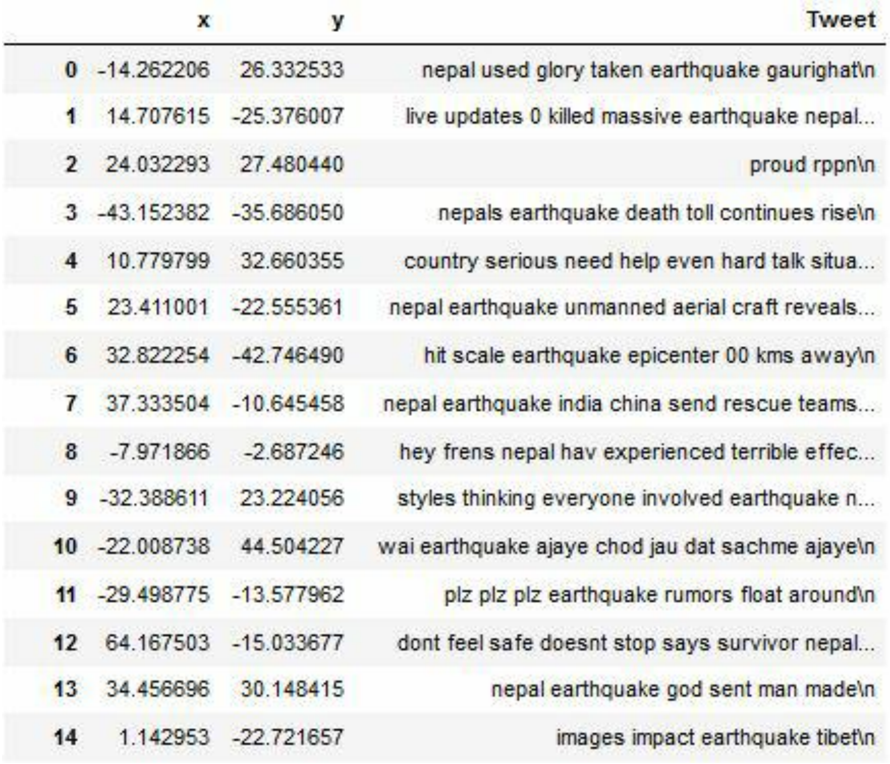
T-SNE is a machine learning algorithm for dimensionality reduction that enables visualization of data. It reduces multi-dimensional vectors into say 2 dimensional vectors all the while preserving the semantic relation between the vectors. That is, vectors that are similar and thus closer in the multi-dimensional space are correspondingly close in the 2 dimensional space and vectors dissimilar and thus far earlier are proportionately placed in the 2D graph.

After dimensionality reduction of the word vectors, the vectors of the words comprising the tweet need to be modified to form a representation of the tweet as a whole, in order to represent the entire tweet. One approach to do this is to calculate the average of all the vectors of the words in the tweet. A neater method is doc2vec.

2.4.2 Doc2vec

Doc2vec is essentially a generalization of word2vec. While word2vec generates vector representations of individual words in a document, doc2vec as the name implies, generates vector representations of the entire document. The document in this case is an individual tweet. Thus doc2vec automatically creates vector representations of the tweets. Having the same properties as that of word2vec, doc2vec also considers similarity of documents thereby plotting similar tweets nearer that dissimilar ones.

A tweet can be represented in a multi-dimensional space. However for the ease of visualization, the dimensionality has been reduced to 2 i.e. x dimension and y dimension using t-SNE. Following are examples of how the tweets are represented in 2D:



	x	y	Tweet
0	-14.262206	26.332533	nepal used glory taken earthquake gaurighat\n
1	14.707615	-25.376007	live updates 0 killed massive earthquake nepal...
2	24.032293	27.480440	proud rppn\n
3	-43.152382	-35.686050	nepals earthquake death toll continues rise\n
4	10.779799	32.660355	country serious need help even hard talk situa...
5	23.411001	-22.555361	nepal earthquake unmanned aerial craft reveals...
6	32.822254	-42.746490	hit scale earthquake epicenter 00 kms away\n
7	37.333504	-10.645458	nepal earthquake india china send rescue teams...
8	-7.971866	-2.687246	hey frens nepal hav experienced terrible effec...
9	-32.388611	23.224056	styles thinking everyone involved earthquake n...
10	-22.008738	44.504227	wai earthquake ajaye chod jau dat sachme ajaye\n
11	-29.498775	-13.577962	plz plz plz earthquake rumors float around\n
12	64.167503	-15.033677	dont feel safe doesnt stop says survivor nepal...
13	34.456696	30.148415	nepal earthquake god sent man made\n
14	1.142953	-22.721657	images impact earthquake tibet\n

FIGURE 2: DATA REPRESENTATION OF TWEETS-1

2.5 Clustering

The first phase of the project for automatic mapping of resource availability tweets to requirement tweets or need tweets entails the separation of situational awareness tweets into need-tweets and availability-tweets. The unsupervised machine learning technique of clustering is essentially employed since the tweets are unlabeled and furthermore the tweets also require grouping.

Clustering is an unsupervised approach to machine learning that, given a set of unlabeled data, clusters the data items in groups or clusters such that each group contains items that are similar to each other while items in different groups are dissimilar.

The algorithm is such that the machine learns of the similarities and dissimilarities without external guidance. The output of clustering then solely depends on the algorithm.

2.5.1 K-means Algorithm

The k-means algorithm is a partitioning based clustering method that partitions the given dataset into ‘k’ specified number of clusters. We need to partition the situational awareness data into 2 categories, namely, needs and availability. Since we know a priori that the number of clusters known or expected is 2, i.e. need cluster and availability cluster, we implement the k-means algorithm with parameter k =2.

2.5.2 DBSCAN Algorithm

The Density based Spatial Clustering of Applications with Noise or the DBSCAN algorithm is a density based clustering algorithm. The crux of this algorithm is that it clusters regions of high density. And since the output of the doc2vec model places similar tweets close to each other in the vector space, the DBSCAN algorithm can detect the regions of high density formed by similar clusters. The dataset, comprising of social media data, has a large amount of noise. The DBSCAN algorithm works well in identifying outliers and can detect to a large extent, the noise present in the dataset.

It requires the following as parameters:

- **Epsilon (eps):** The radial distance from a data point to check the neighbors of that point.
- **MinPts**: The minimum number of points required in the eps neighborhood of a point in order to form a dense region.

The DBSCAN terminology incorporates 3 major points:

- ❖ Core points: The points in the vector space that have at least '**Minpts'** number of points in their eps neighborhood.
- ❖ Border/ Boundary points: The points which do not have Minpts number of points in their eps neighborhood but themselves lie in the neighborhood of some core point. They by themselves lack the capacity to attract density.
- ❖ Outliers: These are points that do not lie in the vicinity of any core points and thus have deviating properties from the rest of the points.

Both the algorithms can be implemented using the machine learning library scikit learn or sklearn.
Another approach to segregate the data into needs and availability and thus to classify the same is using deep learning methods.

2.6 Deep learning

Deep learning can be defined as neural networks with a large number of parameters and layers in one of four fundamental network architectures:

- Unsupervised Pre-trained Networks
- Convolutional Neural Networks
- Recurrent Neural Networks
- Recursive Neural Networks

We implemented the Recurrent Neural Network in order to classify the tweets into the topical classes, such that output of this process will be served as an input for need-availability matching model.

The reason behind choosing RNN over other methods such as CNN is because it is specifically developed for the image data and not for the textual data. More generally, it works well with the data that has a spatial relationship. CNN input is traditionally 2D, a field matrix but can also be transform to 1D. But unlike CNN, RNN were designed to work with sequence prediction problems.Some examples of sequence prediction problems include:

- **One-to-Many**: An observation as input mapped to a sequence with multiple steps as an output.
- **Many-to-One**: A sequence of multiple steps as input mapped to class or quantity prediction.
- **Many-to-Many**: A sequence of multiple steps as input mapped to a sequence with multiple steps as output.

Our problem belongs to the category of Many-to-One i.e sequenced input to be matched to a particular class. Recurrent neural networks were traditionally difficult to train. The Long Short-Term Memory, or LSTM, network is perhaps the most successful RNN because it overcomes the problems of training a recurrent network and in turn has been used on a wide range of applications. Hence from the RNN architecture we used LSTM Algorithm for classification of data.

LSTM-Long Short Term Memory: Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feed forward neural networks, LSTM has feedback connections that make it a "general purpose computer" (that is, it can compute anything that a Turing machine can. It can not only process single data points (such as images), but also entire sequences of data (such as speech or video). For example, LSTM is applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition). A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three *gates* regulate the flow of information into and out of the cell.

LSTM networks are well-suited to classifying, processing and making predictions based on time series or sequence data. LSTMs were developed to deal with the exploding and vanishing gradient problems that can be encountered when training traditional RNNs.

Training includes a selection of the optimizers used for learning of the model they are :

1. Stochastic Gradient Descent (**sgdm)**

   The stochastic gradient descent algorithm can oscillate along the path of steepest descent towards the optimum. Adding a momentum term to the parameter update is one way to reduce this oscillation. The stochastic gradient descent with momentum (SGDM) update is

   $\theta_{\ell+1}=\theta_{\ell}-\alpha\nabla E(\theta_{\ell})+\gamma(\theta_{\ell}-\theta_{\ell-1})$,

   where γ determines the contribution of the previous gradient step to the current iteration. To train a neural network using the stochastic gradient descent with momentum algorithm,

2. Root Mean Square Propagation (**rmsprop)**

   Stochastic gradient descent with momentum uses a single learning rate for all the parameters. Other optimization algorithms seek to improve network training by using learning rates that differ by parameter and can automatically adapt to the loss function being optimized. RMSProp (root mean square propagation) is one such algorithm. It keeps a moving average of the element-wise squares of the parameter gradients,

   $$v_\ell=\beta_2 v_{\ell-1}+(1-\beta_2)[\nabla E(\theta_\ell)]^2$$

   β_2 is the decay rate of the moving average. Common values of the decay rate are 0.9, 0.99, and 0.999. The corresponding averaging lengths of the squared gradients equal $1/(1-\beta_2)$, that is, 10, 100, and 1000 parameter updates, respectively. The RMSProp algorithm uses this moving average to normalize the updates of each parameter individually,

   $$\theta_{\ell+1}=\theta_\ell-\frac{\alpha\nabla E(\Theta)}{\sqrt{v_l}+\epsilon}$$

   Where the division is performed element-wise. Using RMSProp effectively decreases the learning rates of parameters with large gradients and increases the learning rates of parameters with small gradients. ε is a small constant added to avoid division by zero

3. Adaptive Moment Estimation (**adam)**

   Adam is an update to the RMSProp optimizer. In this optimization algorithm, running averages of both the gradients and the seconf moments of the gradients are used. Given loss function θ_ℓ, where t indexes the current training iteration(indexed at 0). Adam uses a parameter update that is similar to RMSProp, but with an added momentum term. It keeps an element-wise moving average of both the parameter gradients and their squared values,

$m_\ell = \beta_1 m_{\ell-1} + (1-\beta_1)\nabla E(\theta_\ell)$

$v_\ell = \beta_2 v_{\ell-1} + (1-\beta_2)[\nabla E(\theta_\ell)]^2$

Where $\boldsymbol{\beta_1}$ and $\boldsymbol{\beta_2}$ are decay factors *viz,* Gradient Decay Factor and Squared Gradient Decay Factor' respectively. Adam uses the moving averages to update the network parameters as

$$\theta_{\ell+1} = \theta_\ell - \frac{\boldsymbol{\alpha m_l}}{\sqrt{\boldsymbol{v_l}} + \boldsymbol{\epsilon}}$$

If gradients over much iteration are similar, then using a moving average of the gradient enables the parameter updates to pick up momentum in a certain direction. If the gradients contain mostly noise, then the moving average of the gradient becomes smaller, and so the parameter updates become smaller too. The full Adam update also includes a mechanism to correct a bias the appears in the beginning of training

Once the Correct Representation of the Data is being done by encoding it using word encoding; the data now can be used to train the classifier model based on the methods which produce the output more accurately, Once the model of the training data set is ready, it could then be exported and, be used for the upcoming testing data set, so as to obtain the classification. This can be done using MATLAB, using it’s one of the applications for training the classifier model and same can be used for running the test data and finding the classified tweet.

Chapter 3
PROJECT OBJECTIVES

3.1 Project Objectives

- A study was conducted to exploit the social web of twitter to investigate and verify the possibility of using machine learning techniques to assist in linking requirements and resources.
- The primary objective was to assay the output of machine learning techniques of clustering and deep learning applied to the data collected during crisis and ascertain the utility of the said techniques.
- This project looks to assist in amalgamating the technological advances and progress with disaster management. The era of technological advancements where technology is used in practically all walks of life, prompts us to extend the benefits of its progress even to natural disasters, on humanitarian grounds.
- We sought to use the results of machine learning techniques to coordinate the rescue and relief efforts and to minimize the loss of life and to stabilize, at a fast pace, the lives of people adversely affected in the disaster stricken areas.
- This was hoped to be achieved by automatically mapping and matching the tweets defining the needs and requirements of the affected regions with the resource availability tweets using machine learning techniques of clustering and deep learning so that it could serve as a ready reckoner for any future unforeseen contingencies.

3.2 Project Methodologies

Baseline Methodologies

We discuss here methods for identifying tweets, posted during disaster events, into tweets informing about need for resources and tweets informing about availability of resources. We initially look at baseline methodologies followed by our proposed approach.

1) Purohit et al. : To classify tweets, into tweets informing about needs and tweets informing about availability of resources, prior works done by Purohit et al. [28] employed a set of 18 regular expressions for classifying tweets that ask for resources to be donated, or inform about donated resources.

2) Different from the prior works of Purohit et al., to classify tweets, again, into tweets informing about needs and tweets informing about availability of resources, prior works have also included pattern matching methods [24] [25] and also word embedding based retrieval techniques like word2vec [26] for capturing the semantics of need and availability tweets for the purpose of tweet retrieval. For this purpose a dataset of 50,068 tweets pertaining to the Nepal earthquake was used.

The existing methodology employs Information Retrieval (IR) techniques where the information needs consists of need and availability resources and where tweets corresponding to each information need are sought to be retrieved. A query that consists of terms corresponding to the information need is posted and tweets that contain the query terms are sought to be retrieved. Then, based on how relevant it is to the query, ranking of the retrieved tweets is done. There are 2 stages in the retrieval process. Initially the tweets are retrieved using a base query and then later the query is extended using additional specific terms. Then the 2nd round of retrieval is carried out with the extended specificity of the query.

a) Retrieval with initial query: A few terms from the initial queries are selected on the basis of intuition and observation of need-tweets and availability-tweets. To retrieve need-tweets an initial query is used consisting of the two terms – ‘need’ and ‘requir’. To retrieve availability tweets, the initial query that consists of three terms – ‘avail’, ‘distribut’ and ‘send’ is used. Then, for retrieving with the queries the 2 models mentioned below are used:

i) Language modeling: Using the Indri system the pre-processed tweets are indexed. Then using the same default retrieval model of Indri, the tweets are ranked and retrieved for the initial queries.

ii) Word embedding based model: A Word2vec based model for retrieval was trained on the tweets of the dataset.

The tweets are ranked in decreasing order of the cosine similarity which is based on the each query vector and the corresponding tweet vector.

b) Query expansion: Query expansion is performed using the following techniques:

i) Rocchio expansion

ii) Query expansion using Word2vec: This is done by calculating the cosine similarity between the query vector and the term vector of every distinct term of the dataset.

Present Methodology

- The project entails the separation of the dataset of tweets obtained using the twitter API, into need-tweets and availability tweets and linking the two.

- Analysis of the information retrieved from the dataset requires categorization, cleaning and understanding before it is put to use. The data retrieved from the social media for the purpose of disaster relief first needs to be pre-processed to filter out unwanted data.

- Then the cleaned information needs to be categorized as need and availability data. This can be achieved using different methodologies, but for this project, this was tried using the machine learning techniques of clustering and deep learning.

- The task of separating the dataset into need tweets and the availability tweets was carried out by exploring different clustering techniques including the k-means algorithm and the DBSCAN algorithm.

- The k-means algorithm is a partitioning based clustering method that partitions the given dataset into 'k' specified number of clusters.

- The Density based Spatial Clustering of Applications with Noise or the DBSCAN algorithm is a density based clustering algorithm. The crux of this algorithm is that it clusters regions of high density. The DBSCAN algorithm works well in identifying outliers and can detect to a large extent, the noise present in the dataset.
  Hence, these two algorithms were considered for experimenting.

- The task of separation was also done using deep learning techniques. The RNN was used to classify the tweets to serve the output of need-availability matching model, as input.

- The LSTM Algorithm from RNN architecture is used to classify the data. Like Turing Machine, LSTM algorithm can work with single data points as well as stream of data. LSTMs were developed to deal with lags of unknown duration between important events in a time series that can be encountered when training traditional RNNs.

- Once the training data is word encoded it can now be used to train the classifier model. Once the model of the training data set is ready, using it can now be used to obtain classification of the testing data.

- Manual annotation was also performed to reveal 207 need tweets and 374 availability tweets. These tweets on scrutiny were found to belong to certain categories some of which are listed below:

	Category	Examples
1.	Evacuation	Bihar govt sending 10 buses from Muzaffarpur today to evacuate earthquake victims in Pokhra…
2.	Blood Donation	Kathmandu hospitals struggling after Nepal earthquake. Generous Blood donors needed asap…
3.	Helplines	RT @insanneha: #MSGHelpEarthquakeVictims Helpline Numbers in Nepal Bishal Bazaar Ambulance service: +977 4244121 Red cross ambulance…
4.	Tracking	India 10 NDRF teams, 400 ppl, in Nepal r equipped with sniffer dogs, steel&amp; concrete cutters, sensors to detect dead/alive…
5.	Donation	1,00,000 Food Packets to Earthquake hit Nepal from the Golden Temple daily….
6.	Medical aid	Rushed 15 tonnes of medicines,34-member team of doctors and medical experts to quake hit areas of Nepal…
7.	Shelter	RT @Krittivasm: As #Nepal I&amp;B minister Rijal said 5,000 tents needed to house patients forced out of damaged hospital wards. https://t.co/v...

8.	Others	1) @UPGovt @yadavakhilesh will send 10 Trucks of Water,10 Trucks of Biscuit and a Truck of Life-Saving Medicines to Nepal #IndiaWithNepal 2) RT @TimesNow: Piyush Goyal has offered to send engineers &amp; equipment to restore the power grids in Nepal: Sushma Swaraj… 3) RT @rahulsNDTV: Pm orders massive relief for Nepal. 15000 blankets and 2500 tents ready at Kanpur airport. One C130J aircraft leaves wit…

TABLE 8: CATEGORIES AND EXAMPLES FOUND IN THE NEED/ AVAILABILITY TWEETS

Chapter 4
DESIGN

BLOCK DIAGRAM OF THE DIFFERENT STAGES OF THE PROJECT

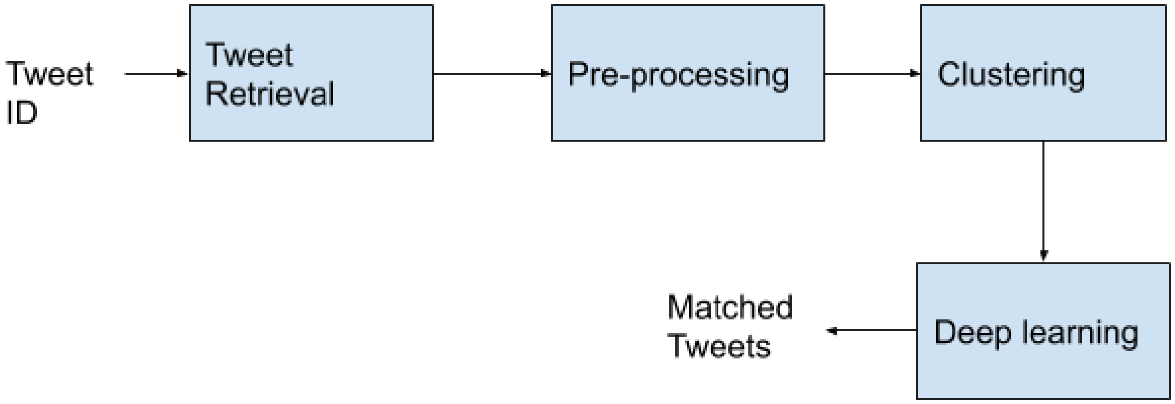


FIGURE 3: STAGES OF PROJECT

The entire system can be divided into four main modules:- tweet retrieval, preprocessing, clustering, deep learning.

1. **Tweet retrieval**
   The tweets are retrieved from the Twitter API by creating an application to generate secret key and consumer key. This provides access to download the tweets.
2. **Pre-processing**
   The retrieved tweets undergo preprocessing step. This removes stopwords, hashtags, URLs, non-ASCII characters. Detailed diagram is given in Figure 3.
3. **Clustering**
   Clustering is an unsupervised approach to machine learning that, given a set of unlabeled data, clusters the data items in groups or clusters such that each group contains items that are similar to each other while items in different groups are dissimilar. We have experimented with the K-means and the DBSCAN algorithm for the problem at hand.

4. **Deep Learning**

   Deep learning can be defined as neural networks with a large number of parameters and layers. Upon taking the pre classified tweets for creating the model, we implement the Recurrent Neural Network in order to classify the tweets into the topical classes, such that output of this process will be served as an input for need-availability matching model.

BLOCK DIAGRAM OF THE PREPROCESSING STAGE

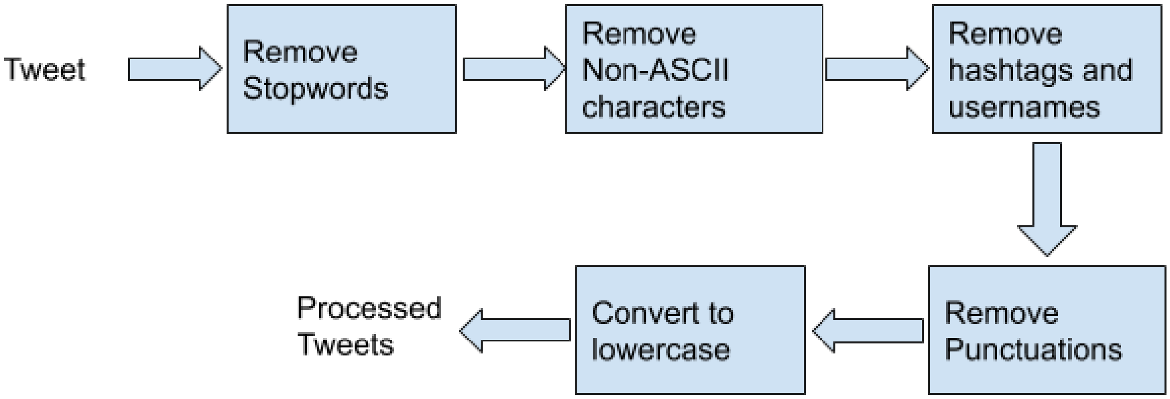


FIGURE 4: PREPROCESSING STAGES

Pre-processing is a crucial step in data mining. We have used NLTK library for text preprocessing.

Text preprocessing operations include:

- Removing all the stopwords (also for Hindi and Nepali).
- Removing non-ASCII characters.
- Remove hashtags and usernames.
- Removal of punctuation.
- Removing URLs.
- Normalizing all the text of tweets into lowercase.
- Removing duplicate tweets and retweets.

DIAGRAM OF THE PROJECT

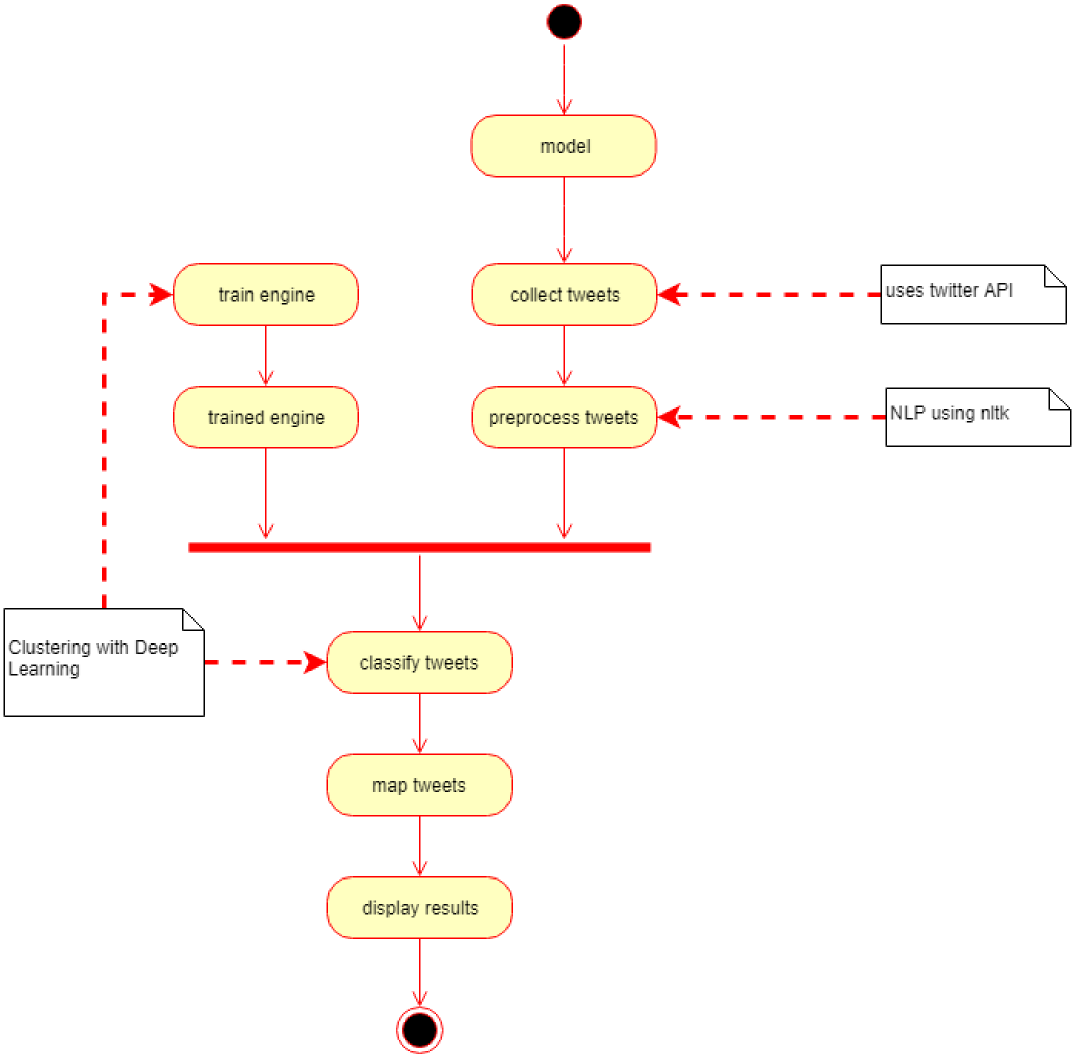


FIGURE 5: PROJECT FLOW

The tweets are collected using the twitter API and stored as text files. These tweets are then given for preprocessing. We use NLP using nltk library to remove hashtags, URLs, usernames, stopwords, punctuations and duplicates. Using clustering and deep learning techniques, engine is trained on the training set of data. This trained engine is then used with clustering with Deep learning techniques to classify the tweets and map them producing the output of the model.

Chapter 5

EXPERIMENTS CONDUCTED

The project entailed 2 main phases; that of separating the tweets into needs and availability and of mapping the required needs to the corresponding availability.

5.1 Doc2vec

The Doc2vec model is used for converting the textual tweets into vector representations. The tweets are translated into feature vectors for processing and analysis. A feature vector is an n-dimensional vector. For the ease of processing, the dimensionality of the vectors has been reduced using the t- distributed stochastic neighbor embedding or the t-SNE dimensionality reduction technique as mentioned in the literature survey above. The tweets are thus represented using the x and the y coordinates in the vector space. The doc2vec model was implemented using Gensim**.** The doc2vec model is implemented using the following code:

```
import gensim

import smart_open

import sklearn.manifold

from sklearn.cluster import KMeans

import numpy as np

from sklearn.cluster import DBSCAN

from sklearn import metrics

import matplotlib

import matplotlib.pyplot as plt

import seaborn as sns
```

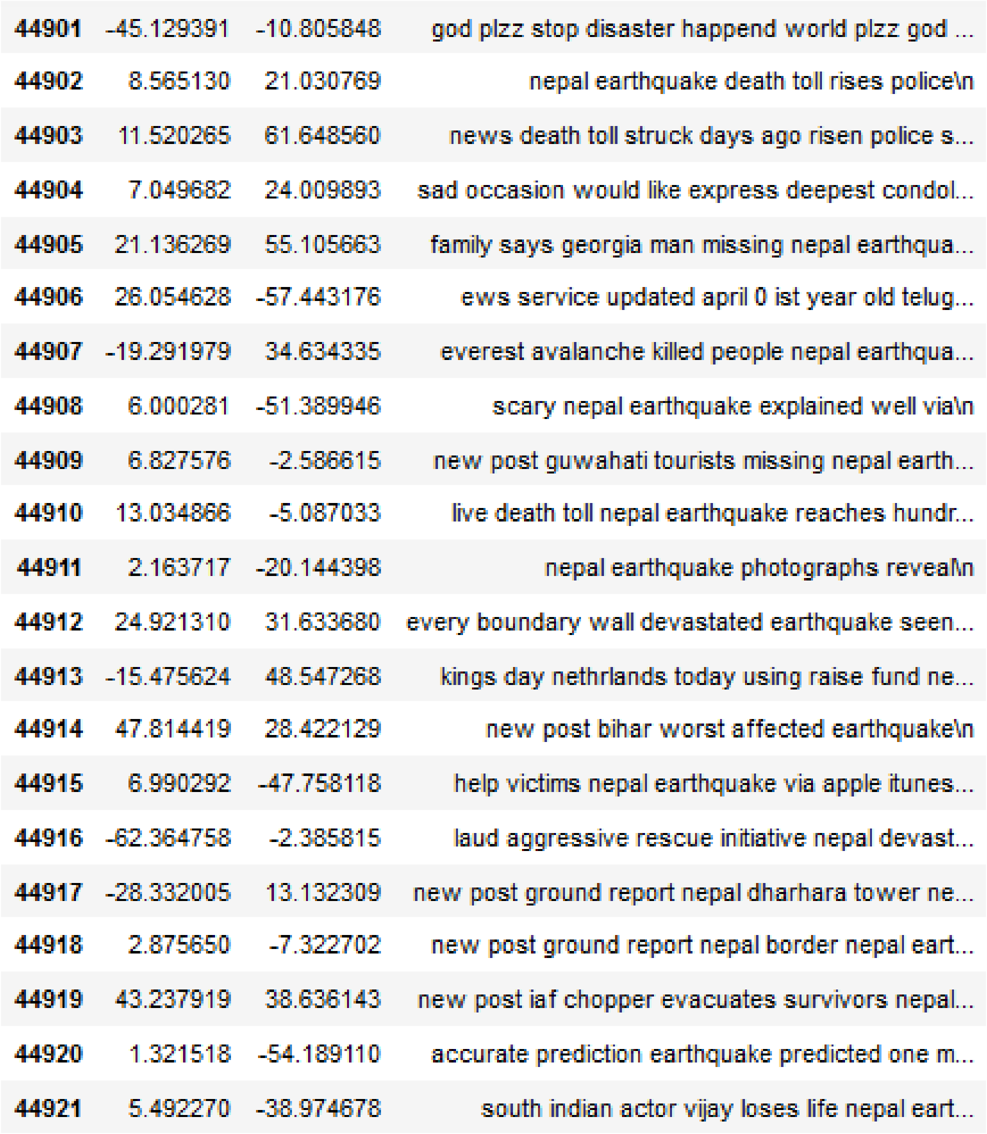
```
From sklearn.preprocessing import StandardScaler

import pandas as pd

filename = 'dataset.txt'

def accept_text(filename):

with smart_open.smart_open(filename, encoding="iso-8859-1") as file:
```

```
index = []

tweet = []

withsmart_open.smart_open(filename, encoding="iso-8859-1") as f:

for i, line in enumerate(f):

    #print(i, ":" , line, "\n")

index.append(i)

tweet.append(line)
```

```
train_corpus = list(read_corpus(filename))

model = gensim.models.doc2vec.Doc2Vec(vector_size=50, min_count=2, epochs=50)

model.build_vocab(train_corpus)

model.train(train_corpus, total_examples=model.corpus_count, epochs=model.epochs)

tsne = sklearn.manifold.TSNE(n_components=2, random_state=0)

tsne_d2v = tsne.fit_transform(model.docvecs.vectors_docs)

tsne_d2v_df = pd.DataFrame(data=tsne_d2v, columns=["x", "y"])

tsne_d2v_df['Tweet'] = tweet

tsne_d2v_df
```

The doc2vec model is trained on 44,921 tweets and the output is the 2 dimensional or 2D vectors of the tweets in the dataset, in the x and y dimensions of the vector space.

Examples are →



	X	Y	Tweet
44901	-45.129391	-10.805848	god plzz stop disaster happend world plzz god ...
44902	8.565130	21.030769	nepal earthquake death toll rises police\n
44903	11.520265	61.648560	news death toll struck days ago risen police s...
44904	7.049682	24.009893	sad occasion would like express deepest condol...
44905	21.136269	55.105663	family says georgia man missing nepal earthqua...
44906	26.054628	-57.443176	ews service updated april 0 ist year old telug...
44907	-19.291979	34.634335	everest avalanche killed people nepal earthqua...
44908	6.000281	-51.389946	scary nepal earthquake explained well via\n
44909	6.827576	-2.586615	new post guwahati tourists missing nepal earth...
44910	13.034866	-5.087033	live death toll nepal earthquake reaches hundr...
44911	2.163717	-20.144398	nepal earthquake photographs reveal\n
44912	24.921310	31.633680	every boundary wall devastated earthquake seen...
44913	-15.475624	48.547268	kings day nethrlands today using raise fund ne...
44914	47.814419	28.422129	new post bihar worst affected earthquake\n
44915	6.990292	-47.758118	help victims nepal earthquake via apple itunes...
44916	-62.364758	-2.385815	laud aggressive rescue initiative nepal devast...
44917	-28.332005	13.132309	new post ground report nepal dharhara tower ne...
44918	2.875650	-7.322702	new post ground report nepal border nepal eart...
44919	43.237919	38.636143	new post iaf chopper evacuates survivors nepal...
44920	1.321518	-54.189110	accurate prediction earthquake predicted one m...
44921	5.492270	-38.974678	south indian actor vijay loses life nepal eart...

FIGURE 6:DATA REPRESENTATION OF TWEETS-2

The spatial layout of the 2 dimensional vectors in the vector space in the x dimension and y dimension, after implementing the doc2vec model, is as shown below:

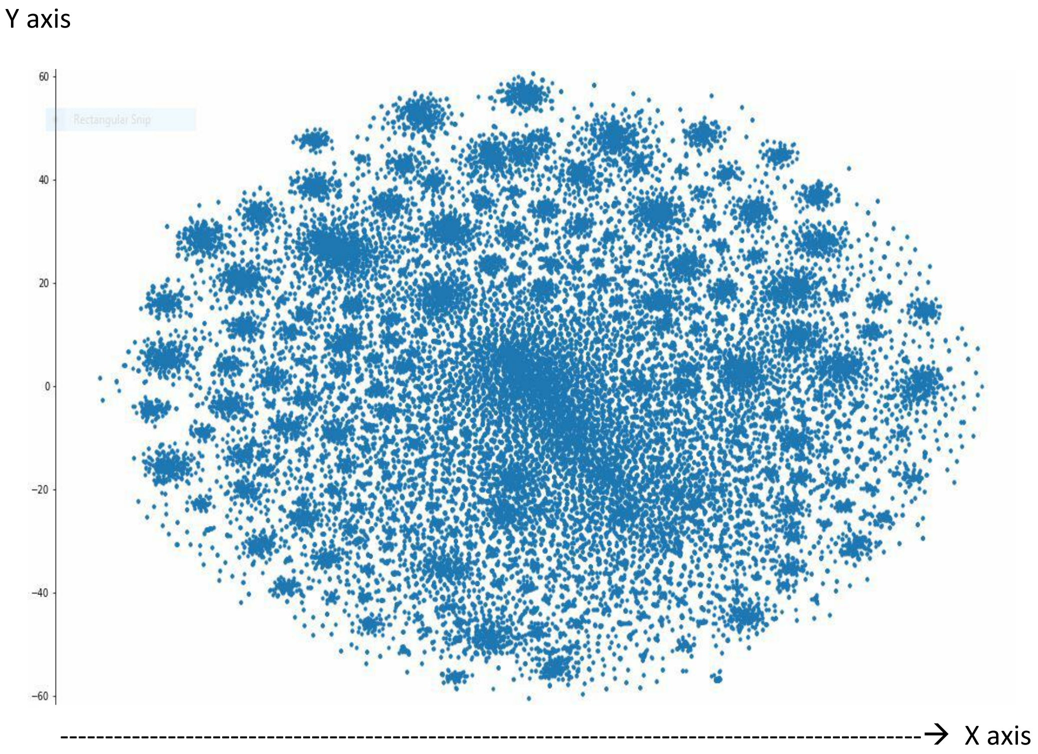


FIGURE 7: VIEW OF THE VECTOR SPACE

Each blue dot is a tweet that has been converted into a term vector by doc2vec. As can be seen from the figure 7 above, there are dense regions present that indicate that points forming those areas are similar and hence the tweets they represent are close in meaning.

5.2 Clustering

5.2.1 K-means

The K-means algorithm implemented with parameter k =2 does not yield a good result because the k-means algorithm works well with data of distinctly separate and spherical shape. The output of the doc2vec model shows that the clusters are not distinct and thus the k-means algorithm does not yield the expected results. The code for the k-means algorithm is as follows:

```
#import the modules

#create the k-means object with the vectors created previously

kmeans = KMeans(n_clusters=2, random_state=0).fit(Y)

fig = plt.figure(figsize=(15, 10))

# plt.rcParams["figure.figsize"] = [15,10]

ax = fig.add_subplot(1, 1, 1)

c = kmeans.labels_

ix = np.where(c==1)

ax.plot(Y[ix,0], Y[ix,1], 'o',markerfacecolor='red', markersize=12)

ix = np.where(c==0)

ax.plot(Y[ix,0], Y[ix,1], 'o',markerfacecolor='green', markersize=12)

plt.show()
```

The following figure depicts clusters formed from the vectors in the vector space of the x dimension and y dimension as shown in doc2vec vector space.

The output of the k-means clustering algorithm is as follows:

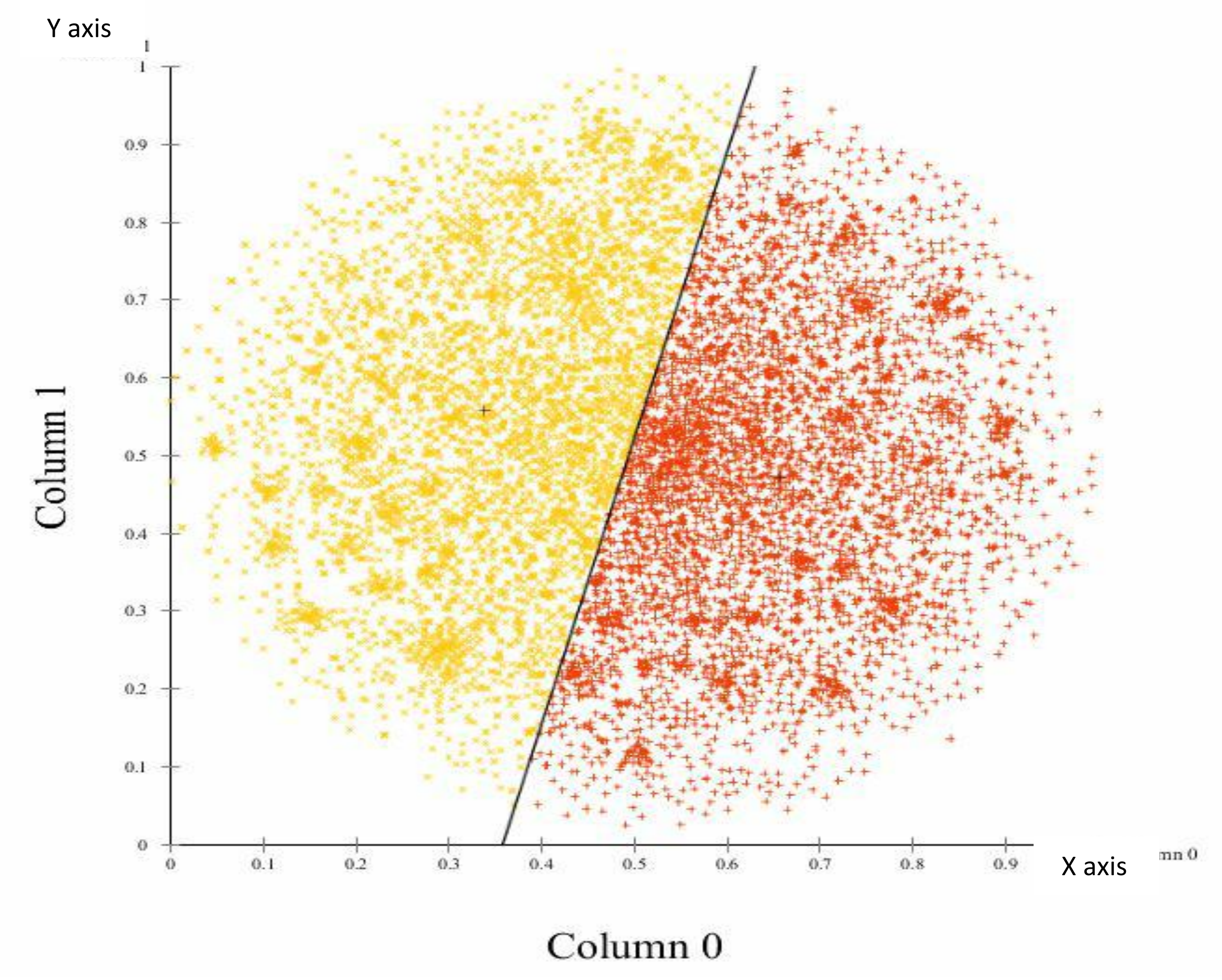


FIGURE 8: K-MEANS CLUSTERING

Each point is a tweet represented as a term vector in the vector space. As can be seen the algorithm simply divides the dataset without forming logical clusters. Thus the k-means is not a good choice for the application at hand.

5.2.2 DBSCAN

The DBSCAN algorithm requires 2 parameters: the Epsilon and Minpts. Using trial and error method we fixed the Epsilon to 0.06 and the Minpts to 26.The code implemented as follows:

```
Y = StandardScaler().fit_transform(Y)

db = DBSCAN(eps=0.06, min_samples=26).fit(Y)

core_samples_mask = np.zeros_like(db.labels_, dtype=bool)

core_samples_mask[db.core_sample_indices_] = True

labels = db.labels_

n_clusters_ = len(set(labels)) - (1 if -1 in labels else 0)

n_noise_ = list(labels).count(-1)

unique_labels = set(labels)

colors = [plt.cm.Spectral(each)

for each in np.linspace(0, 1, len(unique_labels))]

for k, col in zip(unique_labels, colors):

if k == -1:

col = [0, 0, 0, 1]

plt.rcParams["figure.figsize"] = [15,10]

class_member_mask = (labels == k)

xy = Y[class_member_mask&core_samples_mask]

plt.plot(xy[:, 0], xy[:, 1], 'o', markerfacecolor=tuple(col),

markeredgecolor='k', markersize=7)

xy = Y[class_member_mask& ~core_samples_mask]

plt.plot(xy[:, 0], xy[:, 1], 'o', markerfacecolor=tuple(col),

markeredgecolor='k', markersize=5)
```

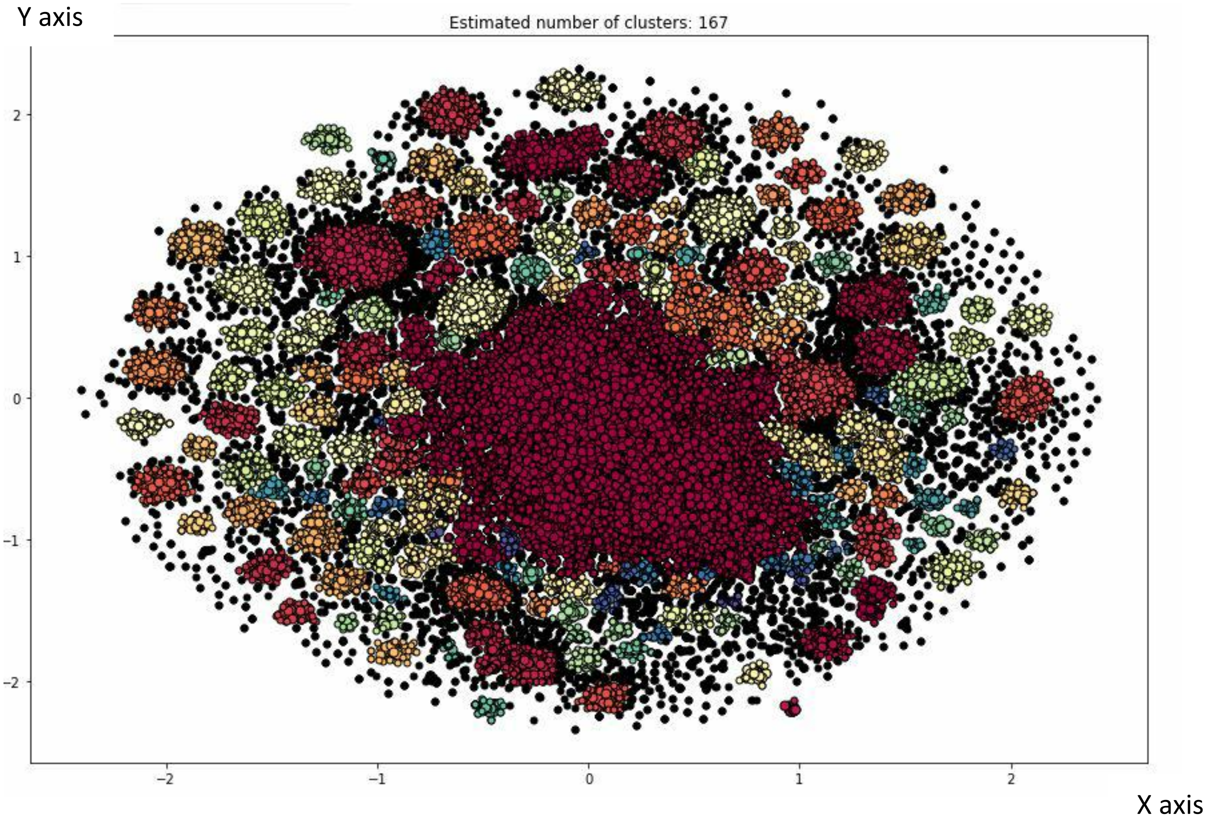


FIGURE 9: CLUSTERS AFTER IMPLEMENTING DBSCAN

The figure above depicts clusters formed from the vectors in the vector space of the x dimension and y dimension as that shown in figure 7 of the doc2vec vector space. The algorithm groups all the points that form dense regions, which can be seen in figure 7, into clusters, as shown in the figure 9 above.

The DBSCAN algorithm does not require the number of clusters expected, as parameter. Rather it takes as parameters, the epsilon and the minpts.

Epsilon (eps): The radial distance from a data point to check the neighbors of that point.

MinPts: The minimum number of points required in the eps neighborhood of a point in order to form a dense region.

Using trial and error method we fixed the Epsilon to 0.06 and the Minpts to 26. In the above visual depicting the output of the DBSCAN algorithm the points on the graph marked in ‘black color’ are the outliers i.e. they have not been included in any cluster.

Only two clusters were expected, that of need class and availability class. However, the output of the DBSCAN algorithm, as can be seen from the figure 9 above, yields 167 heterogeneous clusters and thus is incompliant with the expected result of that of only 2 clusters. Due to the nature of the twitter data, it was observed after manual annotation that there is a large amount of noise making the data heterogeneous in nature. There are a total of 44,921 tweets that have been grouped into 167 clusters instead of only 2 clusters of need and availability as required. On scrutinizing each cluster manually, we observed that all the clusters formed, contained arbitrary tweets and lacked a common characteristic. The groups contain need tweets, availability tweets and a large amount of noise.

From our manual annotation, we found only 207 need tweets and 374 availability tweets. The number of need and availability tweets is extremely less for any clustering algorithm to segregate the total dataset into need tweets and availability tweets. Hence, due to the insufficiency of significant data, clustering techniques for the purpose of segregating the data into need tweets and availability tweets performed less than optimum and yielded results, lower than expected.

5.3 Deep learning

Following is the example along with the snippet of code of how the above process can be implemented:

Import data:

```
data = readtable("training.xlsx");
data.classifier = categorical(data.classifier);
```

Class distribution:

```
f = figure;
f.Position(3) = 1.5*f.Position(3);
h = histogram(data.classifier);
xlabel("Class")
ylabel("Frequency")
title("Class Distribution")
```

The following figure shows about the concentration of tweets of the classified dataset with respect to the 2 classifiers viz, Availability and Need.

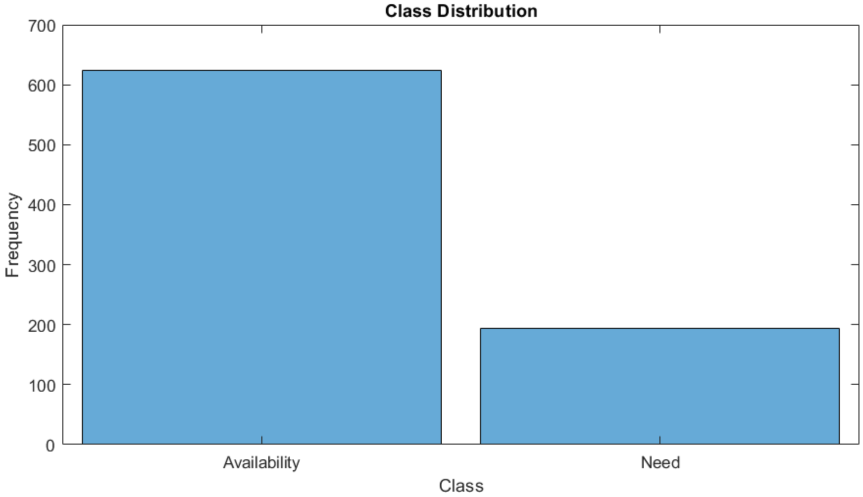


FIGURE 10:CLASS DISTRIBUTION

Word cloud representation:

```
textDataValidation =
dataValidation.tweet;
YValidation =
dataValidation.classifier;

figure
wordcloud(textDataTrain);
title("Training Data");
```

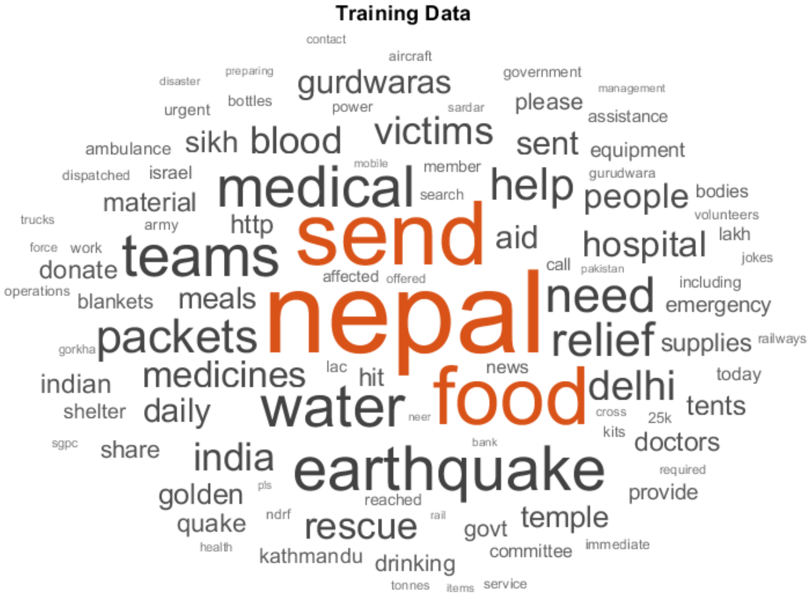


FIGURE 11:WORD CLOUD

This word cloud representation shows the frequency of words used throughout the dataset, e.g ‘nepal’ has the highest frequency followed by ‘food’.

Preprocess data (word encoding):

```
1)cvp = cvpartition(data.classifier,'Holdout',0.30);
  c=cvpartition(data.classifier,'KFold',10);
dataTrain = data(training(cvp),:);
dataValidation = data(test(cvp),:);

2)textDataTrain = dataTrain.tweet;
YTrain = dataTrain.classifier;
textDataTrain = lower(textDataTrain);
documentsTrain = tokenizedDocument(textDataTrain);
documentsTrain = erasePunctuation(documentsTrain);
textDataValidation = lower(textDataValidation);
documentsValidation=
tokenizedDocument(textDataValidation);
documentsValidation =
erasePunctuation(documentsValidation);

3)encTrain = wordEncoding(documentsTrain);
encValidation = wordEncoding(documentsValidation);
tweetLengths = doclength(documentsTrain);
```

1st block: Partition the dataset for, training and validation purpose of the model.

2nd block: Convert the alphabets to lowercase, tokenize each tweet and erase punctuation of both, training and validation dataset.

3rd block: converts the words from the training dataset to indices.

Histogram of Tweet Lengths:

```
figure
histogram(tweetLengths)
title("Tweet Lengths")
xlabel("Length")
ylabel("Number of tweets")
```

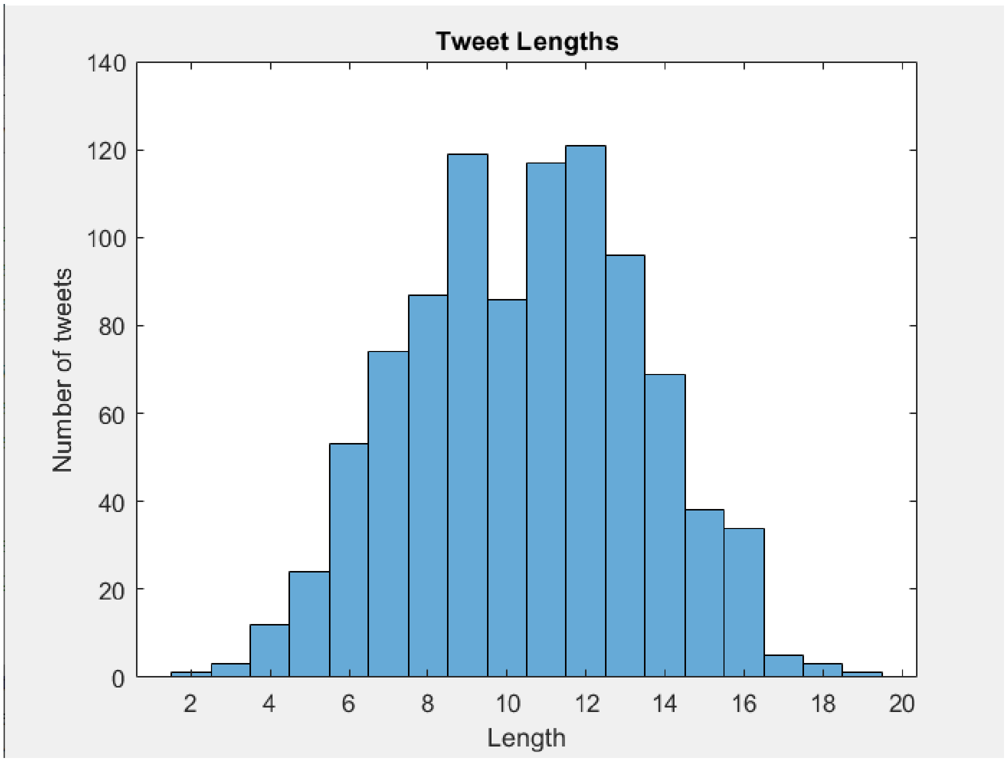
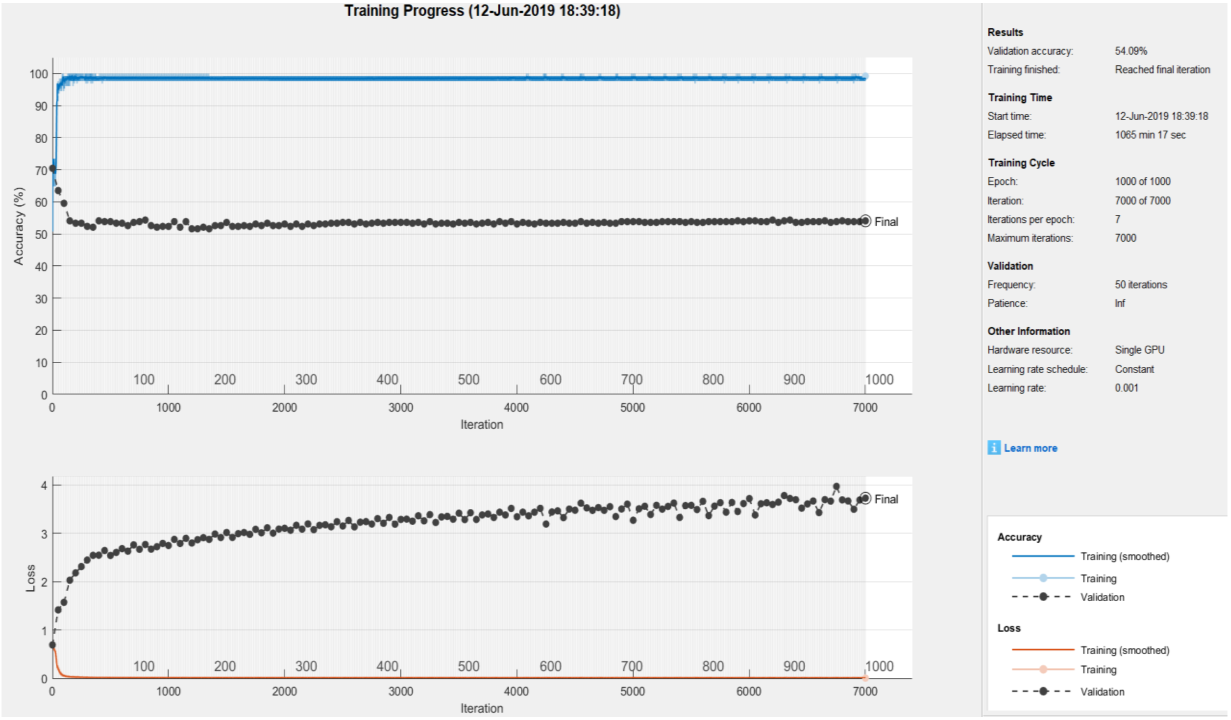


FIGURE 12:TWEET LENGTHS

Following histogram depicts the number of tweets having a certain length.

Create and Train LSTM Network:

```
layers = [ ...
sequenceInputLayer(inputSize)
wordEmbeddingLayer(embeddingDimension,numHiddenUnits)
lstmLayer(hiddenSize,'OutputMode','last')
fullyConnectedLayer(numClasses)
softmaxLayer
classificationLayer];

options = trainingOptions('adam', ...
'MaxEpochs',1000, ...
'GradientThreshold',1, ...
'InitialLearnRate',0.001, ...
'ValidationData',{XValidation,YValidation}, ...
'Plots','training-progress', ...
'Verbose',false);
net = trainNetwork(XTrain,YTrain,layers,options);
```



```
Results
Validation accuracy:        54.09%
Training finished:          Reached final iteration

Training Time
Start time:                 12-Jun-2019 18:39:18
Elapsed time:               1065 min 17 sec

Training Cycle
Epoch:                      1000 of 1000
Iteration:                  7000 of 7000
Iterations per epoch:       7
Maximum iterations:         7000

Validation
Frequency:                  50 iterations
Patience:                   Inf

Other Information
Hardware resource:          Single GPU
Learning rate schedule:     Constant
Learning rate:              0.001
```

FIGURE 13:SNAPSHOT OF ONE OF THE TRAINING PROGRESS

This step trains the dataset using LSTM algorithm having 6 layers of deep learning and validating the model using the validation dataset at the same time. The model was tested for different optimizers – sgdm, rmsprop, adam to obtain the best model out of all, the results obtained have been discussed in the next section.

Output of the experiment conducted is a model of RNN which is capable of classifying any tweets given to it as ‘Need’ and ‘Availability’.

Chapter 6

RESULTS AND DISCUSSION

Prior work:

Prior works include pattern matching methods [24] [25] and also word embedding based retrieval techniques [26] to capture the semantics of need and availability tweets to retrieve the tweets. For this purpose a dataset of 50,068 tweets pertaining to the Nepal earthquake was used.

The study revealed that pattern matching methods could not identify the required need and availability tweets because the tweets seldom contained intuitively complementary terms such as “need”, “availability” or “distribute”.

On the other hand, the contextual word2vec based methods successfully identified need tweets and availability tweets. Thus, the performance of word2vec based methods is much superior compared to pattern matching method establishing the efficiency of contextual matching. The accuracy was found to be 18% and 45% for need and availability respectively.

Evaluation Measures [29]:

It is extremely important for the purpose of accurate evaluation of measures to
i) Identify as many of the need tweets and availability tweets as possible
ii) Also to precisely identify the need tweets and availability tweets
The evaluation measures primarily used were:

1) Precision:
Fraction of retrieved need (availability) tweets that are genuine and authentic (true) need (availability) tweets to the total number of retrieved tweets identified as need (availability) tweets.

2) Recall:
Fraction of retrieved need (availability) tweets that are genuine and authentic (true) need (availability) tweets to the total number of need (availability) tweets.

3) F-Score:

This is the harmonic mean of precision and recall. Since this method incorporates both precision and recall, it is used for comparison of the various methodologies.

The results of the prior work taken from [26] are as shown below

Ranking Model	**Expansion**	**Precision**	**Recall**	**F-score**
Need-tweets				
Indri	None	0.1010	0.2686	0.1468
Word2vec	Word2vec	**0.1880**	**0.5000**	**0.2732**
Availabilty-tweets				
Indri	None	0.3410	0.3317	0.3363
Word2vec	Rocchio	**0.4930**	**0.4796**	**0.4862**

TABLE 9:RESULT OF PRIOR WORK

<u>Present work</u>:

Our study is based on the dataset of the 2015 Nepal Earthquake. Out of the total 66,000 tweets, 44,921 tweets were retrieved successfully for study.

We implement the K-means algorithm and the DBSCAN algorithm of clustering on the dataset. The dataset was required to be divided into 2 classes for segregation, namely the need class and the availability class. To implement the k-means, the required parameter of the number of clusters which is 2, was fed and 2 clusters were obtained. However, it was observed that the k means algorithm divided the 44,921 vectors into 2 specious clusters with both clusters having both need and availability tweets and also noise. So any visible logical base in the formation of the clusters was missing. Instead of dividing the entire dataset of 44,921 vectors into 2 distinct groups of need cluster and availability cluster, the k means algorithm was dividing the 44,921 vectors into 2 mixed groups, each having need, availability and noise altogether. Thus it was inferred that the k-means algorithm fails to give the expected result and therefore is unsuitable for this application. This was due to the varying density in the vector space and also because the clusters were not discrete.

The k-means algorithm works well only when the vectors in the vector space are distanced such that the clusters formed by these vectors are spherical and sufficiently separated from each other. The k-means algorithm is less known work on regions of varying density. So the DBSCAN algorithm was carried out.

On implementing the DBSCAN algorithm on 44,921 vectors we obtained 13,091 outliers. After removing the 13,091 outliers, the remaining 31,830 vectors were clustered by the DBSCAN algorithm into 167 groups instead of only 2 required clusters of need class and availability class. These 167 groups displayed no specific bonding characteristics of need or availability. Rather they were simply a motley cluster of need tweets, availability tweets and also a large amount of noise with no particular outcome.

The reason being, that the DBSCAN clustering algorithm is useful for identifying multiple clusters in vector spaces having arbitrary shapes. It works optimally when the data is homogenous. However, our dataset was heterogeneous in nature. Being social media data, the dataset contained a large amount of noise such as condolences messages, rumors and other irrelevant data etc. The DBSCAN algorithm therefore gave low results.

We even manually annotated 31,830 tweets obtained after removing the 13,091 outliers from the dataset consisting of 44,921 tweets. From our annotation, we discovered that, of the total 31,830 tweets, only 207 tweets were need tweets and just 374 tweets were availability tweets while all other tweets were noise.

The performance of the DBSCAN algorithm was as follows:

	Precision	Recall	F-score
Need	0.2217	0.0965	0.1781
Availability	0.1977	0.0796	0.1476

TABLE 10: RESULT OF DBSCAN

On comparison, the precision, recall and f-score of the prior work with the present work,it is seen that, for need, the precision using word2vec was 0.1880, whereas using DBSCAN it is 0.2217. The recall using word2vec was 0.5000 whereas with DBSCAN it is 0.0965. The f-score for the prior work was 0.2732 while we obtained 0.1781.

For availability, the precision using word2vec was 0.4930, whereas using DBSCAN it is 0.1977. The recall using word2vec was 0.4796 whereas with DBSCAN it is 0.0796. The f-score for the prior work was 0.4862 while we obtained 0.1476.

The F-score is the measure takes into consideration the essence of both precision and recall. Therefore, the F-score is used for comparing the results.

It is seen that the F-score is lower for DBSCAN than for word2vec for both need and availability.

Thus, it was observed that the word embedding based techniques for segregating the tweets into need tweets and availability tweets performed better than the DBSCAN algorithm.

So, we noted that the clustering techniques for the purpose of segregating and grouping the tweets into need tweets and availability tweets are often liable to be incoherent with the results expected. The traditional unsupervised methods of clustering technique were hence intermitted to explore deep learning techniques.

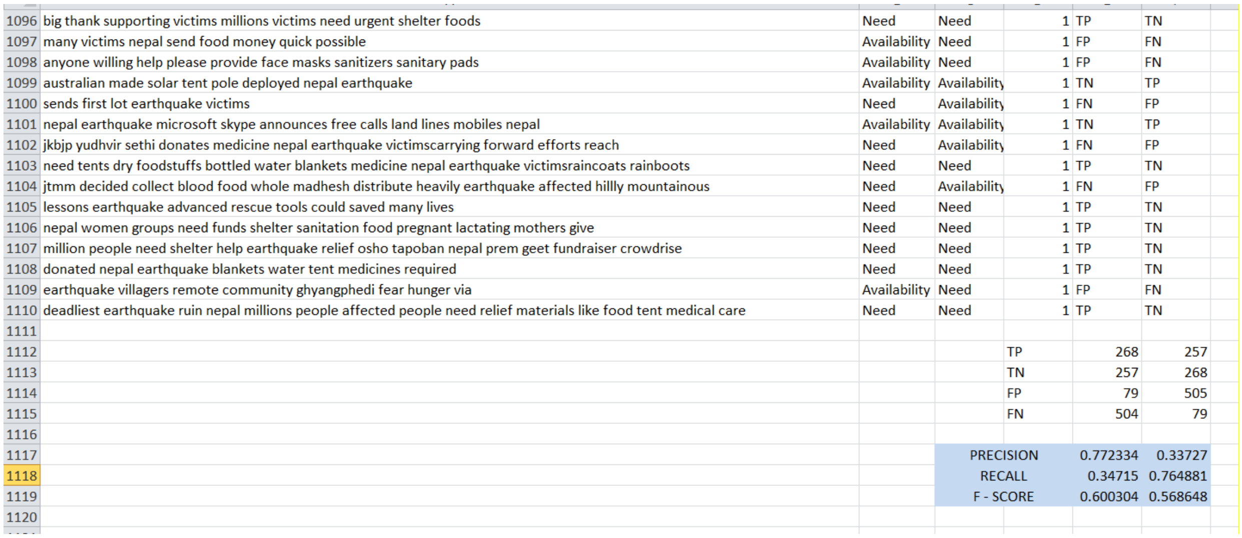
Hence deep learning techniques were also studied.

Validation of Testing and Training data:

```
reportsNew = readtable("training.xlsx");
YTestValidation = reportsNew.classifier;
reportsNew = lower(reportsNew.tweet);
documentsNew = tokenizedDocument(reportsNew);
documentsNew = erasePunctuation(documentsNew);
encTest = wordEncoding(documentsNew);
XNew = doc2sequence(encTest,documentsNew,'Length',19);
[labelsNew,score] = classify(net,XNew);
%to be typed in command window
T=table(reportsNew,string(labelsNew));
testAccuracy =
sum(string(labelsNew)==YTestValidation)/numel(string(labe
lsNew));
writetable(T,"Tested.xlsx");
```

Optimizer	**Precision**	**Recall**	**F-Score**
Need-Tweets			
Adam	**0.77234**	**0.34715**	**0.600304**
Sgdm	0	N.D	N.D
Rmsprop	0.149856	0.253659	0.30677
Availability-tweets			
Adam	**0.33727**	**0.764881**	**0.568648**
Sgdm	1	N.D	N.D
Rmsprop	0.799213	0.674419	0.1588586

Table 11: Result Of Deep Learning



1096	big thank supporting victims millions victims need urgent shelter foods	Need	Need	1	TP	TN
1097	many victims nepal send food money quick possible	Availability	Need	1	FP	FN
1098	anyone willing help please provide face masks sanitizers sanitary pads	Availability	Need	1	FP	FN
1099	australian made solar tent pole deployed nepal earthquake	Availability	Availability	1	TN	TP
1100	sends first lot earthquake victims	Need	Availability	1	FN	FP
1101	nepal earthquake microsoft skype announces free calls land lines mobiles nepal	Availability	Availability	1	TN	TP
1102	jkbjp yudhvir sethi donates medicine nepal earthquake victimscarrying forward efforts reach	Need	Availability	1	FN	FP
1103	need tents dry foodstuffs bottled water blankets medicine nepal earthquake victimsraincoats rainboots	Need	Need	1	TP	TN
1104	jtmm decided collect blood food whole madhesh distribute heavily earthquake affected hillly mountainous	Need	Availability	1	FN	FP
1105	lessons earthquake advanced rescue tools could saved many lives	Need	Need	1	TP	TN
1106	nepal women groups need funds shelter sanitation food pregnant lactating mothers give	Need	Need	1	TP	TN
1107	million people need shelter help earthquake relief osho tapoban nepal prem geet fundraiser crowdrise	Need	Need	1	TP	TN
1108	donated nepal earthquake blankets water tent medicines required	Need	Need	1	TP	TN
1109	earthquake villagers remote community ghyangphedi fear hunger via	Availability	Need	1	FP	FN
1110	deadliest earthquake ruin nepal millions people affected people need relief materials like food tent medical care	Need	Need	1	TP	TN
1111						
1112				TP	268	257
1113				TN	257	268
1114				FP	79	505
1115				FN	504	79
1116						
1117			PRECISION		0.772334	0.33727
1118			RECALL		0.34715	0.764881
1119			F - SCORE		0.600304	0.568648
1120						

FIGURE 14:SNAPSHOT OF RESULT

Here we discuss about the validation of the trained model against training as well as testing data. As can been seen from the Table 9, the highest **Precision** and **F-score** was obtained by using the **adam** optimizer, for both Need as well as Availability tweets, followed by the **rmsprop** optimizer. It is not surprising for some of the results of the **sgdm** optimizer to be not defined, since the Training data had more of Availability tweets as compared to Need, hence all the tweets were classified to be availability.

Chapter 7

CONCLUSION

The major objective of our project was to appraise the machine learning techniques for separating tweets extracted during an on-going disaster, in this case the 2015 Nepal Earthquake, into need-tweets and availability tweets and then to connect them.

The prior works used pattern matching and word embedding techniques to achieve the above. We, on the other hand employ the clustering and the deep learning techniques for this purpose.

The steps taken to accomplish the required tasks were as follows:

a) Extracting thousands of tweets that were relevant to the disaster.
b) Preprocessing the tweets to remove emoticons, abbreviations etc.
c) Transforming the tweets into vector representation.
d) Implementing clustering techniques to cluster into need tweets and availability tweets.
e) Deep learning.

Clustering algorithms such as the k-means algorithm and the DBSCAN algorithm were implemented on the dataset of 44921 tweets.

It was observed that the clusters produced were incoherent with the results expected as indicated by the low f-score. It was then concluded that the performance of unsupervised learning technique of clustering for the purpose of segregating and grouping the tweets into need tweets and availability tweets is marginal as compared to that of word embedding based techniques, as used in the prior work. Thus the clustering techniques lend themselves less than optimal than previously thought, for the purpose of effective exploitation for grouping the tweets into need tweets and availability tweets. Low F-score of DBSCAN clustering algorithm indicates possibility for improvement and the need to further explore the techniques.

So then, deep learning techniques to classify the dataset were used. And as can be seen in the results mention above it can be concluded that the model was able to classify the data to need and availability with a higher Precision and higher F-score than that found in prior work done.

Due to the size of the data set and the dissimilarity in the number of tweets with respect to need and availability the results were restricted to that one obtained. If there had been more training data, then the accuracy of the model would have increased drastically.

Future Work

As part of future work, we intend to map the need tweets to that of availability tweets, so as to find the exact resource for the victims in need. This can be done using the same approach used above for deep learning i.e. by using Manhattan LSTM, which will decide whether any 2 given sentences are same or not, it may not predict about the next word but may arrive at a conclusion that amongst a given set of sentences which one is the most suitable match for the input tweet. With respect to our domain, for a given need tweet, this DNN(RNN) will find among the given availability tweet, which one matches the most to the need tweet mentioned above.

REFERENCES

[1]https://www.bgr.in/news/kerala-floods-twitter-2-62-million-tweets-during-august-2018-deluge

[2]http://www.pewresearch.org/fact-tank/2013/10/28/twitter-served-as-a-lifeline-of-information-during-hurricane-sandy

[3]Cluster Analysis of Twitter Data: A Review of Algorithms NoufaAlnajran, Keeley Crockett, David McLean and Annabel Latham

[4]M.Basu, S. Ghosh, K. Ghosh, M. Choudhury, "Overview of the FIRE 2017 track: Information Retrieval from Microblogs during Disasters (IRMiDis)", Forum of Information Retrieval Evaluation (CEUR Workshop Proceedings), CEUR.WS.org, 2017.

[5] N. Oostdijk, A. Hürriyetoğlu, "Detecting the Need for Resources and their Availability", Forum of Information Retrieval Evaluation (CEUR Workshop Proceedings), CEUR.WS.org, 2017.

[6] A. Talukdar, R.l Bhargava, Y. Sharma, "BITS_PILANI@IMRiDis-FIRE 2017: Information Retrieval from Microblogs during Disasters", Forum of Information Retrieval Evaluation (CEUR Workshop Proceedings), CEUR.WS.org, 2017.

[7] N. Kumar, M. Dubey, "DataBros@Information Retrieval from Microblogs during Disasters(IRMiDis)", Forum of Information Retrieval Evaluation (CEUR Workshop Proceedings), CEUR.WS.org, 2017.

[8] S. Baweja, A. Aggarwal, V. Goyal, S. Mehta, "Automatic Retrieval of Actionable Information from Disaster-related Microblogs", Forum of Information Retrieval Evaluation (CEUR Workshop Proceedings), CEUR.WS.org, 2017.

[9] B. Gautam, A. Basava, "Automatic Identification and Ranking of Emergency Aids in Social Media Macro Community", Forum of Information Retrieval Evaluation (CEUR Workshop Proceedings), CEUR.WS.org, 2017.

[10] M. Raf, S. Ahmed, F. Ahmed, Fawzan Ahmed, “Tweet Classification using Semantic Word-Embedding with Logistic Regression”, Forum of Information Retrieval Evaluation (CEUR Workshop Proceedings), CEUR.WS.org, 2017.

[11] Z. Zicheng, N. Hui, Z. Ziyao, Z. Jinmei, L. Jun, “HLJIT2017-IRMIDIS@IRMiDis-FIRE2017: Information Retrieval from Microblogs during Disasters”, Forum of Information Retrieval Evaluation (CEUR Workshop Proceedings), CEUR.WS.org, 2017.

[12] D. Xin, W. Xiaoyu, Z. Ziyao, Q. Limin, “A Hybrid Model For Information Retrieval From Microblogs During Disaster”, Forum of Information Retrieval Evaluation (CEUR Workshop Proceedings), CEUR.WS.org, 2017.

[13] H. Mehrotra, R. Soni, S. Pal, “IIT BHU at FIRE 2017 IRMiDis Track - Fully Automatic Approaches to Information Retrieval”, Forum of Information Retrieval Evaluation (CEUR Workshop Proceedings), CEUR.WS.org, 2017.

[14] N. Fadaei, T. Mandl, “Use of the Pole-based Overlapping Clustering Method for Labeling Tweets @ IRMiDis FIRE 2017”, Forum of Information Retrieval Evaluation (CEUR Workshop Proceedings), CEUR.WS.org, 2017.

[15] X. Yang, C. Macdonald, I. Ounis, “Using Word Embedding in Twitter Election Classification”, https://arxiv.org/abs/1606.07006.

[16] T. Nazer, F. Morstatter, H. Dani, H. Liu, “Finding Requests in Social Media for Disaster Relief”, IEEE/ACM ASSONAM, 2016.

[17] D. T. Nguyen, S. Joty, M. Imran, H. Sajjad, P. Mitra, “Application of Online Deep Learning for Crisis Response Using Social Media Information”, https://arxiv.org/abs/1610.01030.

[18] G. Cleuziou, L. Martin, C. Vrain, “PoBOC: an Overlapping Clustering Algorithm (Application to Rule-Based Classification and Textual Data)”, http://www.univ-orleans.fr/lifo/Members/cleuziou/papers/CMV_ECAI_04.pdf.

[19] A. K. Jain, M. N. Murty, P. J. Flynn, “Data Clustering:A Review”, ACM Comput Surv.31,3,264–323, Sept 1999, https://doi.org/10.1145/331499.331504

[20]N. alnajran, K. Crockett, D. Mclean, A. Latham, “Cluster Analysis of Twitter Data: A Review of Algorithms”, 9th International Conference on Agents and Artificial Intelligence (ICAART), 24 February 2017 - 26 February 2017.

[21]https://www.kdnuggets.com/2016/09/slangsd-sentiment-dictionary-slang-words.html

[22]KoustavRudra, Subham Ghosh, NiloyGanguly, Pawan Goyal, Saptarshi Ghosh, “Extracting Situational Information from Microblogsduring Disaster Events:a Classification-Summarization Approach” , CIKM, pp. 583-592, 2015

[23]Anirban Sen, KoustavRudra, Saptarshi Ghosh, “Extracting Situational Awareness from Microblogs during Disaster Events”, COMSNET-2015

[24]I. Temnikova, C. Castillo, and S. Vieweg, “EMTerms 1.0: A terminological Resource for Crisis Tweets,” in *Proc. ISCRAM*, 2015

[25]H. Purohit, C. Castillo, F. Diaz, A. Sheth, and P.Meier, “Emergency-relief coordination on social media: Automatically matching resource requests and offers,” *First Monday*, vol. 19, no. 1,Jan 2014

[26]MoumitaBasu, Kripabandhu Ghosh, Somenath Das, RatnadeepDey, SomprakashBandyopadhyay, and Saptarshi Ghosh. 2017. Identifying Post-Disaster Resource Needs and Availabilities from Microblogs. In *Proc. ASONAM*.

[27]NoushinFadaei, Thomas Mandl2017 .Use of the Pole-based Overlapping Clustering Method for Labeling Tweets @ *IRMiDis FIRE 2017*

[28] H. Purohit, C. Castillo, F. Diaz, A. Sheth, and P. Meier, “Emergency-relief coordination on social media: Automatically matching resource requests and offers,” First Monday, vol. 19, no. 1, Jan 2014.

[29] https://en.wikipedia.org/wiki/Precision_and_recall#F-measure