VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY
Department of Computer Engineering

Project Report on

Analysis of Twitter Reactions to Government Policies

In partial fulfillment of the Fourth Year, Bachelor of Engineering (B.E.) Degree in Computer Engineering at the University of Mumbai Academic Year 2017-2018

Submitted by
Anagha Karmarkar (D17 - A , Roll no -38)
Vinit Pawar (D17 - A , Roll no - 60)
Mansi Shivani (D17 - A , Roll no - 72)
Kanchan Tewani (D17 - A , Roll no - 74)

Project Mentor
Mrs. Vidya Zope

(2017-18)

VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY
Department of Computer Engineering

Certificate

This is to certify that ***Anagha Karmarkar, Vinit Pawar, Mansi Shivani and Kanchan Tewani*** of Fourth Year Computer Engineering studying under the University of Mumbai have satisfactorily completed the project on "***Analysis of Twitter Reactions to Government Policies***" as a part of their coursework of PROJECT-II for Semester-VIII under the guidance of their mentor ***Mrs. Vidya Zope*** in the year 2017-2018 .

This thesis/dissertation/project report entitled "***Analysis of Twitter Reactions to Government Policies***" by ***Anagha Karmarkar, Vinit Pawar, Mansi Shivani and Kanchan Tewani*** is approved for the degree of ***Bachelor of Engineering (B.E.) Degree in Computer Engineering at Mumbai University Academic Year 2017-18.***

Programme Outcomes	Grade
PO1,PO2,PO3,PO4,PO5,PO6,PO7, PO8, PO9, PO10, PO11, PO12 PSO1, PSO2	

Date: Project Guide

Project Report Approval
For
B. E (Computer Engineering)

This thesis/dissertation/project report entitled ***Analysis of Twitter Reactions to Government Policies*** by ***Anagha Karmarkar, Vinit Pawar, Mansi Shivani and Kanchan Tewani*** is approved for the degree of ***Bachelor of Engineering (B.E.) Degree in Computer Engineering at Mumbai University Academic Year 2017-18.***

Internal Examiner

External Examiner

Head of the Department

Principal

Date:
Place:

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

---	---
(Signature)	(Signature)
---	---
(Anagha Karmarkar D17A-38.)	(Vinit Pawar D17A-60)
---	---
(Signature)	(Signature)
---	---
(Mansi Shivani D17A-72)	(Kanchan Tewani D17A-74)

Date:

ACKNOWLEDGEMENT

We are thankful to our college Vivekanand Education Society's Institute of Technology for considering our project and extending help at all stages needed during our work of collecting information regarding the project.

It gives us immense pleasure to express our deep and sincere gratitude to Assistant Professor **Mrs. Vidya Zope** (Project Guide) for her kind help and valuable advice during the development of project synopsis and for her guidance and suggestions.

We are deeply indebted to Head of the Computer Department **Dr.(Mrs.) Nupur Giri** and our Principal **Dr. (Mrs.) J.M. Nair ,** for giving us this valuable opportunity to do this project.

We express our hearty thanks to them for their assistance without which it would have been difficult in finishing this project synopsis and project review successfully.

We convey our deep sense of gratitude to all teaching and non-teaching staff for their constant encouragement, support and selfless help throughout the project work. It is great pleasure to acknowledge the help and suggestion, which we received from the Department of Computer Engineering.

We wish to express our profound thanks to all those who helped us in gathering information about the project. Our families too have provided moral support and encouragement at several times.

Computer Engineering Department
COURSE OUTCOMES FOR B.E PROJECT

Learners will be to,

Course Outcome	Description of the Course Outcome
CO 1	Able to apply the relevant engineering concepts, knowledge and skills towards the project.
CO2	Able to identify, formulate and interpret the various relevant research papers and to determine the problem.
CO 3	Able to apply the engineering concepts towards designing solution for the problem.
CO 4	Able to interpret the data and datasets to be utilized.
CO 5	Able to create, select and apply appropriate technologies, techniques, resources and tools for the project.
CO 6	Able to apply ethical, professional policies and principles towards societal, environmental, safety and cultural benefit.
CO 7	Able to function effectively as an individual, and as a member of a team, allocating roles with clear lines of responsibility and accountability.
CO 8	Able to write effective reports, design documents and make effective presentations.
CO 9	Able to apply engineering and management principles to the project as a team member.
CO 10	Able to apply the project domain knowledge to sharpen one's competency.
CO 11	Able to develop professional, presentational, balanced and structured approach towards project development.
CO 12	Able to adopt skills, languages, environment and platforms for creating innovative solutions for the project.

Abstract

Twitter is one of the worlds most popular microblogging site where people openly express their opinions about various topics across the spectrum.This project aims to create a system which can analyze Twitter reactions to various previous and current government policies. The system aims to pinpoint how a particular policy has fared across factors based on the tweets regarding that particular policy. The final goal is to determine the way a policy was received by the general public and the sentiment that exists regarding it amongst the people.

INDEX

Chapter 1 : Introduction

1.1 Introduction to the project

Our project aims to create a system which can analyze Twitter reactions to various previous and current government policies. The system aims to pin point how a particular policy has fared across people of different age groups, genders, localities etc. based on the tweets regarding that particular policy. The final goal is to determine the way a policy was received by the general public and the sentiment that exists regarding it amongst the people

1.2 Motivation

Elections are undoubtedly the biggest test for a political party in a democracy. Campaigns involve a lot of planning, identifying vote banks, addressing community specific issues and more. The advent of Data analytics has revolutionized election planning throughout the country.For a sitting government, the deciding factor going into the next campaign is how their policies fared in the previous tenure. Hence, it is becoming increasingly important to analyze how well a policy did across genders, age groups, localities etc. Traditionally, the only way to tap public opinion was grass root level surveys. These were expensive, time consuming and limited in terms of dataset size. Currently, no such system specifically for Indian government policies exists in the public domain.

1.3 Problem Definition

Twitter is a huge platform with an immensely large and diverse user base. It is one of the most successful social media sites and a leader in reflecting opinions about current events. People put forth their unbiased honest opinions with the advantage of anonymity at times on varied topics of which politics is a leader.

People are quick to react to actions of the government, their policies, amendments or agenda on the microblogging platform. Due to its exclusive character limit, reactions are short and concise.

Analysis of such unbiased public opinion regarding a government policy can turn out to be priceless for a sitting government, the opposition parties, private think tanks and major media houses. It can help paint an extremely accurate picture of the trends and the repercussions of a particular policy which can help in election planning.

1.4 Relevance of the project

The 2014 general elections in India saw exorbitant use of think tanks and data analytics and it resulted in a sweeping majority. It was maybe the first instance of technology being used at this extent in Indian politics. This sparked the idea of having a system which can map public reactions to a government policy which might help in evaluating government performance or planning future campaigns.
The objective of this project is to allow government and private think tanks to observe how a policy fared on the basis of the public reaction to it on Twitter.
The reactions will be further rated on a scale to determine how positive or negative they were based on multiple parameters.

1.5 Methodology Used

Machine Learning Approach will be used to perform sentiment analysis. As no labelled dataset is available in the selected domain as of the date of preparation of this document, we will be creating a manually labelled data set.

- **Data Collection:**
  Scraping tweets using hashtags
- **Data Preprocessing:**
  - Separating actual tweets from twitter metadata
  - Removing emoticons, URLs, hashtags
  - Lemmatization (only for SVM)

- **Feature selection:**

  Bigrams (for SVM)
- **Word Embeddings**:

  Vector representation of words (for RNN-LSTM)
- **Sentiment Classification:**
  - Support Vector Machines
  - Recurrent Neural network using LSTM (Long Short Term Memory)
- **Output Visualization:**
  - Overall sentiment pie chart
  - Visualization of sentiment combined with metadata
  - Variation in sentiments, if any over a certain time period

Chapter 2 : Literature Survey

2.1 Research Papers

Title : Twitter Sentiment Classification Using Naive Bayes
Based on Trainer Perception
Authors : Mohd Naim Mohd Ibrahim,Mohd Zaliman Mohd Yusoff
Published in : 2015 IEEE Conference on e-Learning, e-Management and e-Services
Abstract:
In this paper,25 tweets were used for training and 25 tweets were used for testing. Tweets were collected using keywords Malaysia and Maybank. Naive Bayes algorithm was used for training purpose, data labelling done is manual and then this data is used for training. Same trainer validates accuracy of the system.
Inference:
Various supervised algorithms are available for twitter sentiment analysis.SVM gives better accuracy than Naive Bayes algorithm, but when labelling of data is done by trainer and same trainer validates the results then accuracy of algorithm increases.First the data is converted in structured form,the available tweets are divided into two parts one for training and one for testing,algorithm is applied and validation takes place.Pros of paper are 90% accuracy achieved,
faster in data classification .Algorithm has certain cons like it considers that features in a tweet are not interrelated.

Title : Preprocessing Boosting Twitter Sentiment Analysis
Authors : Zhao jianqiang,Xi'an, Shaanxi
Published in : 2015 IEEE International Conference on Smart City/SocialCom/SustainCom
Abstract:
Sentiment analysis help various organization to monitor public feelings towards the products.For sentiment analysis,the very first step is preprocessing.The various pre-processing methods cause different influence on

performance of classifiers for each dataset. The experiments show that the accuracy of sentiment classification rises after expanding acronym and replacing negation, although hardly change when removal URL, removal numbers and removal stopword are applied

Inference:

In the paper various preprocessing techniques are used like replace negative mentions,remove URL link,remove numbers,remove stopwords.Word n-grams features are the simplest features for Twitter sentiment analysis.Prior polarity score is a sentiment feature of lexicon-based analysis.The pros of paper is removing all features that make data unstructured and cons are

fluctuation in performance when applied to different classifiers. Removing URL reduces the accuracy of SVM by 0.26% and removing stopword improves 1.43% and 1.68% the accuracy using NB.

Title : Sentiment Analysis on Twitter Data:Case Study on Digital India

Authors : Prerna Mishra ,Dr. Ranjana Rajnish ,Dr.Pankaj Kumar

Published in : 2016 International Conference on Information Technology

Abstract:

The paper performs sentiment analysis of the twitter data set that expresses opinion about Modi ji Digital India Campaign.The tweets collected are classified as Positive, Negative or Neutral. Twitter data is collected for analysis using Twitter API. Dictionary Based approach to analyze data posted by different users.The paper discusses about existing tools available for sentiment analysis, related work,and framework used.

Inference:

Tools available for sentiment analysis are NLTK ,GATE ,Red Opal and Opinion Finder. NLTK toolkit is widely used nowadays for sentiment analysis task. Main features of NLTK used in Sentiment analysis process are Tokenization, Stop Word removal, Stemming and tagging.General Architecture for Text Engineering (GATE) is information Extraction System consisting of modules like Tokenizer, Stemming and Part of speech tagger. This tool is written in Java language.Red opal tool is widely used for users who want to buy any products based on different feature.Opinion Finder is used for analysis of different Subjective sentences related to any topic & classification of sentences is done based on their polarity. Pros of paper are Visualization using pie-chart and cons are no negation handling,credibility of reviews is not considered.

Title : Opinion Mining and Sentiment Polarity on Twitter and Correlation Between Events and Sentiment

Authors : Peiman Barnaghi, John G. Breslin, Parsa Ghaffari

Published in:2016 IEEE Second International Conference on Big Data Computing Service and Applications

Abstract:

Twitter data is a reflection of public sentiment towards events. This paper provides a positive or negative sentiment on Twitter posts using a well-known machine learning method for text categorization. In addition, they use manually labeled (positive/negative) tweets to build a trained method to accomplish a task. The task is looking for a correlation between twitter sentiment and events that have occurred. The trained model is based on the Bayesian Logistic Regression (BLR) classification method. External lexicons detect subjective or objective tweets. Unigram and Bigram features are used with TF-IDF (Term Frequency-Inverse Document Frequency) to filter out the features. To find the correlation between sentiment and events, a timestamp is used to associate each tweet and occurred events during the tournament. The correlation is found using the Pearson correlation coefficient comparing two normalized time series of sentiment polarity and occurred events scores.

Inference:

Data collection, data preprocessing, feature extraction and feature filtering are the important steps performed before training the classifier. Performance of Bayesian logistic regression and Naïve Bayes algorithms are compared. The system tracks the reactions of the public to major events across a timeline. The future scope of the project is that the system can be extended to work with streaming feeds.

Title : Text Sentiment Analysis Based on Long Short-Term Memory

Authors : Dan Li, Jiang Qian

Published in:2016 First IEEE International Conference on Computer Communication and the Internet

Abstract:

This paper promotes a RNN language model based on Long Short Term Memory (LSTM), which can get complete sequence information effectively. In LSTM network each cell in the hidden layer of RNN is replaced with an LSTM cell. To forecast the emotion of a new input review, the LSTM models obtained in the training phase are evaluated on the new input review, giving error values. The model giving the smallest

error value is assigned as the emotional category to the new input review. It is applied to achieve multi-classification for text emotional attributes, and identifies text emotional attributes more accurately than the conventional RNN.

Inference:

Compared with the traditional RNN language model, LSTM is better in analyzing emotion of long sentences. Supervised learning algorithms do not give importance to the order of words. But order is important in many cases. For chinese texts an extra step of splitting into participles is needed. The network is trained on each class of data. It performs better in case of structures with conjunctions.

Title : Deep Learning Approach for Sentiment Analysis of Short Texts

Authors: Abdalraouf Hassan, Ausif Mahmood

Published in: 2017 3rd International Conference on Control, Automation and Robotics

Abstract:

Text classification research starts from designing the best feature extractors to choosing the best possible machine learning classifiers. RNN may be able to relate previous words with next but the relation is not back propogated to the start of the sentence. Thus LSTMs come to picture. ced to model each sentence. In this work, they propose a neural language model ConvLstm, which utilizes both convolution and recurrent layers on top of pre-trained vectors.Many different combinations of hyper-parameters can give similar results. We devoted extra time tuning the learning rate, dropout and the number of units in the convolutional layer, since these hyper-parameters has a large impact on the prediction performance.

Inference:

The convolutional layer can extract high-level features from input sequence efficiently. However, it requires many layers of CNN to capture long-term dependencies. Based on these observations, the proposed model combines a convolutional and a recurrent layer on top of word2vec. Adding a recurrent layer as a substitute to the pooling layer it can effectively reduce the number of the convolutional layers needed in the model in order to capture long-term dependencies. ConvLstm performs better than NaIve Bayes ,SVM, RNN, RNTN ,MV-RNN. ConvLstm outperforms other algorithms. Unsupervised pre-trained of word vectors is a significant feature in deep
learning for NLP.

Title: Sentiment classification using Comprehensive Attention Recurrent models

Authors: Yong Zhang, Meng Joo Er, Rajasekar Venkatesan, Ning Wang and Mahardhika Pratama

Published in: 2016 International Joint Conference on Neural Networks (IJCNN)

Abstract:

In the recent years ,neural networks and deep learning are being used widely in NLP for structured data. Recurrent neural networks (RNN) is a widely used tool to deal with the classification
problem of variable-length sentences.

But conventional RNN only considers preceding words, that is there is no consideration of future context. Thus, a CA-RNN is proposed. The LSTM is used as it addresses the problem of "vanishing gradient" by replacing the self-connected hidden units with memory blocks. Words are represented as vectors as it is observed that it increases performance by preventing loss of word order. Convolution layers are used to save local context.

Inference:

A comparison is made among SVM- unigram, SVM- bigram, Recursive Neural Networks, CA-RNN, CA-LSTM, CA-GRU. The concept of attention is introduced to solve the problem of preceding context. LSTM is introduced to solve the problem of vanishing gradient. The new architecture exploits bidirectional model to access the past and the future contextual information and a convolution layer to capture local contextual information. The combined preceding, succeeding and local context representations are encoded by LSTM. Finally, these representations are used as input to a softmax classifier.

Title:Semi-supervised Dual Recurrent Neural Network for Sentiment Analysis **Authors**:Wenge Rong, Baolin Peng, Yuanxin Ouyang, Chao Li, Zhang Xiong

Published in:2013 IEEE 11th International Conference on Dependable, Autonomic and Secure Computing

Abstract:

Sentiment analysis, which makes it possible for users to infer from statements whether or not the overall sentiment is favourable, is a key technology of information gathering for decision making. The

bag-of-words approach is widely used, but sometimes it fails to obtain the rich relational lexicon structure. The paper proposes a semi-supervised dual recurrent neural network (SDRNN) architecture which is characterized by double recursion between hidden layer and output layer. They have used word embedding before using a feed-forward neural network with a linear projection layer and a non-linear hidden layer that was used to learn the word vector representation and a statistical language model simultaneously.

Inference:

The authors use semi-supervised learning algorithm due to the abundance presence of unlabeled data. Semi-supervised learning makes it possible to use both labelled and unlabelled data. It is noted that performance increases when word embedding is used for initialization. But, this increase is observed only when an existing word embedding tool with specific domain knowledge is available.

Chapter 3 : Requirements

3.1 Functional Requirements

Getting Data From Twitter

A module needs to be designed to extract the required data from Twitter according to the specified policy and period requirements.

Preprocessing the data

The extracted data will undergo preprocessing stages like removal of urls, user mentions, retweets, hashtags, punctuation and then the sentence will be split into tokens.

Classification System

Individual tweets will be classified into two classes- Positive and Negative based on various parameters.

Predicting the overall sentiment

Predicting the overall response to the policy and the percentages of positive and negative responses.

Data visualization

Graphs and other statistics would be displayed to show how well was the policy received, and changes in sentiment, if any, over a period of time.

Selecting the policy

The user is able to select the policy for which the analysis is to be done.

Comparison of algorithms

Comparison between various algorithms to decide the most efficient one.

3.2 Non Functional Requirements

Reliability

The normalised data thus obtained is expected to have 99% reliability.

Scalability

As minimal hardware is used in this project, it is safe to assume that scalability of the system is high.

Compatibility

The data presented by the system is compatible for most of browsers like Google chrome, Internet Explorer, Mozilla Firefox etc.

Availability

The normalised and analysed data is expected to be instantly available to the users of the system at any point in time.

Security Requirements

Since the data deals with personal information of professionals, the data should be accessed or viewed only by authorized users at all times.

Safety Requirements

The data that is processed is backed up to protect loss of data due to virus attack or operating system failure.

3.3 Constraints

Language barrier

The system has been designed to process tweets only in the English language and hence will not be able to process other languages. It will face a problem to decipher other scripts while languages using the same script would face a problem of word analysis.

Fake profiles, bot generated tweets

Recently the menace of fake profiles and more importantly bot generated tweets has hit the microblogging sites and the current system wont be able to tell the authentic and bot tweets apart.

Grammatical errors

Grammatical errors wont be considered in their full sense although an one off error will not affect the system.

Equal importance to each tweet

Every tweet will hold the same weightage, thereby making the opinion of an expert, a journalist and a common man hold the same value.

3.4 Software and Hardware Requirements

Software Requirements

- TensorFlow
- Glove word vector model
- Gensim
- Python
- TwitterScraper
- Microsoft Excel
- Tableau
- Wordpress

Hardware Requirements

- Operating System : Windows, 7 or higher
- RAM: 4GB or higher

3.5 Selection of the Hardware, Software , Technology and tools

Word embeddings are a modern approach for representing text in natural language processing. The word vector models Glove is used. The gensim library is an open-source Python library that specializes in vector space and topic modeling. Gensim acts as a wrapper on Glove making it very easy to use. TensorFlow is one of the best libraries to implement deep learning. It has many built-in functions which reduce the size of our code and make implementation easier.

Chapter 4 : Proposed Design

4.1 System design / Conceptual Design (Architectural)

Below is the architecture for our proposed system.

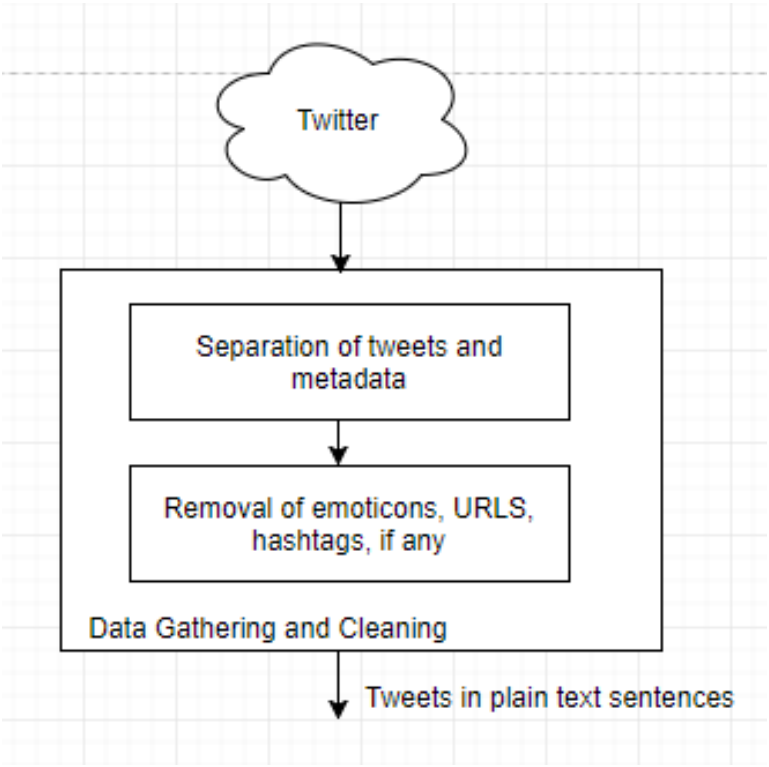


Fig.1.1 : System Architecture

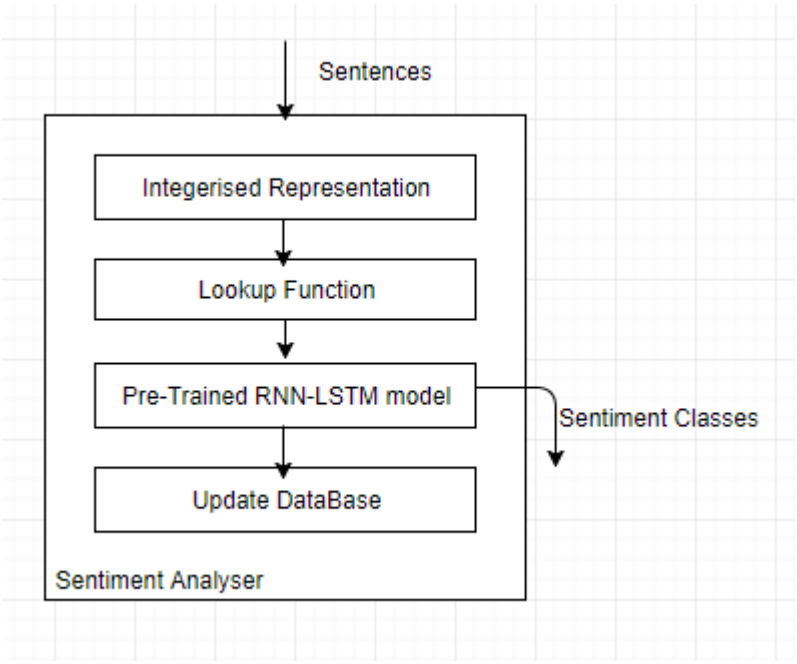


Fig.1.2: System Architecture

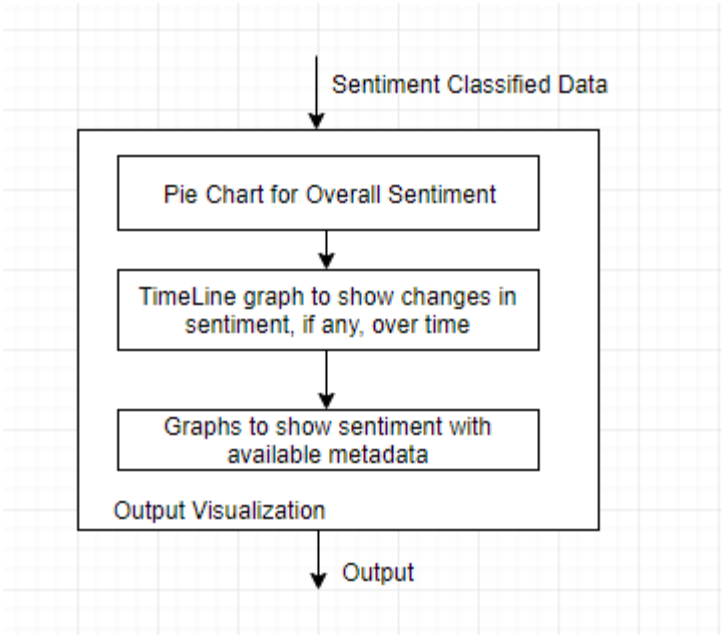


Fig.1.3: System Architecture

4.2 Block Diagram

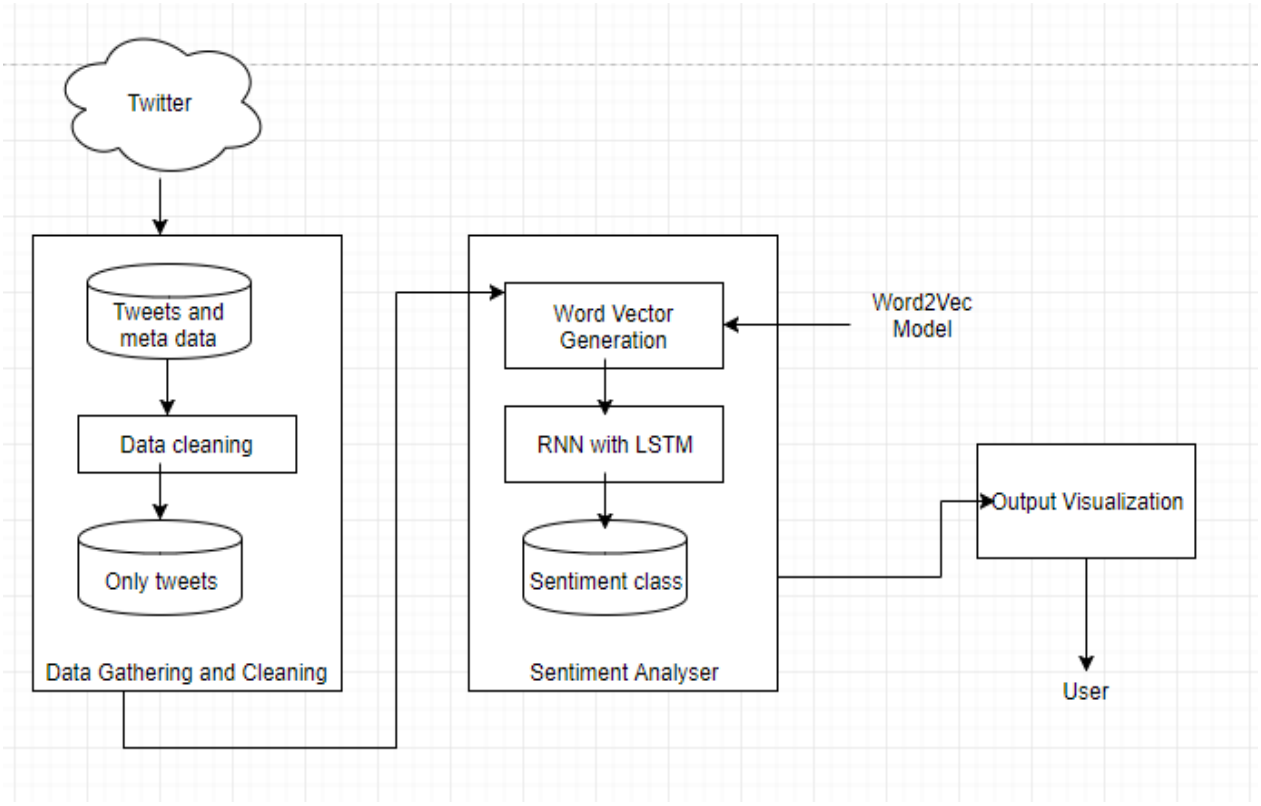


Fig 2 : Block Diagram

The block diagram roughly summarizes the workflow of the project. Data from twitter will be extracted using the Twitter API. After this process of data collection, manual labelling will take place under the preprocessing stage. Following this, this cleaned data is provided to both the RNN algorithm as well as the classifier as an input.

The output of the classifier is then converted into concise visualization and will largely be done using the Tableau tool.

4.3 Design of the proposed system

a) Data Flow Diagram

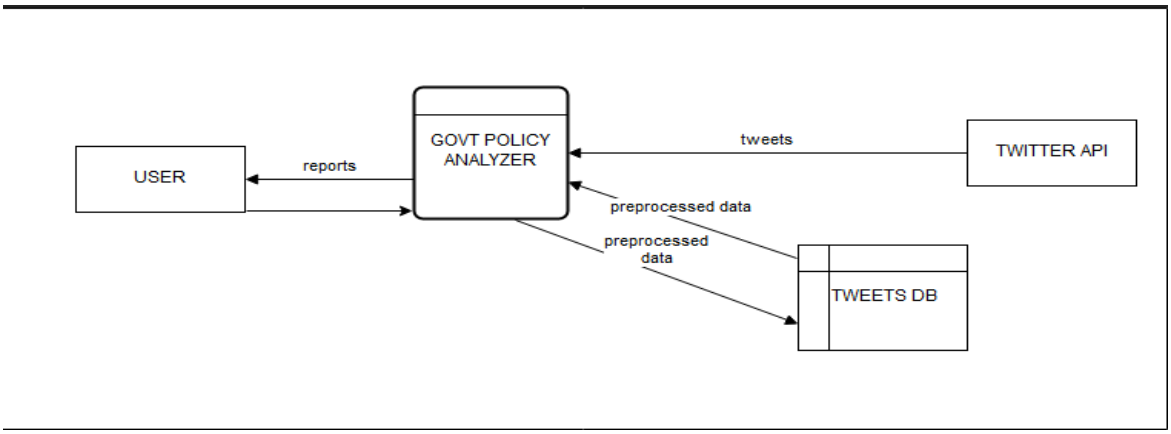


Fig 3 : Level 0 DFD

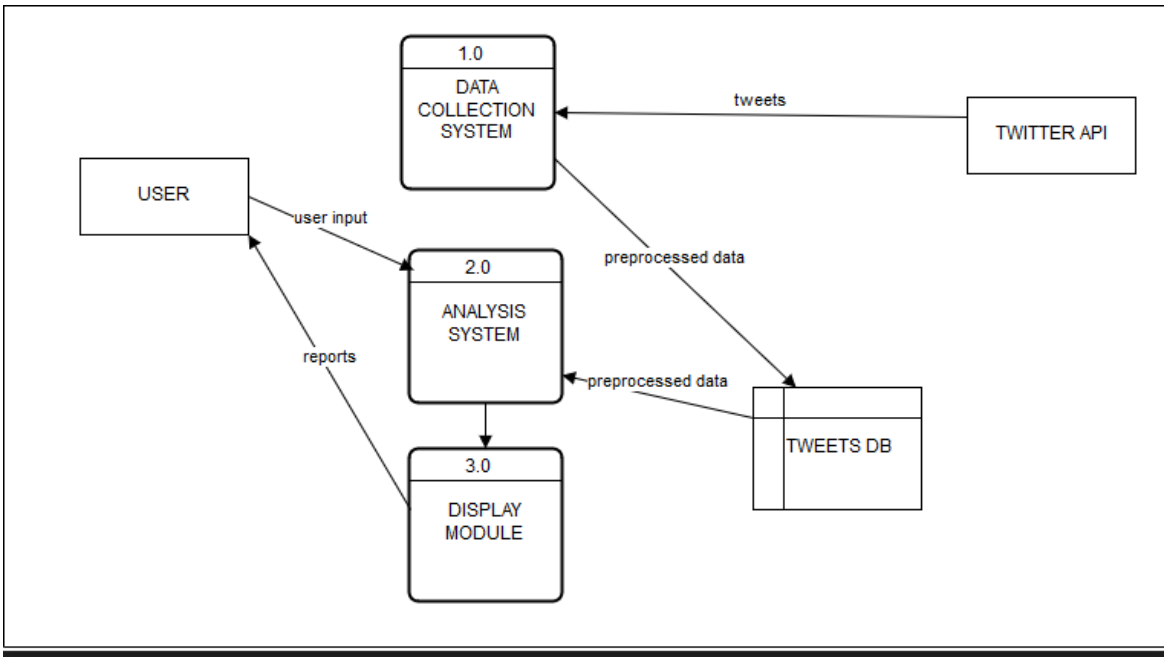


Fig 4 : Level 1 DFD

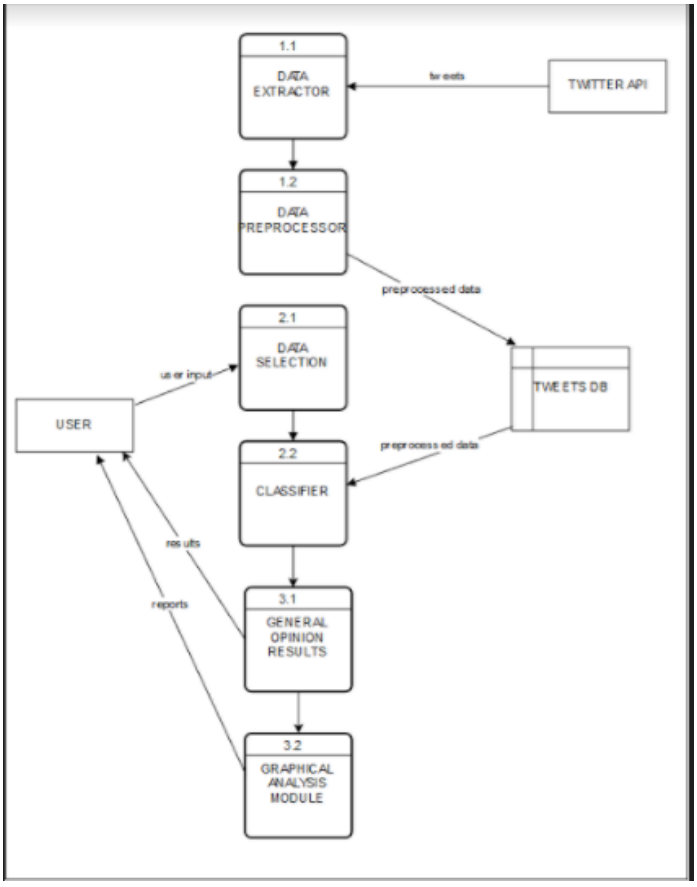


Fig 5 : Level 2 DFD

DFD level 0 has government policy analyzer system which collects data from twitter using Twitter API .Data collected is stored in database which can be used for further processing and analysis. After entire processing, output to user is in the form of reports. In DFD level 1 there is data collection system where twitter API is used for collection of data, analysis system and display module. According to user input analyzer selects the data and applies algorithm for tweet analysis and finally display system provides output to the user. In level 2 Data collection system data collection system ,analysis system, display module are further elaborated to data extractor, data preprocessor, data selection, classification, general opinion results and graphical analysis module for performing different tasks.

b) Flowchart of the proposed system

The flowchart below visually summarises the algorithm we intend to use, its sequence of steps, processing and the eventual output

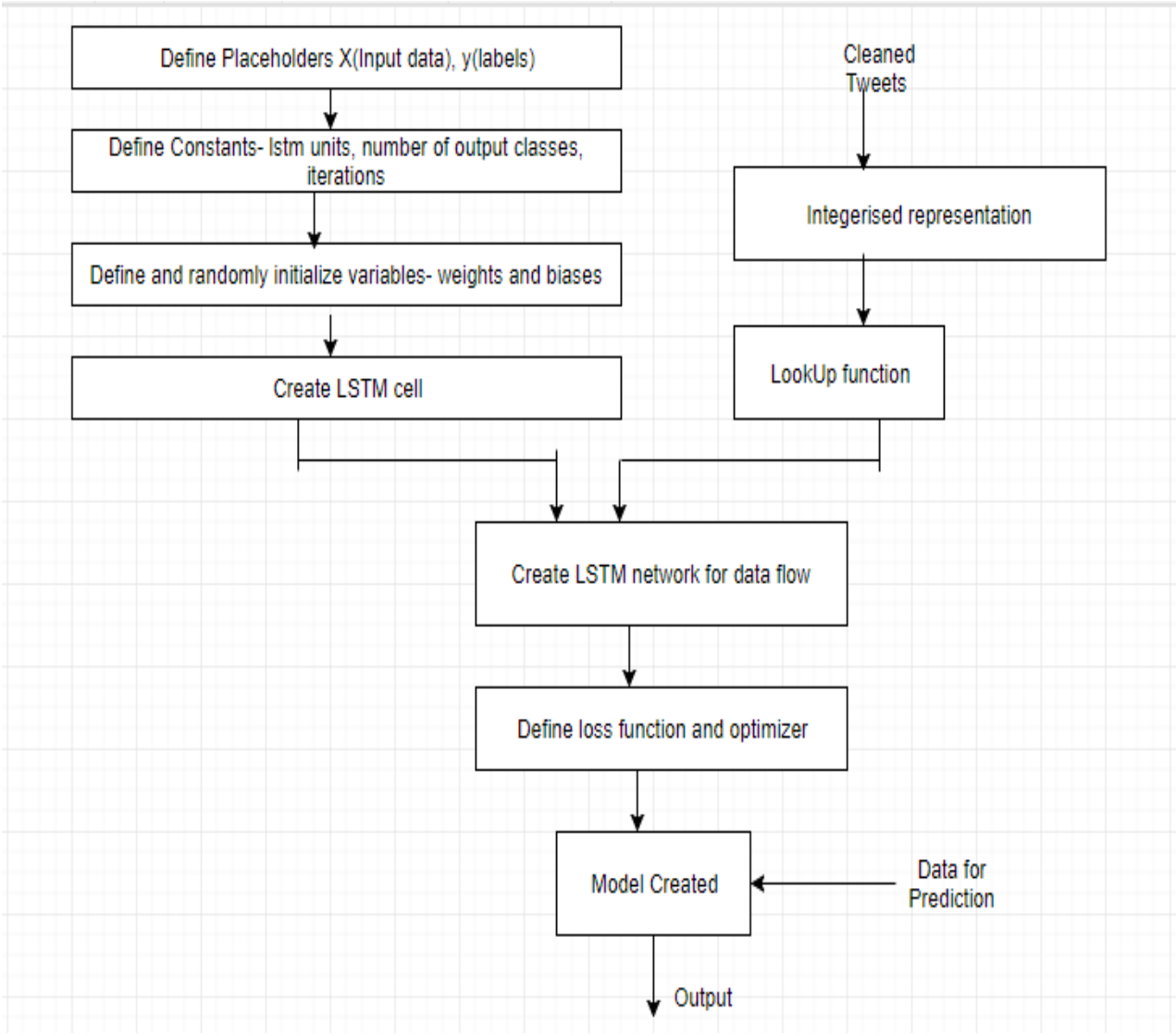


Fig 6: Flowchart

c) Use Case Diagram

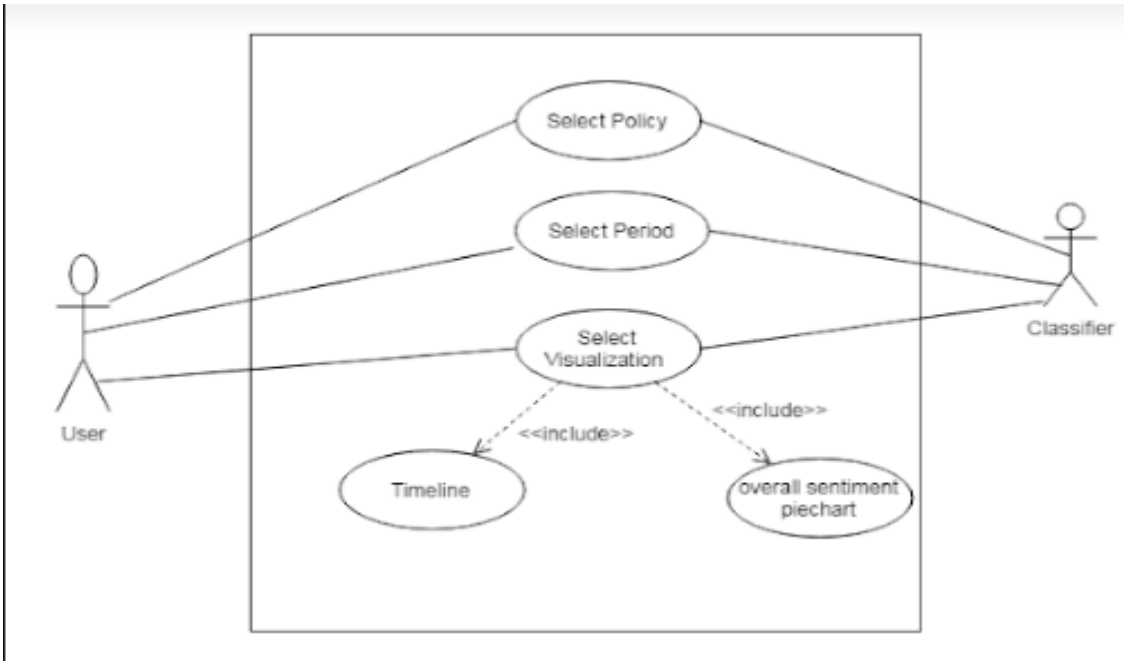


Fig 7: Use Case Diagram

The above Usecase shows how the user interacts with our system. User selects the policy and time period for which analysis is to be performed. Name of the policy and period is provided as input to classifier which then collects and classifies the tweets into positive ,negative. The output from classifier is provided as input to visualization entity. User can select in which form he wants to view the calculated data. Visualization can be of three forms heat maps, timeline based that displays change in views of people over a period of time and overall sentiment score using pie charts. According to user's selection graphs or pie charts are displayed on webpage.

4.4 Project Scheduling and Tracking using timeline/ Gantt Chart

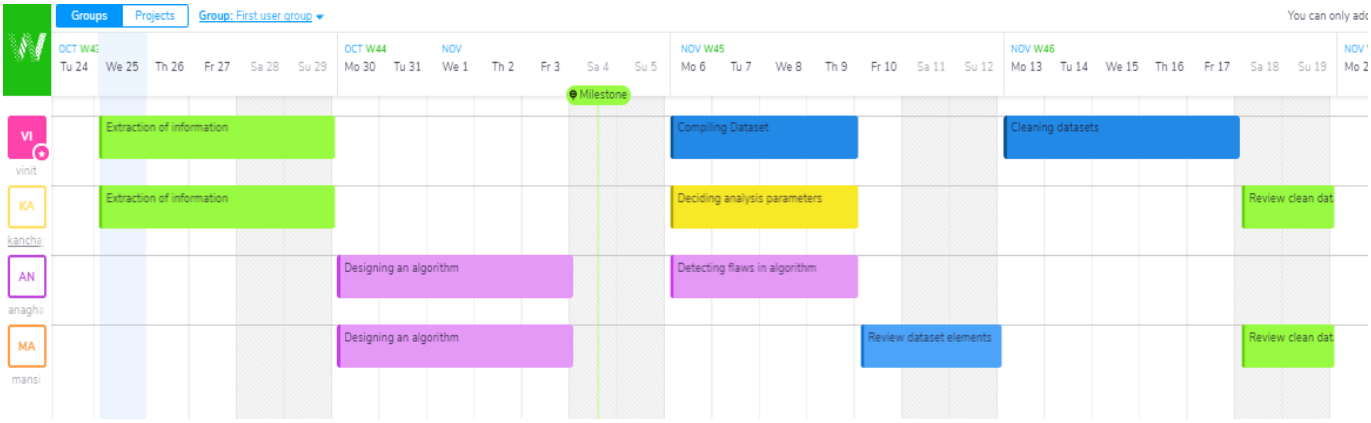


Fig 8 : Timeline 1: 25 October 2017 to November 19 2017

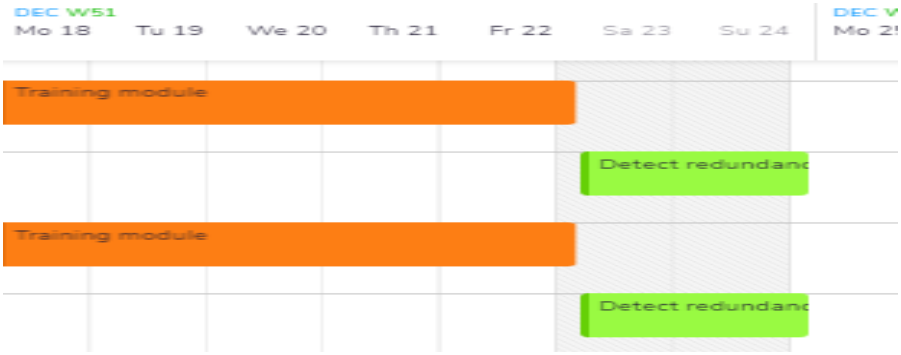


Fig 9 : Timeline 2 : 18 December 2017 to 24 December 2017

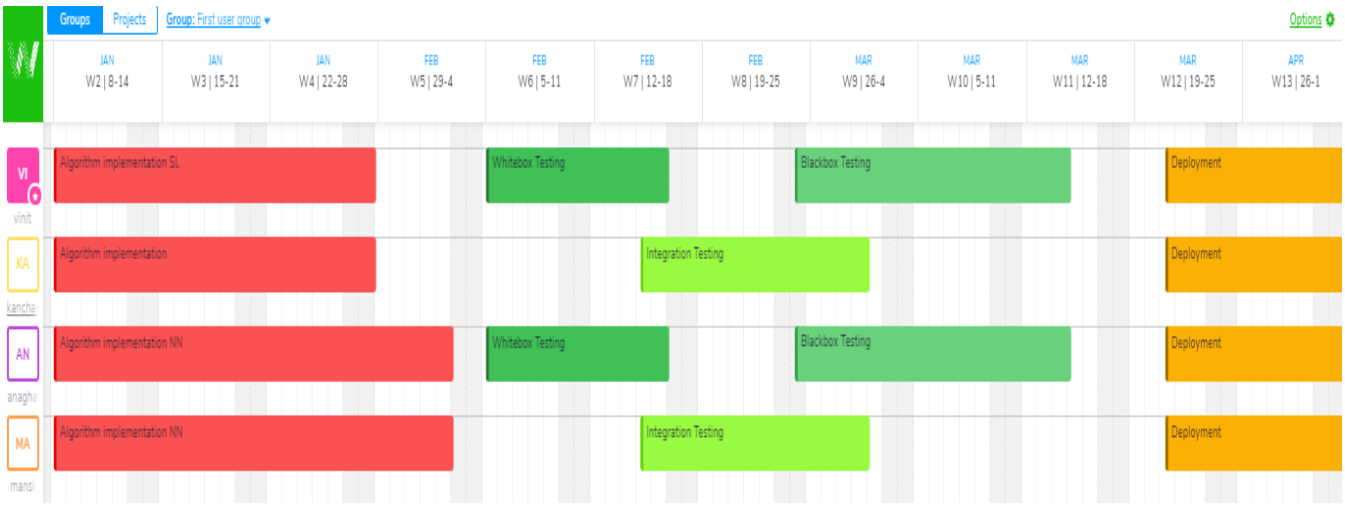


Fig 10 : Timeline 3 : 8 January 2018 to 1 April 2017

\

Chapter 5 : Implementation Details

5.1. Algorithms for the respective modules developed

We propose a system which will analyze twitter reactions to government policies. To get twitter data regarding recent policies, we use the REST API provided by Twitter whereas for older policies we scrape the tweets of the web. Currently, our system focuses only on the Indian government policies implemented after 2014. For supervised learning, we create a manually labeled tweets dataset.

A.Data collection:

Data is collected from twitter using different hashtags related to government policies or schemes.For this purpose twitter api tweepy is used but it doesn't provide data which is older than two weeks so twitter scraper is used for getting older data.Also the urls used in tweets are also scrapped and used for analysis purpose.

All the data is received in JSON file which is then converted to csv format, selecting only the attributes relevant to our analysis.

B.Data preprocessing:

Data is cleaned by removing hashtags,special characters, punctuation,replacing @ with User mention,wide spaces with single space and multiple occurrence of same letter with single letter.Emojis play an important role in expressing sentiment therefore for our model to identify it we replace positive emoji with word positive and negative one with word negative. We store the URLs in a separate file for further analysis.

C.Sentiment analysis method:

For naive bayes and svm algorithm the common steps performed are-

1.For feature generation unigram and bigrams are used.Unigram refers to single word in the dataset and bigrams are group of two words

2.For feature selection TF-IDF is used .It reduces the weight of more common words.

3.The results from step 1 and 2 are passed to naive bayes classifier and svm.

Naive Bayes classification-

Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. It is based on a the principle that the value of a particular feature is independent of the value of any other feature, given the class variable.This classification is based on probabilities and independent assumptions

between different features. It uses parameter estimation for naive Bayes models uses the method of maximum likelihood.

The following steps were used to implement the algorithm in python

1. Scikit learn library is used
2. Multinomialnb () method is used
3. Laplace smoothing is implemented to prevent assigning zero values to features not present in training data set.
4. After training the model predict() function is used to test the data

Support Vector Machine-

An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. It is used to find a hyperplane to categorize the testing dataset.

The following steps were used to implement the algorithm in python:

1. Scikit learn library is used
2. LinearSVC function is used as it is faster to converge as compared to other svm functions
3. l2 penalty is implemented as it gives stable solution and helps in better feature selection
4. Penalty parameter c is set for deciding the margin of hyperplane
5. After training the model ,it is tested with using predict () function.

Rnn with lstm -

The downloaded dataset in json format was loaded and converted to csv. It was further preprocessed to remove urls, user mentions, hashtags. The urls of various blogs were also separated and stored in a text file for future analysis. The urls containing images and videos were removed.

```
#LOADING DATA
with open('Dataset-GST.json', 'r') as f:
    data = json.load(f)
  dataframe = pd.DataFrame(data, columns=['fullname', 'id', 'likes', 'replies','retweets','text','timestamp', 'url','user'])
    dataframe.pop('fullname')
    dataframe.pop('url')
    dataframe.pop('user')

dataframe.to_csv('GST.csv', encoding = "utf-8")

#PREPROCESSING
parsed_tweetlist = []
indexlist = []
```

```
i = 0
while i < 61809:
    parsed_tweet = p.parse(dataframe['text'][i])

    if(parsed_tweet.urls != None):
        parsed_tweetlist.append(parsed_tweet.urls[0].match)
        indexlist.append(i)

    p.set_options(p.OPT.URL, p.OPT.EMOJI, p.OPT.MENTION, p.OPT.RESERVED)
    dataframe['text'][i] = p.clean(dataframe['text'][i])
    i = i + 1


#PREPROCESSING URLS LIST
def check_urls(str_local):
    if(str_local.find("dlvr.it") != -1):
        return 1
    if(str_local.find("pic.twitter.com") != -1):
        return 1
    if(str_local.find("youtu.be") != -1):
        return 1
    if(str_local.find("youtube.com") != -1):
        return 1
    if(str_local.find("/status/") != -1):
        return 1
    return 0

def find_urls(str_local):
    if(str_local.find("timesofindia.com") != -1):
        return 1
    if(str_local.find("financialexpress.com") != -1):
        return 1
    if(str_local.find("economictimes.indiatimes.com") != -1):
        return 1
    if(str_local.find("hindustantimes.com") != -1):
        return 1
    if(str_local.find("thehindu.com") != -1):
        return 1
    if(str_local.find("newindianexpress") != -1):
        return 1
    if(str_local.find("dnaindia.com") != -1):
        return 1
```

```
    if(str_local.find("indianexpress.com") != -1):
        return 1
    if(str_local.find("www.thequint.com") != -1):
        return 1
    if(str_local.find("swarajyamag.com") != -1):
        return 1
    if(str_local.find("www.opindia.com") != -1):
        return 1
    if(str_local.find("www.firstpost.com") != -1):
        return 1
    if(str_local.find("scroll.in") != -1):
        return 1
    if(str_local.find("www.scowhoop.com") != -1):
        return 1
    if(str_local.find("business-standard.com") != -1):
        return 1
    if(str_local.find("theindiancapitalist.com") != -1):
        return 1
    if(str_local.find("newsbharati.com") != -1):
        return 1
    if(str_local.find("deccanchronicle.com") != -1):
        return 1
    if(str_local.find("www.moneycontrol.com") != -1):
        return 1
    if(str_local.find("www.businesstoday.in") != -1):
        return 1
    if(str_local.find("www.ndtv.com") != -1):
        return 1
    if(str_local.find("www.gstindiaexpert.co") != -1):
        return 1
    return 0

url_list = []
url_index = []
i = 0
size = len(parsed_tweetlist)
while i<size:
    if(check_urls(parsed_tweetlist[i]) == 0):
        if(find_urls(parsed_tweetlist[i]) == 1):
            url_list.append(parsed_tweetlist[i])
            url_index.append(indexlist[i])
    i = i+1
```

```
new_size = len(url_list)
print(size)
print(new_size)
url_list = list(set(url_list))

#writing to a text file
output_file = open('GST-URLS.txt', 'w')
for url in url_list:
    output_file.write(url)
    output_file.write("\n")
output_file.close()
```

A 100 dimensional Glove model is downloaded and loaded using the gensim library. All the tweets are taken and each tweet is split into tokens, Each token is represented in 100 dimensions. Thus we get the input cube= no of tweets x no of words in each tweet x 100 values representing the word in each of the 100 dimensions. as the number of words in each tweet maynot be the same, it is zero-padded.

```
glove_input_file = 'glove.6B.100d.txt'
word2vec_output_file = 'glove.6B.100d.txt.word2vec'
glove2word2vec(glove_input_file, word2vec_output_file)
filename = 'glove.6B.100d.txt.word2vec'
glove_model = KeyedVectors.load_word2vec_format(filename, binary=False)

strip_special_chars = re.compile("[^A-Za-z0-9 ]+")
def cleanSentences(string):
    string = string.lower().replace("<br />", " ")
    return re.sub(strip_special_chars, "", string.lower())

max_sequence_length = 55
num_dimensions = 100
size = data.shape[0]
vector_output = np.zeros((size ,max_sequence_length, num_dimensions), dtype='float32')

i = 0
while i < size:
    tweet_text = data['text'][i]
    cleaned_line = clean_sentences(tweet_text)
    split = cleanedLine.split()
    print(split)
    index_counter = 0
    for word in split:
        try:
            vector_output[i][index_counter] = glove_model[word]
            print(vector_output)
```

```
        except KeyError:
            print(split[index_counter])

        index_counter = index_counter + 1
    i = i + 1
```

Specify various parameters of the neural network. Define the basic cell. A dropoutwrapper is used to prevent overfitting. The network has a fully connected hidden layer of 30 LSTM units. It has a output layer of 2 units, where one unit represents a positive class and other unit represents the negative class. Unroll the network. As Tensorflow operates on graphs, reset the default graph. Define placeholders for input and labels. Define the variables of weights and biases and randomly initialize them. Define error and accuracy measures, calculate loss and minimize the loss. Initialize a session and run the model inside it. Save the model and use it to make predictions on unlabelled data.The final output is whether a given tweet is positive or negative.

```
batch_size = 200
num_inputs = 55
num_dimensions = 100
num_outputs = 2
num_neurons = 30
training_iterations = 1000
learning_rate = 0.001

tf.reset_default_graph()

#PLACEHOLDERS
X = tf.placeholder(dtype = tf.float32, shape = [None , num_inputs, num_dimensions], name = 'input_placeholder' )
y = tf.placeholder(dtype = tf.int32, shape = [None ,num_outputs], name = 'labels_placeholder')

#RNN cell
cell = tf.contrib.rnn.BasicLSTMCell(num_units = num_neurons)
lstm_cell = tf.contrib.rnn.DropoutWrapper(cell, output_keep_prob=0.85)
output, _ = tf.nn.dynamic_rnn(cell, X, dtype=tf.float32)

output = tf.transpose(output, [1, 0, 2])
last = tf.gather(output, int(output.get_shape()[0]) - 1)

weight = tf.Variable(tf.truncated_normal([num_neurons, num_outputs]), name = 'weight')
bias = tf.Variable(tf.constant(0.1, shape=[num_outputs]), name ='bias')
prediction =tf.add(tf.matmul(last, weight), bias, name= 'prediction')
```

```
correct_prediction = tf.equal(tf.argmax(prediction,1), tf.argmax(y,1))
accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))
answer = tf.nn.softmax(prediction)

loss = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(logits = prediction, labels = y))
optimizer = tf.train.AdamOptimizer(learning_rate= learning_rate)
train = optimizer.minimize(loss, name ='final_operation')

#Saving a model
saver = tf.train.Saver()

init = tf.global_variables_initializer()

with tf.Session() as sess:
    sess.run(init)
    writer = tf.summary.FileWriter("./outputModel", sess.graph)

    for iteration in range(training_iterations):
        sess.run(train, feed_dict = {X: vector_output, y: labels})

        if (iteration + 1) % 10 == 0:
            error = loss.eval(feed_dict = {X:vector_output , y: labels})
            print(iteration + 1, "\tError:", error)
    saver.save(sess, "./rnn_model", global_step=iteration)
    writer.close()
    error = loss.eval(feed_dict = {X: test_vector_output , y: testBatchLabels})
    print("\tError:", error)
    predictions_testset = answer.eval(feed_dict = {X: test_vector_output})
    #predicting unlabelled data
    unlabelled_answer = answer.eval(feed_dict={X:unlabelled_vector_output}, session=sess)
    print(unlabelled_answer.shape)
```

5.2. Comparative Analysis with the existing algorithms

Opinion Mining is a type of Natural Language processing technique that is used to mine the reviews/opinions about any particular topic, product, service or event. Sentiment classification methods can be classified into three approaches: Dictionary approach, Machine learning approach and Deep learning approach.

A.Dictionary Based Approach

The dictionary methods for sentiment and other text analysis include counting words from a predefined lexicon in a big corpus, in order to explore or test hypotheses about the corpus. In particular, this is often done for sentiment analysis: count positive and negative words (according to a sentiment polarity lexicon, which was derived from human raters or previous researchers' intuitions). For dictionary methods to work well, the scores attached to words must closely align with how the words are used in a particular context. If a dictionary is developed for a specific application, then this assumption should be easy to justify. But when dictionaries are created in one substantive area and then applied to another problems, serious errors can occur.

B.Machine learning Approach

Machine learning techniques have been widely used in sentiment analysis using methods such as Naive Bayes classification, Support Vector Machine and maximum entropy to study the sentiment contained in sentences. The traditional machine learning methods focus on designing hand-crafted features and use sentiment classifiers such as support vector machines (SVM) to accomplish the task. Feature representation is a key component of many machine learning systems because the performance of a machine learner heavily depends on it.A widely used sentence representation model is the bag-of-words (BOW) model which uses vectors to represent words, where each dimension of the vector corresponds to a distinct word. The model achieves very good performance on a variety of tasks but faces the problem of losing word order which is critical for semantic analysis. The bag-of-n-grams model is proposed to consider the word order in short context. Both BOW and bag-of-n-grams models have difficulty capturing the semantic information.

C.Deep learning Approach

Due to the inability of ML approaches in capturing the semantic and structural information deep learning approaches like RNN are being used for sentence classification. Deep learning models have also been effective in tackling the feature representation problem because they can learn features from data automatically. Convolutional neural networks (CNN) and recurrent neural networks (RNN) have been proven to be powerful semantic composition models for sentiment classification. They leverage on randomly initialized word vectors or pre-trained word vectors to represent words and employ the deep neural networks to learn vector representations for sentences of variable length. When the interval between the relative information of texts and the current location to be predicted becomes large, some problems will come out.To overcome this difficulty,LSTM is used.LSTM through deliberate design to avoid long-term dependence, remembers the long term information.

5.3. Evaluation of the developed system (accuracy, Effectiveness, Efficiency)

The RNN-LSTM algorithm gave a accuracy of 88.33%. The Naive Bayes algorithm gave an accuracy of 84.167%. The Support Vector Machine algorithm gave an accuracy of 81.67%

For training RNN-LSTM, first a softmax function was used as the classes were mutually exclusive and then cross entropy was used as a error measure. The predicted labels were compared with the actual labels to find out the misclassified ones and a reduce_mean function was applied to calculate loss and later minimize it using AdamOptimizer.

Chapter 6 Testing

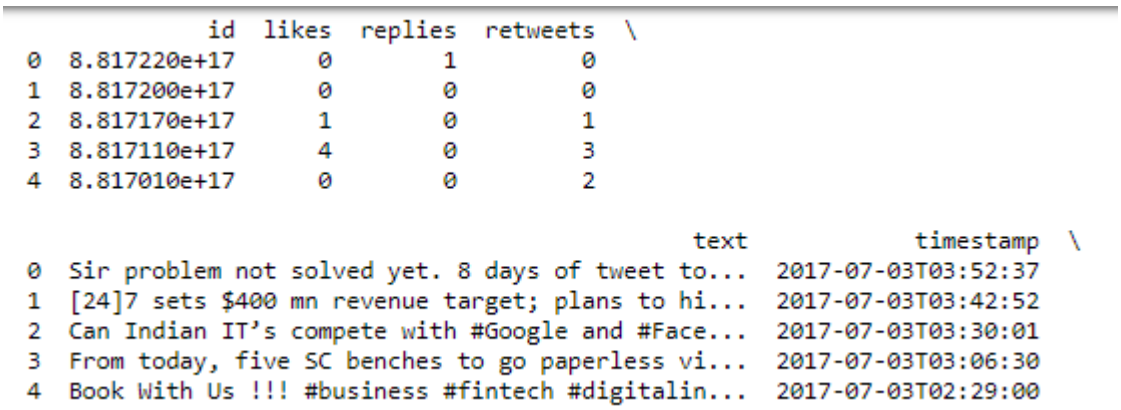
Unit Testing:

1 - Getting data:

Using hashtag #Digital India tweets were scrapped and a total of 96667 tweets were downloaded in json format. A part of the obtained file is attached below:

{"fullname": "Ramanathan B", "id": "953291055608356864", "likes": "0", "replies": "0", "retweets": "0", "text": "Academic certificates across universities- be it Central University, State University, Deemed University or Private University and CBSE, State Boards,ICSE -across boards are digitised, a repository made for easy download or access given for employer verification #DigitalIndia", "timestamp": "2018-01-16T15:41:13", "url": "/ramanathan_b/status/953291055608356864", "user": "ramanathan_b"},"},
{"fullname": "apoorv arora", "id": "953285032314941441", "likes": "0", "replies": "0", "retweets": "0", "text": "@nitin_gadkari @MORTHIndia the fasttag app on play store is useless. Please remove the app. please don't say this is part #DigitalIndia #useless. At least check the reviews on play store.", "timestamp": "2018-01-16T15:17:17", "url": "/apoorv_arora/status/953285032314941441", "user": "apoorv_arora"},
{"fullname": "Lokesh Gupta", "id": "953284397171552262", "likes": "0", "replies": "0", "retweets": "0", "text": "@UIDAI, @ceo_uidai #Aadhaar #DigitalIndia Aadhar is set to remove corruption from India, but feel sad when Aadhar enrollment centers become place of corruption. Setback to digital India. In reference to permanent center syndicate bank shipra suncity indirapuram ghaziabad pic.twitter.com/e0tUwQHnv1", "timestamp": "2018-01-16T15:14:46", "url": "/drlokesh529/status/953284397171552262", "user": "drlokesh529"}, {"fullname": "Sudipta Sengupta", "id": "953283518775242752", "likes": "1", "replies": "0", "retweets": "0", "text": "#DigitalIndia Hard reality. pic.twitter.com/JpKuEODxDn", "timestamp": "2018-01-16T15:11:16", "url": "/SenguptaSudipto/status/953283518775242752", "user": "SenguptaSudipto"},

This is then converted to a csv format and the relevant fields are extracted. A snapshot of a part of the csv file is attached below:



	id	likes	replies	retweets	\
0	8.817220e+17	0	1	0	
1	8.817200e+17	0	0	0	
2	8.817170e+17	1	0	1	
3	8.817110e+17	4	0	3	
4	8.817010e+17	0	0	2	

	text	timestamp	\
0	Sir problem not solved yet. 8 days of tweet to...	2017-07-03T03:52:37	
1	[24]7 sets $400 mn revenue target; plans to hi...	2017-07-03T03:42:52	
2	Can Indian IT’s compete with #Google and #Face...	2017-07-03T03:30:01	
3	From today, five SC benches to go paperless vi...	2017-07-03T03:06:30	
4	Book With Us !!! #business #fintech #digitalin...	2017-07-03T02:29:00	

Fig 11 Output 1

2 - Preprocessing Data:

Tweet Before Preprocessing:

"@nitin_gadkari @MORTHIndia the fasttag app on play store is useless. Please remove the app. please don't say this is part #DigitalIndia #useless. At least check the reviews on play store."

Tweet After Preprocessing:

the fasttag app on play store is useless. Please remove the app. please don't say this is part #DigitalIndia #useless. At least check the reviews on play store.

3- Splitting data into Tokens and creating vector representations:

Preprocessed tweet:

Academic certificates across universities- be it Central University, State University, Deemed University or Private University and CBSE, State Boards,ICSE -across boards are digitised, a repository made for easy download or access given for employer verification #DigitalIndia

Tokens generated:

['academic', 'certificates', 'across', 'universities', 'be', 'it', 'central', 'university', 'state', 'university', 'deemed', 'university', 'or', 'private', 'university', 'and', 'cbse', 'state', 'boardsicse', 'across', 'boards', 'are', 'digitised', 'a', 'repository', 'made', 'for', 'easy', 'download', 'or', 'access', 'given', 'for', 'employer', 'verification', 'digitalindia']

Vector representation:

Each word is represented as a 100 dimensional vector, given that the above sentence has 35 words, a 35 x 100 array is created. A snap of initial values of this array is shown below:

```
[[[ 9.3277e-02  5.8116e-01 -9.1645e-01  5.9441e-01  1.3727e-01
    4.7263e-02 -6.2173e-02 -2.9051e-01 -6.4246e-01  1.2054e+00
   -2.9904e-01 -4.3561e-01  6.8445e-01 -2.6390e-02 -9.3905e-01
    6.5104e-02  2.8819e-01  4.8421e-02 -5.7881e-01  4.8836e-01
   -1.5051e+00  1.7803e-01  8.9856e-02 -5.9085e-02 -2.4697e-01
   -8.2687e-01  2.5968e-01 -1.1509e+00 -4.1868e-01  1.6420e-01
   -4.8780e-01  3.1422e-01 -5.8969e-01 -4.0922e-01 -6.1252e-01
    1.1828e-01 -7.7830e-01  6.2881e-01 -1.1886e+00  7.1297e-01
   -7.1828e-01 -8.4442e-03 -1.3985e-01  2.1247e-01  2.8039e-01
   -4.4731e-01  5.5717e-01 -2.1173e-01 -2.7768e-01 -7.3766e-03
    3.2260e-01 -6.1955e-01 -3.0851e-01  3.2807e-01  1.1295e-01
   -1.7032e+00  5.6008e-01 -6.3917e-01  1.1946e+00  2.8336e-01
   -2.0415e-01  2.4888e-01 -5.5995e-01  1.9578e-02  4.9013e-01
   -6.6842e-01  3.0731e-01 -6.5632e-02  1.0649e+00  5.4847e-01
    4.3204e-01  5.3957e-01  3.6061e-01 -2.8068e-01 -1.0612e+00
    4.5680e-01 -1.6299e-01  1.6706e-02 -5.5744e-02 -1.0555e+00
    2.0016e-01  5.4683e-01  3.0542e-03 -3.7420e-01 -1.3576e+00
    6.2262e-02 -2.5211e-01 -2.1830e-01  7.5724e-01 -5.9877e-01
    3.3005e-02  5.7674e-01  1.0316e-01  2.9476e-01 -1.5509e-01
    5.9038e-01 -1.0464e-01 -6.8233e-01  5.4254e-01  8.3344e-01]
```

Fig 12 Output 2

4- The RNN- LSTM model:

The output of this model is the printing of loss values after every 10 iterations. A snapshot of the same has been attached below:

```
10      Error: 0.68876314
20      Error: 0.65891814
30      Error: 0.6073391
40      Error: 0.5266077
50      Error: 0.4081329
60      Error: 0.28944466
70      Error: 0.19911642
80      Error: 0.15784754
90      Error: 0.13486001
100     Error: 0.1369611
```

Fig 13 Output 3

Thus, unit testing was performed on various modules as mentioned above, then they were integrated together and the entire flow of program was checked once again to see whether proper outputs were obtained.

Chapter 7 : Result Analysis

7.1 Parameters considered

While the dataset had quite a few attributes, the various parameters conisdered for producing the final graphical outputs are

1. Tweet ID
2. Likes
3. Timestamp
4. Sentiment Class

7.2 Graphical outputs

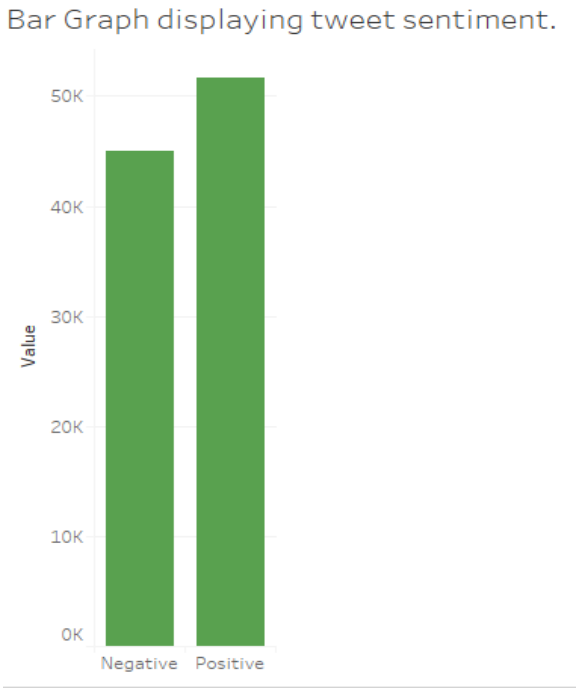


Fig 14 Bar Graph displaying sentiment.

The basic bar graph just shows the split between the positive and negative sentiment. Such a basic representation doesn't give us a deep insight but a brief view regarding the public opinion.

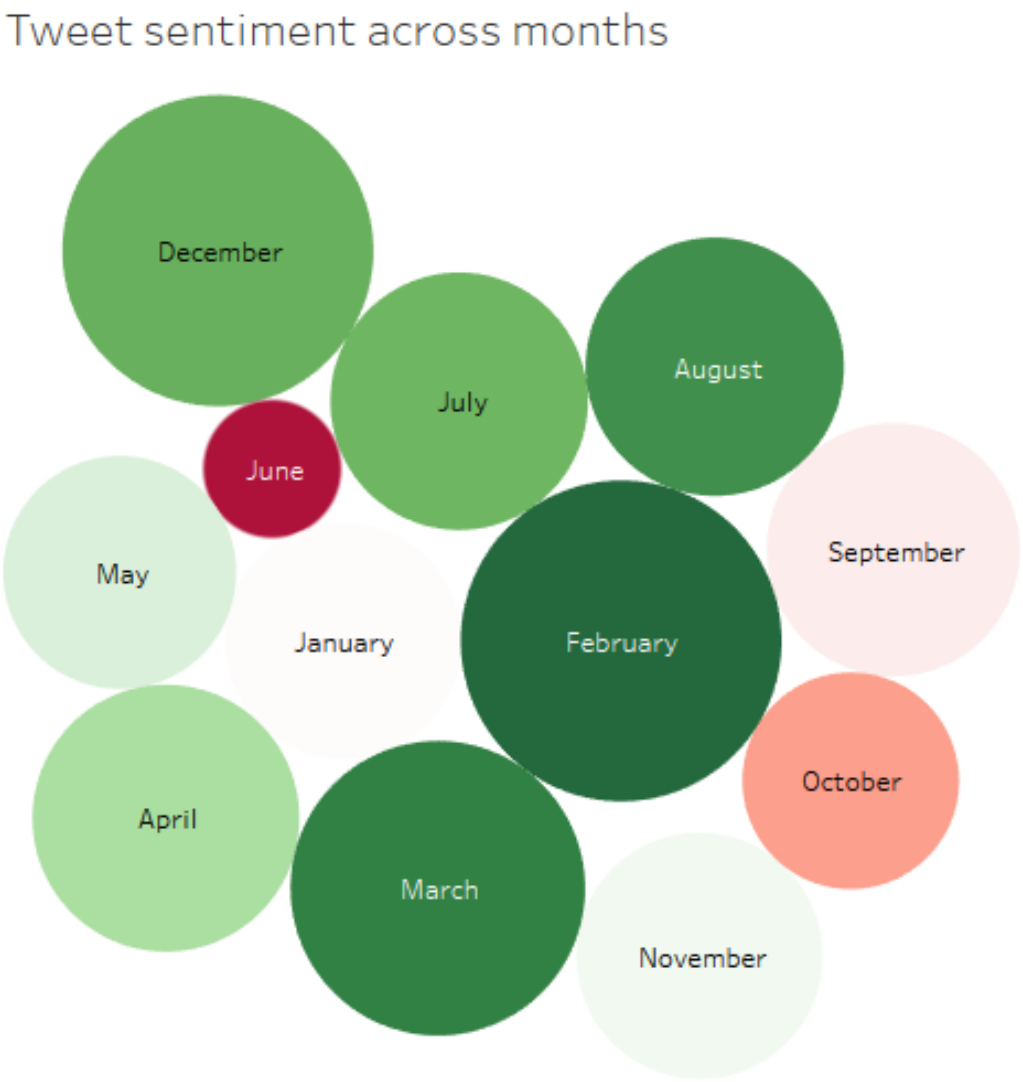


Fig. 15 Bubble chart showing sentiment across months

This particular visual tells us how the sentiment changes across months. The darker the green, the more positive the sentiment, while white indicates a balanced general opinion, while red indicates negative opinions. The size of the circle indicates the number of tweets that month.

Such an analysis across timelines gives us an insight into how the public reaction has changed over time and thereby what actions taken in that period led to a swing in the trend. This can largely help in political engineering.

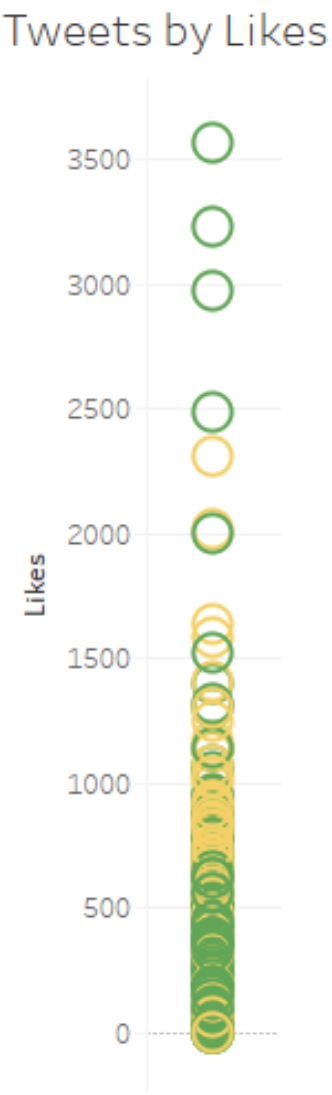


Fig.16 Scatter plot analysing Tweets across likes .

A discrete plot of the tweets against the likes help us taking into consideration the views of the people who dont usually tweet, and hence arent reflected in simple visuals like bar graph, but react to relevant political tweets. This particular visual shows us how the tweets with maximum likes are mostly positive, which means more people agree with positive sentiment.

Chapter 8 : Conclusion

8.1 Limitations

The project faces certain limitations which will turn to be the major challenges going ahead. The basic challenge is the language barrier. The current system only considers tweets in English. Also, the system is not designed to detect sarcasm. The other limitations are grammatical errors, fake profiles and bot generated tweets.

8.2 Conclusion

The exponential rise of social media has allowed people to make their honest opinions about various topics public without a sense of fear. Twitter is one of the leading platforms people prefer to voice their opinions, especially about various government related issues such as current or previous policies undertaken or implemented by the government.

Our approach largely involves carrying out a comparison between the supervised learning method and recurrent neural network in order to establish the better technique . Such insights from the system would have a considerable effect and could prove to be a pivotal factor in election campaign planning, helping think tanks come up with targeted rallies and crucial pressure points.

8.3 Future scope

Recent events have clearly shown how data is crucial in influencing election results. We have created a flat file containing the list of all the URLs tweeted ny the users regarding policies. By scraping these articles or blog posts from these URLs, we can get a deeper and better insight into the opinions.

References

Journal Papers

[1] Mohd Naim Mohd Ibrahim,Mohd Zaliman Mohd Yusoff , “Twitter Sentiment Classification Using Naive Bayes Based on Trainer Perception” published in 2015 IEEE Conference on e-Learning, e-Management and e-Services

[2] Zhao Jianqiang,Xi’an, Shaanxi , “ Preprocessing Boosting Twitter Sentiment Analysis” published in 2015 IEEE International Conference on Smart City/SocialCom/SustainCom

[3] Prerna Mishra ,Dr. Ranjana Rajnish ,Dr.Pankaj Kumar , “Sentiment Analysis on Twitter Data:Case Study on Digital India, published in, 2016 International Conference on Information Technology

[4] Peiman Barnaghi, John G. Breslin, Parsa Ghaffari, “Opinion Mining and Sentiment Polarity on Twitter and Correlation Between Events and Sentiment”, published in 2016 IEEE Second International Conference on Big Data Computing Service and Applications

[5] Dan Li, Jiang Qian, “Text Sentiment Analysis Based on Long Short-Term Memory”, published in 2016 First IEEE International Conference on Computer Communication and the Internet

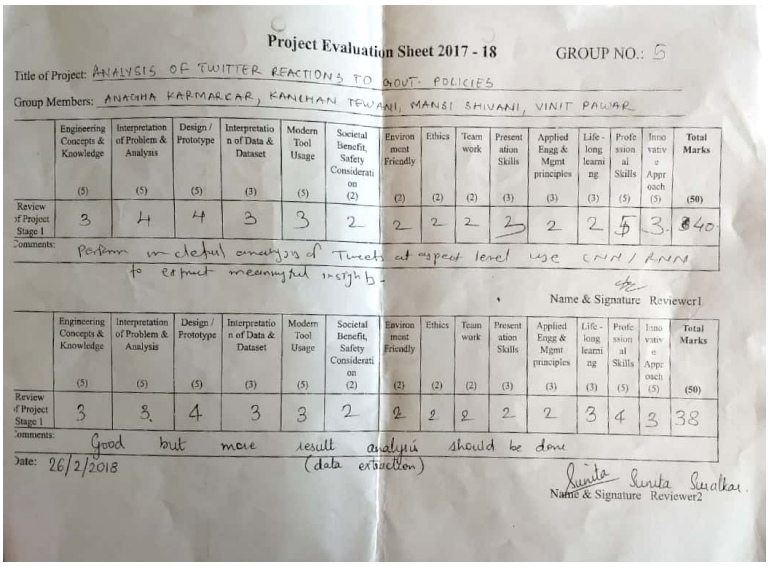
[6] Abdalraouf Hassan, Ausif Mahmood , “Deep Learning Approach for Sentiment Analysis of Short Texts”, published in 2017 3rd International Conference on Control, Automation and Robotics

[7]Yong Zhang, Meng Joo Er, Rajasekar Venkatesan, Ning Wang and Mahardhika Pratama, “Sentiment classification using Comprehensive Attention Recurrent models” published in 2016 International Joint Conference on Neural Networks (IJCNN)

[8] Wenge Rong, Baolin Peng, Yuanxin Ouyang, Chao Li, Zhang Xiong, “Semi-supervised Dual Recurrent Neural Network for Sentiment Analysis”, published in 2013 IEEE 11th International Conference on Dependable, Autonomic and Secure Computing

Project Progress Review Sheets

Review Sheet 1



Project Evaluation Sheet 2017 - 18 GROUP NO.: 5

Title of Project: ANALYSIS OF TWITTER REACTIONS TO GOVT. POLICIES

Group Members: ANAGHA KARMARKAR, KANCHAN TEWANI, MANSI SHIVANI, VINIT PAWAR

	Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (3)	Applied Engg & Mgmt principles (3)	Life - long learning (3)	Professional Skills (5)	Innovative Approach (5)	Total Marks (50)
Review of Project Stage 1	3	4	4	3	3	2	2	2	2	3	2	2	5	3	40

Comments: Perform in detail analysis of Tweets at aspect level use CNN / RNN to extract meaningful insights.

Name & Signature Reviewer1

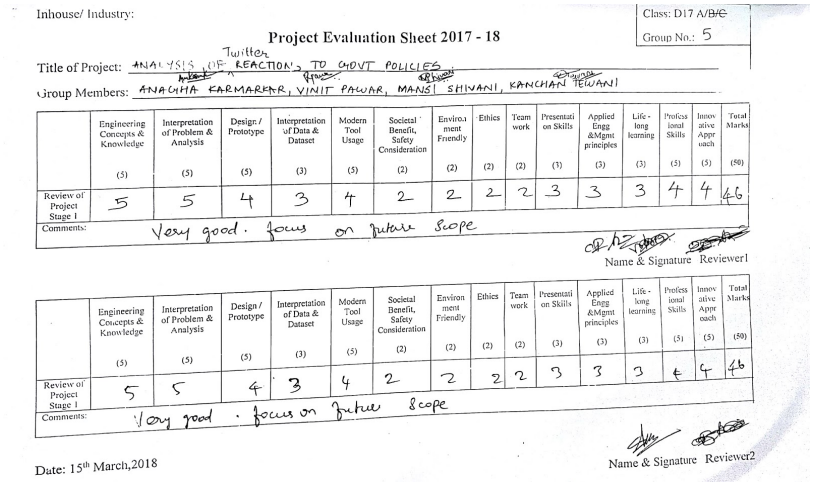
	Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (3)	Applied Engg & Mgmt principles (3)	Life - long learning (3)	Professional Skills (5)	Innovative Approach (5)	Total Marks (50)
Review of Project Stage 1	3	3	4	3	3	2	2	2	2	2	2	3	4	3	38

Comments: Good but more result analysis should be done (data extraction)

Date: 26/2/2018

Sunita Sunita Sawalkar.
Name & Signature Reviewer2

Review Sheet 2



Inhouse/ Industry:

Class: D17 A/~~B/C~~

Group No.: 5

Project Evaluation Sheet 2017 - 18

Title of Project: ANALYSIS OF ~~TWITTER~~ REACTIONS TO GOVT POLICIES

Group Members: ANAGHA KARMARKAR, VINIT PAWAR, MANSI SHIVANI, KANCHAN TEWANI

	Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (3)	Applied Engg &Mgmt principles (3)	Life - long learning (3)	Professional Skills (5)	Innovative Approach (5)	Total Marks (50)
Review of Project Stage 1	5	5	4	3	4	2	2	2	2	3	3	3	4	4	46

Comments: Very good. focus on future scope

Name & Signature Reviewer1

	Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (3)	Applied Engg &Mgmt principles (3)	Life - long learning (3)	Professional Skills (5)	Innovative Approach (5)	Total Marks (50)
Review of Project Stage 1	5	5	4	3	4	2	2	2	2	3	3	3	4	4	46

Comments: Very good . focus on future scope

Date: 15th March,2018

Name & Signature Reviewer2

Appendix

List of Figures

Analysis of Twitter Reactions to Government Policies

Mrs. Vidya Zope
Assistant Professor
Department of Computer Engineering
Vivekanand Education Society's Institute of Technology
Mumbai, India

Anagha Karmarkar
Department of Computer Engineering,
Vivekanand Education Society's Institute of Technology
Mumbai, India

Vinit Pawar
Department of Computer Engineering
Vivekanand Education Society's Institute of Technology
Mumbai, India

Mansi Shivani
Department of Computer Engineering
Vivekanand Education Society's Institute of Technology
Mumbai, India

Kanchan Tewani
Department of Computer Engineering
Vivekanand Education Society's Institute of Technology
Mumbai, India

***Abstract*— With the increasing use of social media, especially twitter, the way governments policies are perceived can be understood through the reactions they garner on twitter. In India, in recent times, there have been many significant policy changes like demonetization, Goods and Services Tax to name a few. Sentiment analysis of twitter reaction to these policy changes across different regions can provide very useful insights.**

In this paper, we talk about sentiment analysis of twitter reactions, its use in the government policies context, the various methods which can be used for it and output visualization for better understanding.

Keywords*— *sentiment analysis, twitter reactions, government policies, social media analysis, RNN, LSTM, neural network, opinion mining

I. Introduction

In recent times, twitter has gained a lot of importance in shaping public opinion. Many politicians and other eminent public figures use the platform to connect with people and many governmental departments have their official twitter handles too.

Currently, India is witnessing introduction of new reforms like Goods and Services Tax, demonetization, Aadhar for delivering government aid and many others. The penetration of social media, especially Twitter, now offers a cheaper and extremely efficient approach to accumulate public opinions about such changes. Individuals spread across the map voice their opinions in the forms of tweets, which could give us an insight as to how the policy was received. This is largely helpful to government and private think tanks as well as news organizations.

Traditionally, the only way to tap public opinion was grass root level surveys. These were expensive, time consuming and limited in terms of dataset size. Sentiment analysis of twitter data offers a better way of analyzing public opinion. The tweets are classified as positive, negative or neutral. Many supervised machine learning algorithms can be used for this. However, the advent of deep learning techniques in recent years is providing a greater impetus to such kind of text analysis by way of more accurate predictions.

This paper is divided VII sections. In section II we have talked about the social media giant Twitter and data extraction from it. Section III and IV discuss the various algorithms which can be used for sentiment analysis and the related work done in this field. Section V describes our proposed system. Section VI specifies the result and conclusions of our work.

II. Twitter Data for Analysis

Twitter, a popular microblogging site has a 140 character limit for tweets as of writing this paper. The historical tweet data can be extracted using the REST API provided by Twitter by using hash tags. For live tweet gathering the Twitter streaming API can be used. Twitter also provides metadata along with the tweets like geo-tagged location of the user.

The twitter urls and emoticons are removed to clean the data. If using supervised learning other preprocessing steps

like lemmatization which reduces a word to its root (lemma), stop words removal are also performed.

III. Methods For Sentiment Analysis

A dictionary based method is the simplest approach to this problem. A dictionary listing the positive and negative words is created and the individual tweets are separated into their lemmatized words and compared with the dictionary to give a overall sentiment score which helps in classifying the tweet as positive or negative. [3]

A supervised machine learning approach uses algorithms like Support Vector Machines, Logistic Regression, Naïve Bayes to predict the sentiment score. Words in a tweet act as features to these classifiers. Naïve Bayes uses conditional probability to predict the probability of a tweet belonging to a particular class. Support Vector Machines map an input to high dimensional planes using various kernels. [1][4]

However these supervised learning techniques operate on individual words. We may use bigrams, trigrams or n-grams to group two, three or n words together respectively. However, long sequences of words which affect the sentiment cannot be efficiently captured by the above mentioned techniques.

In recent times, deep learning has been used to perform sentiment analysis. The emotion conveyed by a word is influenced by what words come before and after it. In recurrent neural networks, each word corresponds to a particular time step. It has hidden function which contains information about words seen previously. Thus, value of hidden state for a current word input is determined by previous hidden state value and current word vector. This helps preserve relationships between different words separated by distance and often leads to more accurate results. [5][6]

IV. Related Work

In this paper, the author has carried out sentiment analysis on the government initiative “Digital India”. Various data preprocessing techniques were discussed and a dictionary-based approach was used.[3]

In this paper the tweets during the FIFA 2014 world cup were analyzed. A Bayesian logistic regression classifier was used as a classifier, n-grams was used for feature extraction and the WEKA machine learning framework was employed. Correlation between events and sentiments were studied by studying how the sentiment fluctuated in the aftermath of the Suarez biting Italy’s defender and later apologising and finalizing the contract with Barcelona.[4]

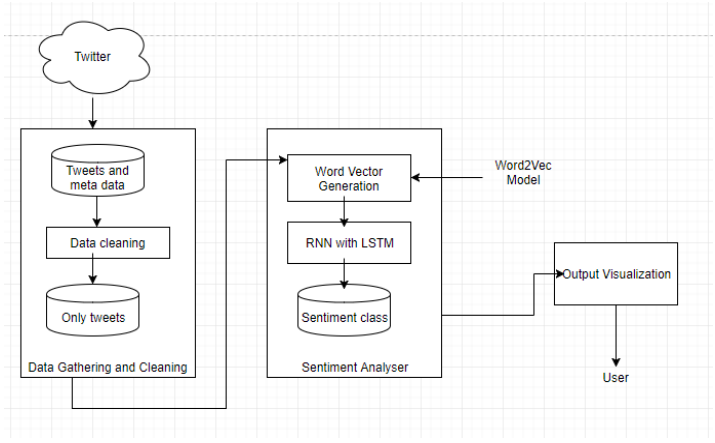
This paper discusses various deep learning methodologies like Convolutional Neural Network(CNN), Recurrent Neural Network(RNN), Long Short Term Memory Cell(LSTM) for sentiment classification on IMDB movie review dataset and Stanford sentiment treebank(SST) dataset. It uses word2vec for obtaining word embeddings. It combines historical, present and future context representations to obtain a comprehensive context representation.[7]

This paper uses RNN and LSTM to perform sentiment analysis. This model results in better analysis of long sentences used to convey emotions leading to better accuracy. Three separate LSTMs were trained on positive, negative and neutral tweets.[5]

V. Proposed System

We propose a system which will analyze twitter reactions to government policies. To get twitter data regarding historical policies, we use the REST API provided by Twitter whereas the data for a recent policy we use the Twitter Streaming API to get tweets. Currently, our system focuses only on the Indian government policies implemented after 2014. For supervised learning, we create a manually labeled tweets dataset.

The block Diagram of the system is specified below:



A user will be able to select a policy from the drop down menu and also period for the analysis if he/she wishes to. We store the tweets in CSV files. We use two different algorithms Support Vector Machines (SVM) and Recurrent Neural networks- Long Term Short Memory (RNN- LSTM) and also provide a comparison between these two methods for sentiment analysis in particular. For SVM, we perform various preprocessing steps like lemmatization and stopword removal and use the bigram technique for feature extraction. Then we train the SVM classifier on our manually labeled dataset and classify the tweets as positive, negative or neutral. We use word vectors as input to RNN-LSTM and Word2Vec is used to create the word embeddings. We use the Tensorflow environment to implement the neural network. The output of these two algorithms is compared.

We can also obtain the user location from the Twitter. This enables us to analyze how the policies fared across various states and in particular cities. Also, we show how the

sentiment varied across time, this will enable us to ascertain if certain events like announcements or speeches by the authorities lead to a change in people's perception. We will be able to correlate sentiment changes, if any, with certain events.[4] We use the Tableau tool for all the output visualizations

VI. Results And Conclusion

This paper discusses the importance of sentiment analysis to help policy makers and private think tanks analyze how people perceive certain government policies. It further discusses the various dictionary based, supervised learning based and deep learning based approaches to sentiment analysis. Finally, a system is proposed for analysing twitter reactions to Indian government policies.

References

[1] Mohd Naim Mohd Ibrahim, Mohd Zaliman Mohd Yusoff "Twitter sentiment classification using Naive Bayes based on trainer perception," Learning, e-Management and e-Services (IC3e), 2015 IEEE Conference on 24-26 Aug. 2015, INSPEC Accession Number. 15787751

[2] Zhao Jianqiang "Pre-processing Boosting Twitter Sentiment Analysis," Learning, 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), INSPEC Accession Number. 15986223

[3] Prem Mishra, Ranjana Ranjish, Pankaj Kumar, "Sentiment analysis of Twitter data: Case study on digital India," Information Technology (InCITe) - The Next Generation IT Summit on the Theme - Internet of Things: Connect your Worlds, International Conference on 6-7 Oct. 2016, INSPEC Accession Number. 16674097

[4] Peiman Barnaghi, Parsa Ghaffari, John G. Breslin, "Opinion Mining and Sentiment Polarity on Twitter and Correlation between Events and Sentiment," Big Data Computing Service and Applications (BigDataService), 2016 IEEE Second International Conference on 29 March-1 April 2016, INSPEC Accession Number. 16022629

[5] Dan Li, Jiang Qian, "2016 First IEEE International Conference on Computer Communication and the Internet (ICCCI)," INSPEC Accession Number 16540645.

[6] Qurat Tul Ain, Mubashir Ali, Amna Riaz, Amna Noureen, Muhammad Kamran, Babar Hayat and A. Rehman, "Sentiment Analysis Using Deep Learning Techniques: A Review,". International Journal of Advanced Computer Science and Applications, Vol. 8, No. 6, 2017

[7] Yong Zhang, Meng Joo Er, Rajasekar Venkatesan, Ning Wang and Mahardhika Pratama, "Sentiment Classification Using Comprehensive Attention Recurrent Models," Neural Networks (IJCNN), 2016 International Joint Conference on 24-29 July 2016, INSPEC Accession Number16446844 .

Plagiarism Report

Plagiarism Scan Report

Summary	
Report Genrated Date	21 Apr, 2018
Plagiarism Status	100% Unique
Total Words	922
Total Characters	6078
Any Ignore Url Used	

Content Checked For Plagiarism:

With the increasing use of social media, especially twitter, the way governments policies are perceived can be understood through the reactions they garner on twitter. In India, in recent times, there have been many significant policy changes like demonetization, Goods and Services Tax to name a few. Sentiment analysis of twitter reaction to these policy changes across different regions can provide very useful insights.
In this paper, we talk about sentiment analysis of twitter reactions, its use in the government policies context, the various methods which can be used for it and output visualization for better understanding.
Keywords— sentiment analysis, twitter reactions, government policies, social media analysis, RNN, LSTM, neural network, opinion mining
Introduction
In recent times, twitter has gained a lot of importance in shaping public opinion. Many politicians and other eminent public figures use the platform to connect with people and many governmental departments have their official twitter handles too.

Plagiarism Scan Report

Summary	
Report Genrated Date	21 Apr, 2018
Plagiarism Status	100% Unique
Total Words	352
Total Characters	2215
Any Ignore Url Used	

Content Checked For Plagiarism:

We propose a system which will analyze twitter reactions to government policies. To get twitter data regarding historical policies, we use the REST API provided by Twitter whereas the data for a recent policy we use the Twitter Streaming API to get tweets. Currently, our system focuses only on the Indian government policies implemented after 2014. For supervised learning, we create a manually labeled tweets dataset.
The block Diagram of the system is specified below:

A user will be able to select a policy from the drop down menu and also period for the analysis if he/she wishes to. We store the tweets in CSV files. We use two different algorithms Support Vector Machines (SVM) and Recurrent Neural networks- Long Term Short Memory (RNN- LSTM) and also provide a comparison between these two methods for

Publication Certificates

Sentiment Analysis of tweets on Government Policies

Mrs Vidya Zope
Assistant Professor
Department of Computer Engineering
Vivekanand Education Society's Institute of Technology
Mumbai, India

Anagha Karmarkar
Department of Computer Engineering,
Vivekanand Education Society's Institute of Technology
Mumbai, India

Vinit Pawar
Department of Computer Engineering
Vivekanand Education Society's Institute of Technology
Mumbai, India

Mansi Shivani
Department of Computer Engineering
Vivekanand Education Society's Institute of Technology
Mumbai, India

Kanchan Tewani
Department of Computer Engineering
Vivekanand Education Society's Institute of Technology
Mumbai, India

***Abstract*— With the advent of social media, it has become possible to tap into public opinion on a large scale. Though other social media giants are trying hard,Twitter remains our best indicator of the wider pulse of the world and what's happening within it. Sentiment classification is an important topic in NLP. In our work we are working on analyzing public reaction to some of the policies and initiatives of the sitting government in India. We are demonstrating a comparative study among supervised learning approaches like Naive Bayes and SVM and a neural network approach using RNN-LSTM. We also discuss about improving our analysis by working on the URLs attached by the users in their tweets. This can provide a powerful insight into the user's opinion as twitter restricts text limit to 280 characters. The experimental results show that RNN, though time consuming to train, works better than the supervised approaches..**

Keywords*— *sentiment analysis, twitter reactions, government policies, social media analysis, RNN, LSTM, neural network, opinion mining

I. Introduction

Sentence classification is a basic task of Natural Language Processing(NLP) that serves as a foundation for many further high level analysis. Social media provides a pool of public opinion. People react to a lot of social subjects that affect them. In recent times, twitter has gained a lot of importance in shaping public opinion. Many politicians and other eminent public figures use the platform to connect with people and many governmental departments have their official twitter handles too.

Traditionally, the only way to tap public opinion was grass root level surveys. These were expensive, time consuming and limited in terms of dataset size. Sentiment analysis of twitter data offers a better way of analyzing public opinion. The tweets are classified as positive, negative or neutral. Many supervised machine learning algorithms can be used for this. However, the advent of deep learning techniques in recent years is providing a greater impetus to such kind of text analysis by way of more accurate predictions.

The words present at the end can be related to the first word. Being able to incorporate this relation can improve a classifier greatly. Neural networks can better imbibe the structural data into the model. In this paper we use datasets pertaining to policies such as digital india, demonetisation, GST, etc. we are comparing various traditional and newer methods such as RNN.

II. Methods For Sentiment Analysis

A dictionary based method is the simplest approach to this problem. A dictionary listing the positive and negative words is created and the individual tweets are separated into their lemmatized words and compared with the dictionary to give a overall sentiment score which helps in classifying the tweet as positive or negative. [3]

A supervised machine learning approach uses algorithms like Support Vector Machines, Logistic Regression, Naïve Bayes to predict the sentiment score. Words in a tweet act as features to these classifiers. Naïve Bayes uses conditional probability to predict the probability of a tweet belonging to a particular class. Support Vector Machines map an input to high dimensional planes using various kernels. [1][4]

However these supervised learning techniques operate on individual words. We may use bigrams, trigrams or n-grams to group two, three or n words together respectively. However, long sequences of words which affect the sentiment cannot be efficiently captured by the above mentioned techniques.

In recent times, deep learning has been used to perform sentiment analysis. The emotion conveyed by a word is influenced by what words come before and after it. In recurrent neural networks, each word corresponds to a particular time step. It has hidden function which contains information about words seen previously. Thus, value of hidden state for a current word input is determined by previous hidden state value and current word vector. This helps preserve relationships between different words separated by distance and often leads to more accurate results. [5][6]

III. Related Work

Cleaning of data is seen to have an impact on accuracy . [1] explores the effect of different preprocessing components like stopword removal, URL removal, reverting words to their original forms, removing numbers, etc on different algorithms.

[2] has used dictionary based approach on a dataset of 500 tweets on the Digital India policy.

Supervised learning has been extensively used on twitter data. [3] have used Naive Bayes algorithm with a trainer's perception. They obtained 90% accuracy with a standard deviation of 14% proving that Naive Bayes isn't weaker than SVM.

Mapping the correlation of events occuring in real time with the opinions being given out on twitter can add an extra level of analysis. [4] extracted sentiment polarity for some major events that occurred during the World Cup and analysed how the positive and negative reaction of people towards such events changed based on incidents during those events.

Supervised algorithms treat words as atomic objects. But, for short text like tweets, it is important to take in account the relationship among them. A number of studies provides a fair comparison among various traditional and deep learning methods. The performance of deep learning methods is affected by dataset size, vanishing and exploding of the gradient, choosing the best feature extractors and classifiers. A lot of research is yet to be done.

As we said, the sequential relationship among words has to be considered while performing sentiment classification. RNN is considered suitable for text sequences. RNN consists of three layers- input, hidden and output layer. The hidden layer has loop structure that allows the information to be persistent. [5] discusses the improvements in RNN by replacing the basic cell with an LSTM cell. It fixes the problem of long term dependence effectively. This paper uses RNN and LSTM to perform sentiment analysis. This model results in better analysis of long sentences used to convey emotions leading to better accuracy.

Correlation among sequences can be established both ways. [7] have put forth deep learning models that combine a convolution layer into a dual recurrent model.

The availability and quality of labeled data is a major issue in sentiment analysis. [6] have made use of unlabelled data to train a set of word embeddings for weight initialization between input and hidden layer of RNN. They have then used labelled data to fine tune the parameters.

IV. Proposed System

We propose a system which will analyze twitter reactions to government policies. To get twitter data regarding recent policies, we use the REST API provided by Twitter whereas for older policies we scrape the tweets of the web. Currently, our system focuses only on the Indian government policies implemented after 2014. For supervised learning, we create a manually labeled tweets dataset.

The block Diagram of the system is specified below:

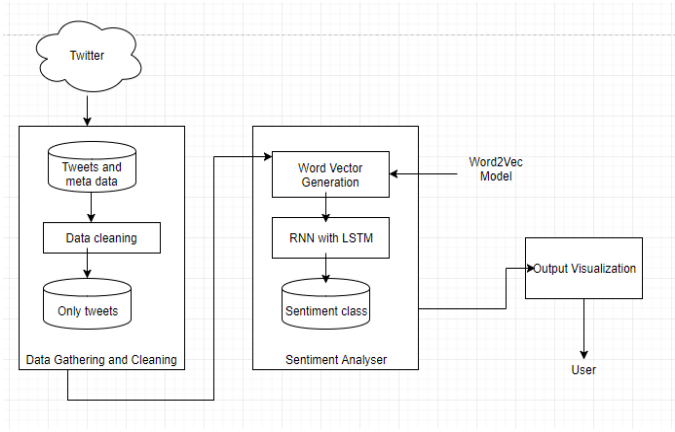


Figure 1.Block Diagram

A user will be able to select a policy from the drop down menu and also period for the analysis if he/she wishes to. We store the tweets in CSV files. We use three different algorithms Support Vector Machines (SVM) , Naive Bayes Classifier and Recurrent Neural networks- Long Term Short Memory (RNN- LSTM) and also provide a comparison between these two methods for sentiment analysis in particular.

A.Data collection:

Data is collected from twitter using different hashtags related to government policies or schemes.For this purpose twitter api tweepy is used but it doesn't provide data which is older than two weeks so twitter scraper is used for getting older data.Also the urls used in tweets are also scrapped and used for analysis purpose.

All the data is received in JSON file which is then converted to csv format, selecting only the attributes relevant to our analysis.

B.Data preprocessing:

Data is cleaned by removing hashtags,special characters, punctuation,replacing @ with User mention,wide spaces with single space and multiple occurrence of same letter with single letter.Emojis play an important role in expressing sentiment therefore for our model to identify it we replace positive emoji with word positive and negative one with word negative.

C.Sentiment analysis method:

For naive bayes and svm algorithm the common steps performed are-

1.For feature generation unigram and bigrams are used.Unigram refers to single word in the dataset and bigrams are group of two words

2.For feature selection TF-IDF is used .It reduces the weight of more common words.

3.The results from step 1 and 2 are passed to naive bayes classifier and svm.

1)Naive Bayes classification-

Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. It is based on a the principle that the value of a particular feature is independent of the value of any other feature, given the class variable.This classification is based on probabilities and independent assumptions between different features. It uses parameter estimation for naive Bayes models uses the method of maximum likelihood.

The following steps were used to implement the algorithm in python

1. Scikit learn library is used
2. Multinomialnb () method is used
3. Laplace smoothing is implemented to prevent assigning zero values to features not present in training data set.
4. After training the model predict() function is used to test the data

2)Support Vector Machine-

An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. It is used to find a hyperplane to categorize the testing dataset.

The following steps were used to implement the algorithm in python:

1. Scikit learn library is used
2. LinearSVC function is used as it is faster to converge as compared to other svm functions
3. l2 penalty is implemented as it gives stable solution and helps in better feature selection
4. Penalty parameter c is set for deciding the margin of hyperplane
5. After training the model ,it is tested with using predict () function.

3)RNN with LSTM -

Recurrent neural network (RNN) is used to improve working of neural networks.All the achievement maybe due to its importation feature called recursion, through which the network can be implicitly deeper than traditional neural network.The recursion feature is able to compress the arbitrary long windows history into a fixed sized hidden layer and then recorded history can help make improved result.present. Using LSTM replaces RNN node in hidden layer with LSTM cell, which is designed to save the text history information. LSTM uses three gates to control the usage and update of the text history information, which are input gates, forget gates and output gates respectively. The memory cell and three gates are designed to enable LSTM to read, save and update long-distance history information.

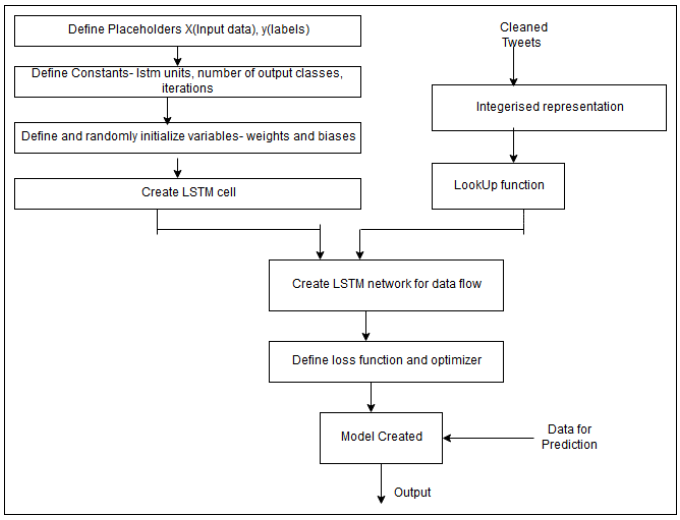


Figure 2.Flow Chart

Following steps were followed-

1. Download pre trained word embedding model Glove(4,00,000 x 100)

2. Split the cleaned tweets to tokens and get the dimension vector for each word. Repeat this for all the tweets to get a final input 3D matrix
3. Specify parameters of the network, unroll the lstm cells to create the final graph
4. Adjust weights to minimize loss and predicts whether sentiment is positive or negative and calculate accuracy.

V. Results And Analysis

A.Accuracy

The RNN-LSTM algorithm gave a accuracy of 88.33%.
The Naive Bayes algorithm gave an accuracy of 84.167%.
The Support Vector Machine algorithm gave an accuracy of 81.67%

B. Visualizations

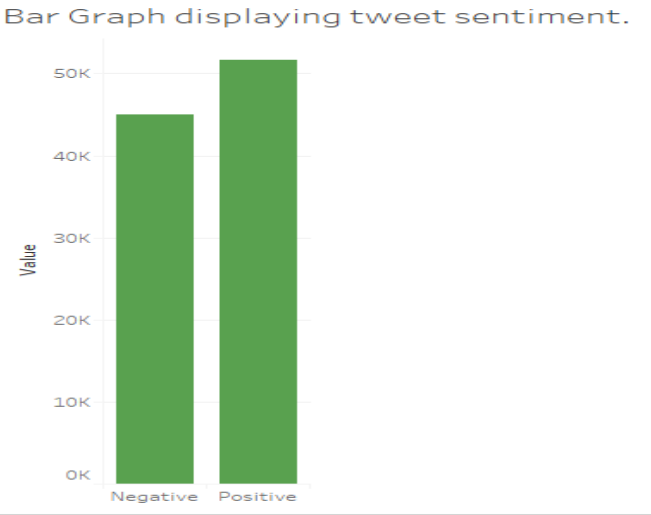


Figure 3.Bar Graph

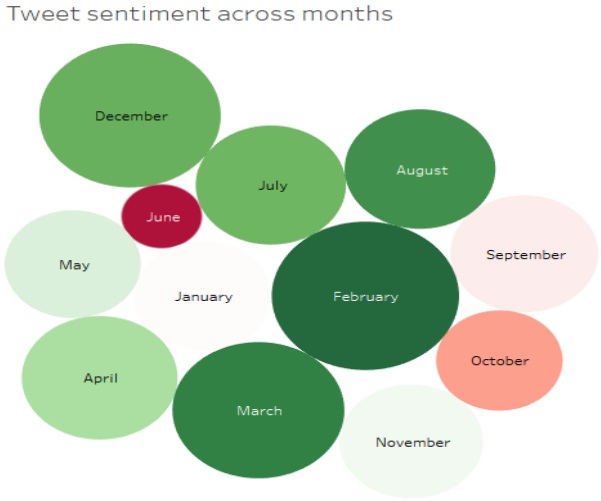


Figure 4.Month wise Sentiment distribution

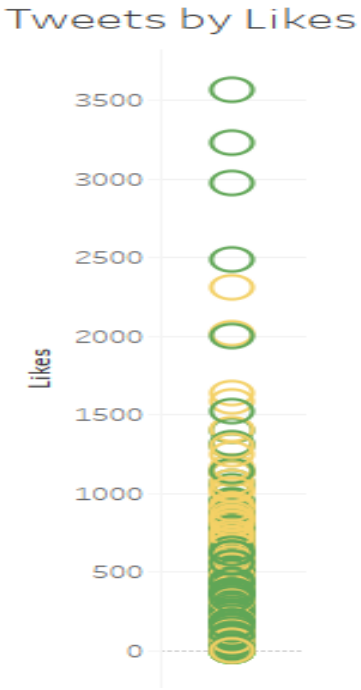


Figure 5.Analysis using metadata

VI. Conclusion And Future Scope

Tweets provide a honest insight into the public opinion. The policymakers can make use of these insights to plan the future. But, the abundant data available has to be first organized and analyzed. The task of of labeling can be automated using our model with good accuracy. Using a powerful visualization tool like Tableau helped in exploiting the metadata available such as number of likes, number of retweets, timestamp, etc. We can also improve our analysis by working on the URLs attached by the users in their tweets. This can provide a powerful insight into the user's opinion as twitter restricts text limit to 280 characters. A scoring algorithm can be added to the model . The top 5 tweets having an URL cited for each class can then be considered for summarization purpose. These articles can be summarized and used for further analytical purposes.

References

[1] Zhao Jianqiang "Pre-processing Boosting Twitter Sentiment Analysis," Learning, 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), INSPEC Accession Number. 15986223

[2] Prem Mishra, Ranjana Ranjish, Pankaj Kumar, "Sentiment analysis of Twitter data: Case study on digital India," Information Technology (InCITe) - The Next Generation IT Summit on the Theme - Internet of

Things: Connect your Worlds, International Conference on 6-7 Oct. 2016, INSPEC Accession Number. 16674097

[3] Mohd Naim Mohd Ibrahim, Mohd Zaliman Mohd Yusoff "Twitter sentiment classification using Naive Bayes based on trainer perception," Learning, e-Management and e-Services (IC3e), 2015 IEEE Conference on 24-26 Aug. 2015, INSPEC Accession Number. 15787751

[4] Peiman Barnaghi, Parsa Ghaffari, John G. Breslin, "Opinion Mining and Sentiment Polarity on Twitter and Correlation between Events and Sentiment," Big Data Computing Service and Applications (BigDataService), 2016 IEEE Second International Conference on 29 March-1 April 2016, INSPEC Accession Number. 16022629

[5] Dan Li, Jiang Qian, "2016 First IEEE International Conference on Computer Communication and the Internet (ICCCI)," INSPEC Accession Number 16540645.

[6] Qurat Tul Ain, Mubashir Ali, Amna Riaz, Amna Noureen, Muhammad Kamran, Babar Hayat and A. Rehman, "Sentiment Analysis Using Deep Learning Techniques: A Review,". International Journal of Advanced Computer Science and Applications, Vol. 8, No. 6, 2017

[7] Yong Zhang, Meng Joo Er, Rajasekar Venkatesan, Ning Wang and Mahardhika Pratama, "Sentiment Classification Using Comprehensive Attention Recurrent Models," Neural Networks (IJCNN), 2016 International Joint Conference on 24-29 July 2016, INSPEC Accession Number16446844 .

Plagiarism Report

Plagiarism Scan Report

Summary	
Report Genrated Date	21 Apr, 2018
Plagiarism Status	100% Unique
Total Words	644
Total Characters	4128
Any Ignore Url Used	

Content Checked For Plagiarism:

Abstract— With the advent of social media, it has become possible to tap into public opinion on a large scale. Though other social media giants are trying hard,Twitter remains our best indicator of the wider pulse of the world and what's happening within it. Sentiment classification is an important topic in NLP. In our work we are working on analyzing public reaction to some of the policies and initiatives of the sitting government in India. We are demonstrating a comparative study among supervised learning approaches like Naive Bayes and SVM and a neural network approach using RNN-LSTM. We also discuss about improving our analysis by working on the URLs attached by the users in their tweets. This can provide a powerful insight into the user's opinion as twitter restricts text limit to 280 characters. The experimental results show that RNN, though time consuming to train, works better than the supervised approaches..
Keywords— sentiment analysis, twitter reactions, government policies, social media analysis, RNN, LSTM, neural network, opinion mining
Introduction
Sentence classification is a basic task of Natural Language Processing(NLP) that serves as a foundation for many further high level analysis. Social media provides a pool of public opinion. People react to a lot of social subjects that affect them. In recent times, twitter has gained a lot of importance in shaping public opinion. Many politicians and other eminent public figures use the platform to connect with people and many governmental departments have their official twitter handles too.
Traditionally, the only way to tap public opinion was grass root level surveys. These were expensive, time consuming and limited in terms of dataset size. Sentiment analysis of twitter data offers a better way of analyzing public opinion. The tweets are classified as positive, negative or neutral. Many supervised machine learning algorithms can be used for this. However, the advent of deep learning techniques in recent years is providing a greater impetus to such kind of text analysis by way of more accurate predictions.
The words present at the end can be related to the first word. Being able to incorporate this relation can improve a classifier greatly. Neural networks can better imbibe the structural data into the model. In this paper we use datasets pertaining to policies such as digital india, demonetisation, GST, etc. we are comparing various traditional and newer methods such as RNN.

Methods For Sentiment Analysis
A dictionary based method is the simplest approach to this problem. A dictionary listing the positive and negative words is created and the individual tweets are separated into their lemmatized words and compared with the dictionary to give a overall sentiment score

Plagiarism Scan Report

Summary	
Report Genrated Date	21 Apr, 2018
Plagiarism Status	100% Unique
Total Words	396
Total Characters	2472
Any Ignore Url Used	

Content Checked For Plagiarism:

3)RNN with LSTM -
Recurrent neural network (RNN) is used to improve working of neural networks.All the achievement maybe due to its importation feature called recursion, through which the network can be implicitly deeper than traditional neural network.The recursion feature is able to compress the arbitrary long windows history into a fixed sized hidden layer and then recorded history can help make improved result.present. Using LSTM replaces RNN node in hidden layer with LSTM cell, which is designed to save the text history information. LSTM uses three gates to control the usage and update of the text history information, which are input gates, forget gates and output gates respectively. The memory cell and three gates are designed to enable LSTM to read, save and update long-distance history information.
Figure 2.Flow Chart