Sentiment Analysis of Twitter Data using NLTK in Python

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Technology

in

Computer Science and Applications

Submitted By

Prateek Garg

(Roll No. 601403019)

Under the supervision of:

Vineeta Bassi

Assistant Professor

(CSED)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

JUNE 2016

Certificate

I hereby certify that the work which is being presented in the thesis entitled, ***"Sentiment Analysis of Twitter Data using NLTK in Python"*** , in partial fulfillment of the requirements for the award of degree of Master of Technology in ***Computer Science and Applications*** submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of ***Ms. Vineeta Bassi*** and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Prateek Garg

(Prateek Garg)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Vineeta Bassi

(Vineeta Bassi)
Assistant Professor
Computer Science and Engineering Department
Thapar University
Patiala

Countersigned by

(Dr. Maninder Singh)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S. S. Bhatia)
Dean (Academic Affairs)
Thapar University
Patiala

Acknowledgement

First of all, I would like to express my thanks to my guide **Ms. Vineeta Bassi** Assistant Professor, Computer Science and Engineering Department, Thapar University, Patiala for being an excellent mentor for me during my whole course of thesis. Her encouragement and valuable advice during the entire period has made it possible for me to complete my work.

I am thankful to **Dr. Maninder Singh**, Head of Computer Science Engineering Department, Thapar University for setting high standards for his students and encouraging them time to time so that they can achieve them as well. I would also like to give my regards to **Dr. Sanmeet Bhatia**, P.G. Coordinator, Computer Science and Applications, Thapar University for the motivation and inspiration in this journey. I would also like to thank entire faculty and staff of Computer Science and Engineering Department and my friends who devoted their valuable time in completion of this work.

Lastly, I would like to thank my parents for their years of unyielding love and encourage. They have wanted the best for me and I admire their sacrifice and determination.

Prateek Garg

(601403019)

Abstract

In today's world, Social Networking website like Twitter, Facebook, Tumbler, etc. plays a very significant role. Twitter is a micro-blogging platform which provides a tremendous amount of data which can be used for various applications of Sentiment Analysis like predictions, reviews, elections, marketing, etc. Sentiment Analysis is a process of extracting information from large amount of data, and classifies them into different classes called sentiments.

Python is simple yet powerful, high-level, interpreted and dynamic programming language, which is well known for its functionality of processing natural language data by using NLTK (Natural Language Toolkit). NLTK is a library of python, which provides a base for building programs and classification of data. NLTK also provide graphical demonstration for representing various results or trends and it also provide sample data to train and test various classifiers respectively.

The goal of this thesis is to classify twitter data into sentiments (positive or negative) by using different supervised machine learning classifiers on data collected for different Indian political parties and to show which political party is performing best for public. We also concluded which classifier gives more accuracy during classification.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

NLTK: Natural Language Toolkit

NLP: Natural Language Processing

NB: Naïve-Bayes

SVM: Support Vector Machines

MAP: Maximum A Posterior

OvR: One-vs-Rest

OvA: One-vs-All

SGDC: Stochastic Gradient Decent Classifier

SVC: Support Vector Classifier

BJP: Bharatiya Janta Paty

AAP: Aam Aadmi Party

INC: Indian National Congress

WOM: World of Mouth

VOM: Voice of the Market

BRM: Brand Reputation Management

API: Application programming Interface

CSV: Comma Separated Values

URL: Uniform Resource locator

tf-idf: term frequency-inverse document frequency

Chapter 1

Introduction

In this chapter we are going to give the introductions on Sentiment Analysis, Python and Natural Language Toolkit (NLTK). Then we are explaining the objective of our thesis. After this we will discuss why there is a need of sentiment analysis and some of the applications of Sentiment Analysis which are used in our daily life.

1.1 Introduction to Sentiment Analysis

Sentiment Analysis is process of collecting and analyzing data based upon the person feelings, reviews and thoughts. Sentimental analysis often called as opinion mining as it mines the important feature from people opinions. Sentimental Analysis is done by using various machine learning techniques, statistical models and Natural Language Processing (NLP) for the extraction of feature from a large data.

Sentiment Analysis can be done at document, phrase and sentence level. In document level, summary of the entire document is taken first and then it is analyze whether the sentiment is positive, negative or neutral. In phrase level, analysis of phrases in a sentence is taken in account to check the polarity. In Sentence level, each sentence is classified in a particular class to provide the sentiment.

Sentimental Analysis has various applications. It is used to generate opinions for people of social media by analyzing their feelings or thoughts which they provide in form of text. Sentiment Analysis is domain centered, i.e. results of one domain cannot be applied to other domain. Sentimental Analysis is used in many real life scenarios, to get reviews about any product or movies, to get the financial report of any company, for predictions or marketing.

Twitter is a micro blogging platform where anyone can read or write short form of message which is called tweets. The amount of data accumulated on twitter is very huge. This data is unstructured and written in natural language. Twitter Sentimental Analysis is the process of accessing tweets for a particular topic and predicts the

sentiment of these tweets as positive, negative or neutral with the help of different machine learning algorithm.

1.2 Introduction to Python

Python is a high level, dynamic programming language which is used for this thesis. Python3.4 version was used as it is a mature, versatile and robust programming language. It is an interpreted language which makes the testing and debugging extremely quickly as there is no compilation step. There are extensive open source libraries available for this version of python and a large community of users.

Python is simple yet powerful, interpreted and dynamic programming language, which is well known for its functionality of processing natural language data, i.e. spoken English using NLTK. Other high level programming languages such as 'R' and 'Matlab' were considered because they have many benefits such as ease of use but they do not offer the same flexibility and freedom that Python can deliver.

1.3 Introduction to NLTK

Natural Language Toolkit (NLTK) is library in Python, which provides a base for building programs and classification of data. NLTK is a collection of resources for Python that can be used for text processing, classification, tagging and tokenization. This toolbox plays a key role in transforming the text data in the tweets into a format that can be used to extract sentiment from them.

NLTK provides various functions which are used in pre-processing of data so that data available from twitter become fit for mining and extracting features. NLTK support various machine learning algorithms which are used for training classifier and to calculate the accuracy of different classifier.

In our thesis we use Python as our base programming language which is used for writing code snippets. NLTK is a library of Python which plays a very important role in converting natural language text to a sentiment either positive or negative. NLTK also provides different sets of data which are used for training classifiers. These datasets are structured and stored in library of NLTK, which can be accessed easily with the help of Python.

1.4 Introduction to Supervised Machine learning Classifiers

Supervised machine learning is a technique whose task is to deduce a function from tagged training samples. The training samples for supervised learning consist of large set of examples for a particular topic. In supervised learning, every example training data comes in a pair of input (vector quantity) and output value (desired result). These algorithms analyze data and generate an output function, which is used to mapped new data sets to respective classes. Different machine learning classifiers which we are going to use to build our classifier are:

- ∑ Naïve-Bayes Classifier
- ∑ MultinomialNB Classifier
- ∑ BernoulliNB Classifier
- ∑ Logistic Regression Classifier
- ∑ SGDC (Stochastic Gradient Decent Classifier)
- ∑ SVC (Support Vector Classifier): LinearSVC and NuSVC

1.4.1 Naïve-Bayes (NB) Classifier [1]

Naïve-Bayes classifiers are probabilistic classifiers which come under machine learning techniques. These classifiers are based on applying Bayes' theorem with strong (naïve) assumption of independence between each pair of features. Let us assume, there is a dependent vector from x_1 to x_n, and a class variable 'y'. Therefore, according to Bayes' :

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)P(x_1, \ldots x_n \mid y)}{P(x_1, \ldots, x_n)}$$

Now according to assumption of independence

$$P(x_i|y, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) = P(x_i|y),$$

For every 'i', this function becomes

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y) \prod_{i=1}^{n} P(x_i \mid y)}{P(x_1, \ldots, x_n)}$$

In this $P(x_1, \ldots, x_n)$ on given input is constant, hence we can apply classification rule as:

$$P(y \mid x_1, \ldots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

$$\Downarrow$$

$$\hat{y} = \arg\max_y P(y) \prod_{i=1}^{n} P(x_i \mid y),$$

And for estimating we can use MAP (Maximum A Posterior) estimation $P(y)$ and $P(x_i \mid y)$; the $P(y)$ of class 'y' in training sample is relative frequency.

1.4.2 MultinomialNB Classifier

MultinomialNB expands the use of NB algorithm. It implements NB for data distributed multinomially, and also uses one of its version for text classification (in which word counts are used to represent data, and also tf-idf works extremely well in regular practice). We parameterized the distribution data by vectors $\theta_y = (\theta_{y1}, \ldots, \theta_{yn})$ for every$y$, where 'n' gives the total features (which means, the size of vocabulary for text classification) and probability $P(x_i \mid y)$ of each i that appears in the sample of class 'y' is θ_{yi}

We use smoothed version of maximum likelihood for estimation of parametersθ_y, which is relative frequency of counting:

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n}$$

where N_{yi} represents number of times 'i' appeared in any sample of class 'y' which belongs to training sample T, and $N_y = \sum_{i=1}^{|T|} N_{yi}$ gives the total number of

features in class ‘y’. To prevent zero probabilities for further calculations, we add smoothing priors’ $\alpha \geq 0$ for features that are not present in any learning samples. If $\alpha = 1$, smoothing is termed as Laplace and for $\alpha < 1$ the smoothing is termed as Lidstone.

1.4.3 BernoulliNB Classifier [2][3]

BernoulliNB also implements NB algorithm for training and classification. It use NB for multivariate Bernoulli distribution of data; i.e., there can be many features but each and every one is assumed to have a binary value or Boolean (true or false) variable. Hence, every class requires samples which have to be represented in binary value variables; also if any other kind of data is given then BernoulliNB can binaries its input.

The BernoulliNB decision rule is explained as:

$$P(x_i \mid y) = P(i \mid y)x_i + (1 - P(i \mid y))(1 - x_i)$$

which is different from MultinomialNB’s rule, this rule directly punish any unavailability of feature i which behave like a feature of class y where as in multinomial it simply ignore if there is any non-occuring feature.

1.4.4 Logistic Regression Classifier [4]

Despite its name Logistic regression, is not a regression model but a linear model for classification. This model is also known by other names as Maximum-Entropy (MaxEnt) classification or log-linear classifier. A logistic function is used in this model, where probability describe the outcome of single trial.

The logistic regression can be implemented from Scikit-learn library of Python in which there is a class named LogisticRegression. This implementation fits a OvR (one-vs-rest) multiclass regression with an optional L1 or L2 regularization.

L2 penalized logistic regression helps in minimizing the following cost function:

$$\min_{w,c} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \log(\exp(-y_i(X_i^T w + c)) + 1).$$

Similarly, L1 regularized logistic regression can solve following problem of optimization:

$$\min_{w,c} \|w\|_1 + C \sum_{i=1}^{n} \log(\exp(-y_i(X_i^T w + c)) + 1).$$

1.4.5 SGDC (Stochastic Gradient Decent Classifier) [5]

SGD is a simple yet powerful and efficient approach for learning of classifiers that comes under convex loss functions such as SVM and Logistic Regression. SGDC combines multiple binary classifiers in OvA (One-vs-All) method. Therefore it supports multi-class classification. During testing phase, we also calculate confidence score for each and evey classifier and thus choosing the class with the highest score. Figure 1.1 shows the OvA approach on 'iris dataset' (sample). In this figure dashed lines represent the OVA classifiers and the background colors show decision surface that are included in classifer.

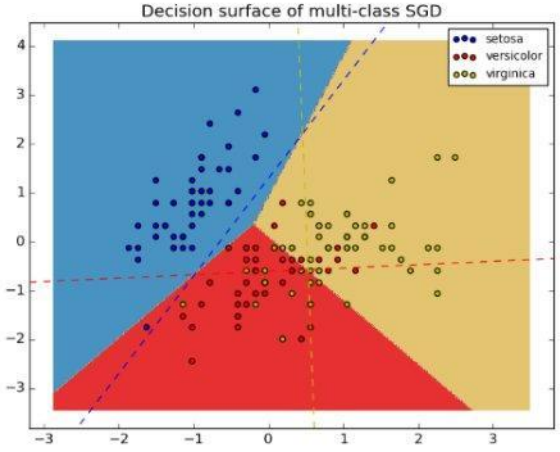


Figure 1.1 OVA approach on iris dataset [6]

In multi-class, classification 'coef' is a 2-D array of shape = [classes, features] and 'intercept' is a 1-D array of shape = [classes] only. The i^{th} row of coef matix contains

the weight quantity for OvA classifier of the i^{th} class. Also, classes are arranged in increasing order.

1.4.6 SVC (Support Vector Classifier): LinearSVC and NuSVC [6]

SVM are supervised machine learning methods used for classification, regression and detection models. SVM are more effective for high dimensional space. SVCs are capable for multi-class classification. SVC and NuSVC are similar whereas, LinearSVC are based on linear kernels.

All these SVCs take two input array: an array X of size [samples, features] and array Y of size [samples]. NuSVC implements ‘one-against-once’ scheme for multi-class, hence it provides consistent interface with other classifiers. Whereas, LinearSVC implement ‘one-vs-rest’ scheme

NuSVC implementation is based on ‘libsvm’ library, whereas LinearSVC implementation is based on ‘liblinear’ library. A SVM classification, regression and other tasks are done with the help of hyperplanes. These hyper-planes or set of hyper-planes are contructes in high dimensional space. Thus, from hyper-planes we can understand, a good separation is achieved by those that have the maximum distance to the nearest data points of any class which is called functional margin. It is concluded that larger the margin lower the generalization error of multiclass classifier. An example of hyper-plne use is shown in Figure 1.2

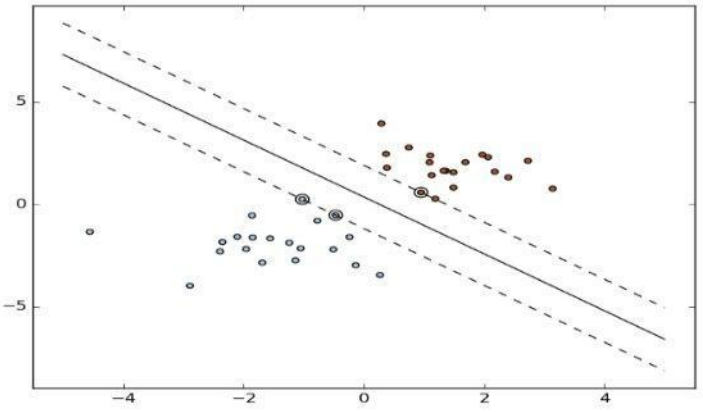


Figure 1.2 Hyper-plane used for classification in SVM [7]

1.5 Goal of Thesis

With the emergence of social networking, many websites have evolved in the past decade like Twitter, Facebook, Tumbler, etc. Twitter is one the website which is widely used all over the world. According to Twitter it has been recorded that around 200 billion tweets posts every year. Twitter allows people to express their thoughts, feelings, emotions, opinions, reviews, etc. about any topic in natural language within 140 characters. Python is the standard high-level programming language which is best for NLP. Thus, for processing natural language data, Python uses one of its libraries called Natural Language Toolkit. NLTK provides large amount of corpora which helps in training classifiers and it helps in performing all NLP methodology like tokenizing, part-of-speech tagging, stemming, lemmatizing, parsing and performing sentiment analysis for given datasets.

It is a challenging task to deal with a large dataset, but with the use of NLTK we can easily classify our data and give more accurate results based on different classifiers. The goal of this thesis is to perform sentiment analysis on different Indian Political Parties, like BJP (Bharatiya Janta Party), AAP (Aam Aadmi Party), INC (Indian National Congress). Public opinions of these parties are mined from Twitter and then classified into sentiments, whether positive or negative by using supervised machine learning classifiers. These results will let us know about the reviews and opinions of people on these political parties.

To achieve this goal, a module is created which can perform live sentimental analysis. In live sentimental analysis user can obtain the trend of any live trending topic depicted by two sentiment category (positive and negative) in live graphs. Further accuracy and reliability of the module can be checked with the help of various machine learning classifiers.

To many companies and organizations a customer's perception of a product or service is extremely valuable information. From the knowledge gained from an analysis such as this a company can identify issues with their products, spot trends before their competitors, create improved communications with their target audience, and gain valuable insight into how effective their marketing campaigns were. Through this

knowledge companies gain valuable feedback which allows them to further develop the next generation of their product.

In this thesis we work on different political parties because in our country politics plays a very vital role. Winning an election by any party is different from how that party works after winning.

In the context of the sentiment analysis being carried out for this application, the results will allow user to gain insight into how each party is being perceived by the public. This is very valuable information as public is uploading their expectations, opinions and views on the political parties. This really revolutionizes the feedback process. An application such as this has the potential to analyze the sentiment in real time giving the users immediate feedback on how a party is being help in the eyes of its audience. Such an application could be expanded to use clustering algorithms to give insight into particular member or position.

1.6 Need of Sentimental Analysis

1.6.1 Industry Evolution

Only the useful amount of data is required in the industry as compared to the set of complete unstructured form of the data. However the sentiment analysis done is useful for extracting the important feature from the data that will be needed solely for the purpose of industry. Sentimental Analysis will provide a great opportunity to the industries for providing value to their gain value and audience for themselves. Any of the industries with the business to consumer will get benefit from this whether it is restaurants, entertainment, hospitality, mobile customer, retail or being travel.

1.6.2 Research Demand

Another important reason that stands behind the growth of SA deals with the demand of research in evaluation, appraisals, opinion and their classification. Present solutions for the purpose of sentiment analysis and opinion mining are rapidly evolving, specifically by decreasing the amount of human effort that will be required to classify the comments. Also the research theme that will be based in the long established disciplines of computer science like as text mining, machine learning, natural language processing and artificial intelligence, voting advise applications, automated content analysis, etc.

1.6.3 Decision Making

Every person who stores information on the blogs, various web applications and the web social media, social websites for getting the relevant information you need a particular method that can be used to analyze data and consequently return some of the useful results. It is going to be very difficult for company to conduct the survey that will be on the regular basis so that there comes the need to analyze the data and locate the best of the products that will be based on user's opinions, reviews and advices. The reviews and the opinions also help the people to take important decisions helping them in research and business areas.

1.6.4 Understanding Contextual

As human language is getting very complex day by day so it has become difficult for the machine to be able to understand human language that can be expressed in the slangs, misspelling, nuances, and the cultural variation. Thus, there will be a need of system that will make better understanding between the human and the machine language.

1.6.5 Internet Marketing

Another important reason behind the increase in the demand of sentimental analysis is the marketing done via internet by the business and companies organization. Now they regularly monitor the opinion of the user about their brand, product, or event on blog or the social post. Thus, we see that the sentimental Analysis could also work as a tool for marketing too.

1.7 Applications of Sentiment Analysis

Sentiment analysis has large amount of applications in the NLP domain. Due to the increase in the sentiment analysis, social network data is on high demand. Many companies have already adopted the sentimental analysis for the process of betterment. Some of major applications are mentioned as following:

1.7.1 Word of Mouth (WOM)

Word of Mouth (WOM) is the process by which the information is given from one

person to another person. It would essentially help the people to take the decisions. Word of Mouth has given the information about the opinions, attitudes, reactions of consumers about the related business, services and the products or even the ones that can be shared with more than one person. Therefore, this is going to be where Sentiment Analysis comes into picture. As the online review blogs, sites, social networking sites have provided the large amount of opinions, it has helped in the process of decision-making so much easier for the user.

1.7.2 Voice of Voters

Each of the political parties usually spent a major chunk of the amount of money for the aim of campaigning for their party or for influencing the voters. Thus if the politicians know the people opinions, reviews, suggestions, these can be done with more effect. This is how process of Sentimental analysis does not only help political parties but on the other hand help the news analysts alongside. Also the British and the American administration had already used some of the similar techniques.

1.7.3 Online Commerce

There is vast number of websites related to ecommerce. Majority of them had the policy of getting the feedback from its users and customers. After getting information from various areas like service and quality details of the users of company users experience about features, product and any suggestions. These details and reviews have been collected by company and conversion of data into the geographical form with the updates of the recent online commerce websites who use these current techniques.

1.7.4 Voice of the Market (VOM)

Whenever a product is to be launched by a specific company, the customers would to know about the product ratings, reviews and detailed descriptions about it. Sentiment Analysis can help in analyzing marketing, advertising and for making new strategies for promoting the product. It provides the customer an opportunity to choose the best among the all.

1.7.5 Brand Reputation Management (BRM)

Sentiment analysis would help to determine how would be a company’s brand,

service and the service or product that would be perceived by the online community. Brand Reputation Management will be concerned about the management of the reputation of market. It has focuses on the company and product rather than customer. Thus the opportunities were created for the purpose of managing and strengthening the brand reputation of the organizations.

1.7.6 Government

Sentiment Analysis has helped the administration for the purpose of providing various services to the public. Fair results have to be generated for analyzing the negative and positive points of government. Thus sentiment analysis is helpful in many fields like decision making policies, recruitments, taxation and evaluating social strategies. Some of the similar techniques that provide the citizen oriented government model where the services and the priorities should be provided as per the citizens. One of the interesting problems which can be taken up is applying this method in the multi-lingual country like the India where content of the generating mixture of the different languages (e.g. Bengali English) is a very common practice.

Chapter 2

Literature Review

Many research have been done on the subject of sentiment analysis in past. Latest research in this area is to perform sentiment analysis on data generated by user from many social networking websites like Facebook, Twitter, Amazon, etc. Mostly research on sentiment analysis depend on machine learning algorithms, whose main focus is to find whether given text is in favor or against and to identify polarity of text. In this chapter we will provide insight of some of the research work which helps us to understand the topic deep.

P. Pang, L. Lee, S. Vaithyanathan *et al* [8]

They were the first to work on sentiment analysis. Their main aim was to classify text by overall sentiment, not just by topic e.g., classifying movie review either positive or negative. They apply machine learning algorithm on movie review database which results that these algorithms out-perform human produced algorithms. The machine learning algorithms they use are Naïve-Bayes, maximum entropy, and support vector machines. They also conclude by examining various factors that classification of sentiment is very challenging. They show supervised machine learning algorithms are the base for sentiment analysis.

P. Pang, L. Lee *et al* [9]

By collecting large amount of data has always been a key to find out what people is thinking or expecting. With the emergence in the field of social media, availability of data which is full of opinion resources is very high. Other resources such as blogs, review sites, messages, etc. are helping us to know what people can do and their opinion about the topic. The sudden increase of work in the field of data mining and sentiment extraction deals with the computational power to solve the problem of opinion mining or subjectivity in text. Hence various new systems are created based on different languages and commands that can deal directly with opinion mining as the first class object and direct response or live research also becoming the area of interest.

They take a survey which covers that methodology and approaches that are used in direct response of opinion mining are more helpful than others. Their focus is on functions that can solve new challenges rising in sentiment analysis applications. They also compared these new techniques to already present traditional analysis which is based on facts.

E. Loper, S. Bird *et al* [10]

Natural Language Toolkit (NLTK) is a library which consists of many program modules, large set of structured files, various tutorials, problem sets, many statistics functions, ready-to-use machine learning classifiers, computational linguistics courseware, etc. The main purpose of NLTK is to carry out natural language processing, i.e. to perform analysis on human language data. NLTK provides corpora which are used for training classifiers. Developers create new components and replace them with existing component, more structured programs are created and more sophisticated results are given by dataset.

H. Wang, D. Can, F. Bar, S. Narayana *et al* [11]

They were the researchers who proposed a system for real time analysis of public responses for 2012 presidential elections in U.S. They collect the responses from Twitter, a micro blogging platform. Twitter is one the social network site where people share their views, thoughts and opinions on any trending topic. People responses on Twitter for election candidates in U.S. created a large amount of data, which helps to create a sentiment for each candidate and also created a prediction of whom winning.

A relation is created between sentiments that arise from people response on twitter with the complete election events. They also explore how sentiment analysis affects these public events. They also show this live sentiment analysis is very fast as compared to traditional content analysis which takes many days or up to some weeks to complete. The system they demonstrated analyzes sentiment of entire Twitter data about the election, candidates, promotions, etc. and delivering results at a continuous

rate. It offers media, politicians and researchers a new way which is timely effective which is completely based on public opinion.

O. Almatrafi, S. Parack, B. Chavan *et al* [12]

They are the researchers who proposed a system based on location. According to them, Sentiment Analysis is carried out by Natural Language Processing (NLP) and machine learning algorithms to extract a sentiment from a text unit which is from a particular location. They study various applications of location based sentiment analysis by using a data source in which data can be extracted from different locations easily. In Twitter, there is field of tweet location which can easily be accessed by a script and hence data (tweets) from particular location can be collected for identifying trends and patterns.

In their research they work on Indian general elections 2014. They perform mining on 600,000 tweets which were collected over a period of 7 days for two political parties. They apply supervised machine learning approach, like Naïve-Bayes algorithm to build a classifier which can classify the tweets in either positive or negative. They identify the thoughts and opinions of users towards these two political parties in different locations and they plot their finding on India map by using a Python library. An example of their results on tweets of BJP in 2014 is shown in Figure 2.1, which shows different locations in India where BJP got positive reviews.

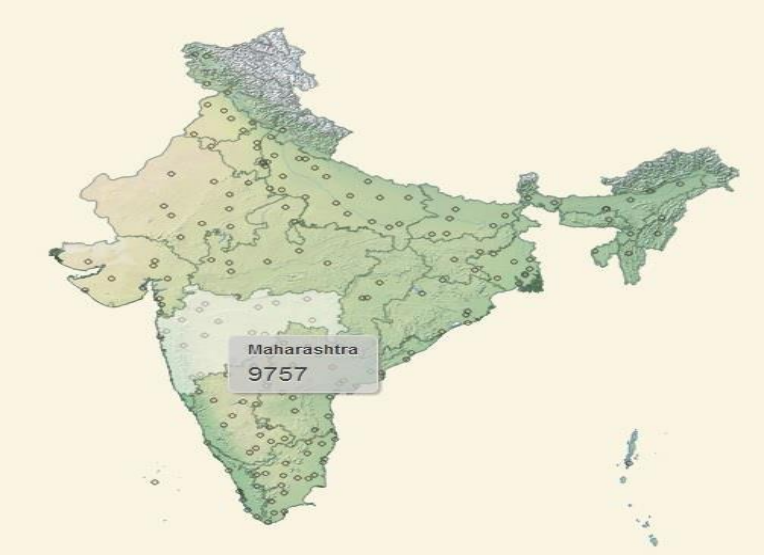


Figure 2.1 Positive tweets of BJP for different states in 2014 [12]

L. Jiang, M. Yu, M. Zhou, X. Liu, T. Zhao *et al* [13]

Twitter sentiment analysis was growing at faster rate as amount of data is increasing. They created a system which focuses on target dependent classification. It is based on Twitter in which a query is given first; they classify the tweets as positive, negative or neutral sentiments with respect to that query that contain sentiment as positive, negative or neutral. In their research, query sentiment serves as target. The target-independent strategy is always adopted to solve these problems with the help of state-of-the-art approaches, which may sometime assign immaterial sentiments to target. Also, when state-of-the-art approaches are used for classification they only take tweet into consideration. These approaches ignore related tweet, as they classify based on current tweet.

However, because tweets have property to be short and mostly ambiguous, considering current tweet only for sentiment analysis is not enough. They propose a system to improve target-dependent Twitter sentiment classification by:

1) Integrating target-dependent features, and

2) Taking related tweets into consideration.

According to their experimental results, these new advancement highly improves the efficiency and performance of target-dependent sentiment classification.

C. Tan, L. Lee, J. Tang, L. Jiang, M. Zhou, P. Li, *et al* [14]

They show that information that can be used to improve user-level sentiment analysis. There base of research is social relationships, i.e. users that are connected in any social platform will somehow hold similar opinions, thoughts; therefore, relationship information can supplement what they extract from user's viewpoint. They use Twitter as there source of experimental data and they use semi-supervised machine learning framework to carry out analysis. They propose systems that are persuade either from the network of Twitter followers or from the network formed by users in Twitter in which users referring to each other using "@username". According to them, these semi-supervised learning results shows that by including this social-network information leads to statistically significant improvement in performance of sentiment analysis classification over the performance based on the approach of SVM (Support Vector Machines) that have only access to textual features.

A. Pak, P. Paroubek *et al* [15]

Micro-blogging nowadays has become very popular communication platform among users in social network. Billions of tweets share every year among millions of users, in which they share opinions, feelings on different aspects of daily life. Thus micro-blogging websites like Twitter, Friendfeed, Tumbler, etc. are rich sources of data for feature extraction and sentiment analysis. They also use Twitter one of the most popular micro-blogging website, for the implementation of sentiment analysis. They automatically collect a corpus (database) for training classifier to carry out sentiment analysis and opinion mining.

They perform linguistic inspection of collected corpus and build a sentiment classifier that is used to determine positive, negative and neutral sentiments of twitter document. They also proposed a system for emoticons used in tweets, in which they create a corpus for emoticons such that they can replace each emoticon with their respective meaning so that it can extract feature from emoticons. An example of emoticons is shown is Table 2.1, experimental calculations show that their proposed techniques are more efficient and give more performance than previous proposed models.

Table 2.1 Example of emoticons and their corresponding meanings

Emoticons	Meanings
☺, :D, =), =D, ;-)	Happy
;-(, ;(, =(, =[,)-:	Sad
:P, =P	Joking

B. Sun, V. Ng, *et al* [16]

Many efforts have been done to gather information from social networks to perform sentiment analysis on internet users. Their aim is to show how sentimental analysis influences from social network posts and they also compare the result on various topics on different social-media platforms. Large amount of data is generated every day, people are also very curious in finding other similar people among them. Many researchers' measures the influence of any post through the number of likes and

replies it received but they are not sure whether the influence is positive or negative on other post. In their research some questions are raised and new methodologies are prepared for sentimental influence of post.

Chapter 3

Problem Statement

Sentiment Analysis is a process of extracting feature from user's thoughts, views, feelings and opinions which they post on any social network websites. The result of sentiment analysis is classification of natural language text into classes such as positive, negative and neutral. The amount of data generated from social network sites is huge; this data is unstructured and cannot give any meaningful information until it is analyzed. Thus, to make this huge amount of data useful we perform sentiment analysis, i.e. extracting feature from this data and classify them. Sentiment analysis is very necessary in today's world, as people always get affected by the thinking and opinions other people. Today, if any one wants to purchase a product or to give vote or to watch a movie, etc. then that person will first wants to know what are other people reviews, reactions and opinions about that product or candidate or movie on social media websites like Twitter, Facebook, Tumbler, etc. So there is a need of system that can automatically generate sentiment analysis from this huge amount of data.

3.1 Objectives

The main objective of this thesis work is to perform the sentiment analysis on Indian Political Parties like BJP, INC and AAP, such that people opinions about these parties progress, workers, policies, etc. which are extracted from Twitter.

Thus to achieve this objective we build a classifier based on supervised learning and perform live sentiment analysis on data collected of different political parties.

3.2 Methodology

To achieve this objective discussed above in section 3.1, the following methodology is used:

- ∑ A thorough study of existing approaches and techniques in field of sentiment analysis.
- ∑ Collection of related data from Twitter with the help of Twitter API
- ∑ Pre-processing of data collected from Twitter so that it can be fit for mining.

- To build a classifier based on different supervised machine learning techniques.
- Training and testing of build classifier using large datasets
- Computing the result of different classifier using dataset collected from Twitter.
- Comparing results of each classifier and plotting a graph that show the trend of positive and negative sentiment for different political parties.

Chapter 4

Implementation

Data collection is not a simple task, as it may seem. Various decisions have to be made for collecting data. For our thesis we maintain dataset for training, testing and for twitter sentiment analysis. In this chapter we are going to study how data is collected, stored, processed and classified. Before discussing these process and different dataset, let us discuss our proposed architecture.

4.1 Proposed Architecture

As our goal is to achieve sentiment analysis for data provided from Twitter. We are going to build a classifier which consists of different machine learning classifiers. Once our classifier is ready and trained we are going to follow the steps shown in Figure 4.1

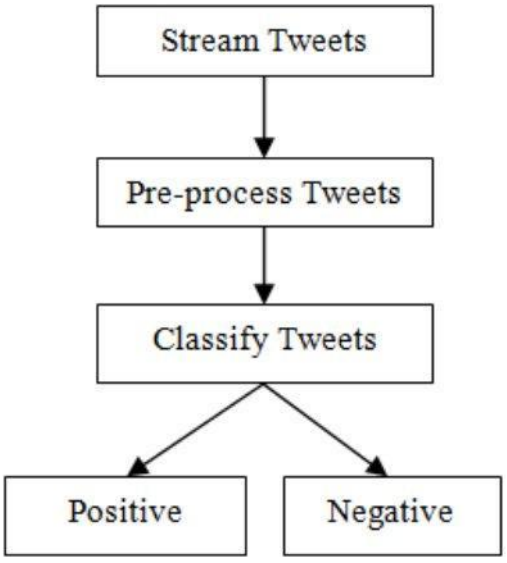


Figure 4.1 Process to classify tweets using build classifier

Step-1 First we are going to stream tweets in our build classifier with the help of Tweepy library in python

Step-2 Then we pre-process these tweets, so that they can be fit for mining and feature extraction.

Step-3 After pre-processing we pass this data in our trained classifier, which then classify them into positive or negative class based on trained results.

Since, Twitter is our source of data for analysis. We are going to stream the tweets from twitter in our database. For this we are going to use Twitter Application.

4.2 Twitter API (Application Programming Interface)

Twitter allows users to collect tweets with the help of Twitter API. Twitter provides two kinds of APIs: REST API and Streaming API. The differences between these are: REST APIs support connections for short time interval and only limited data can be collected at a time, whereas Streaming API provides tweets in real-time and connection for long time. We use Streaming API for our analysis. For collecting large amount of tweets we need long-lived connection and no limit data rate.

4.3 Data Collection

4.3.1 Twitter Data

To use Twitter API we must first have a twitter account. It can be easily created by filling the sign up details in twitter.com website. After this you will be provided with a username and password which is use for login purpose. Once your account is created, you can now read and send tweets on any topic you want to explore.

Twitter provider a platform from which we can access data from twitter account and can use it for our own purpose. For this we have to login with our twitter credentials in dev.twitter.com website. In this website, we first create an application which will be used for streaming tweets by providing necessary details. Once our API is created we can get to know customer key, customer secret key, access token key and access secret key. These keys are used to authenticate user when user want to access twitter data.

As the objective of this thesis is to analyze the sentiment of Tweets posed for political parties, only tweets about related to this should be collected. Hence for this we create a Python script which will be used to fetch tweets from twitter. Before creating this script we first install a library in Python called **tweepy.**

Python is a very powerful language which provides many services with the help of many Python libraries. Tweepy is one of the open source Python library which enables Python to communicate with twitter and use its API to collect data so that we can use it in our program. To install tweepy, just provide a command 'pip install tweepy' in command prompt or bash and we ready to go with our script.

In this script we use all the keys and secrets which we got in API, we first create listener class which is used to load the data from the twitter. Now to gather data we first set up 'OAuth' protocol. OAuth is a standard protocol which is used for authorization. It allow user to log in any third party websites by using any social network website account without exposing passwords. OAuth provides security and authorization to user. The script which we use to access data with the help of twitter is shown is Figure 4.2

```
from tweepy import Stream
from tweepy import OAuthHandler
from tweepy.streaming import StreamListener


#consumer key, consumer secret, access token, access secret.
ckey="####"
csecret="####"
atoken="####"
asecret="####"

class listener(StreamListener):

    def on_data(self, data):
        print(data)
        saveFile = open('twitDB.csv','a')
        saveFile.write(data)
        saveFile.write('\n')
        saveFile.close()
        return(True)

    def on_error(self, status):
        print(status)

auth = OAuthHandler(ckey, csecret)
auth.set_access_token(atoken, asecret)

twitterStream = Stream(auth, listener())
twitterStream.filter(track=["BJP"])
```

Figure 4.2 Code for getting tweets using Twitter API

In this script we have to provide all the keys which are given by Twitter API. To get the tweet for a particular topic we import 'Stream' library from tweepy. In this we pass the authorization detail and the class in which we import tweets. We also apply a filter in the stream which will help us to provide the tweets for the particular topic by providing a keyword related to that topic in filter. Once we run our script, we see tweets are imported from Twitter and we can then use them for our purpose.

4.3.2 Training Data

Other data which we collected for this thesis is training data. This data is used to train the classifier which we are going to build. To collect this data we use NLTK library of Python. NLTK consists of corpora, which is very large and consists of structured set of text files which are used to perform analysis. In these corpora there are various types of text files like quotes, reviews, chat, history, etc. From these corpora we will select files of movie reviews for our training purpose. Sample of these reviews is shown in Table 4.1

Table 4.1 Sample movie reviews in NLTK Corpus

Movie Reviews	CLASS
foolish, idiotic and boring it's so lad dish and youngish , only teenagers could find it funny	NEGATIVE
the rock is destined to be the 21st century's new conan and that he's going to make a splash even greater than arnold schwarzenegger	POSITIVE
Barry Sonnenfeld owes frank the pug big time the biggest problem with roger avary's uproar against the map	NEGATIVE
the seaside splendor and shallow , beautiful people are nice to look at while you wait for the story to get going	POSITIVE

In movie reviews corpus there are around 5000 reviews each for positive and negative feedback. These reviews are short and arranged in text files which are easy to access. We train our classifier from around 80% of the data and then we test it with remaining 20% to check that trained classifier is working properly or not.

4.4 Data Storage

Once, we start getting our data from Twitter API our next step is to store that data so that we can use it for sentiment analysis. We ran our scripts for period of month and collect the tweets for different political parties. Every time we ran the script described in figure a .csv (comma separated values) file is generated which consists of tweets that are extracted from Twitter API. We use .csv format for our collected data files because data consists of many fields. CSV separate each field with a comma, thus make it very easier to access the particular field which consists of text. CSV files also provide faster read/write time as compared to others.

We make separate directories to store tweets of different political parties for respective month. We store them in our hard drive from where these can be easily imported to our snippet and further proceed for analysis. Once we stored our tweet we have to pre process the data stored before applying it to classifier because the data we collect from API is not fit for mining. Therefore pre-processing the data is our next step. Figure 4.3 shows glimpse of different files stored in hard drive.

Figure 4.3 Database of collected tweets

4.5 Data Pre-Processing

Data obtained from twitter is not fit for extracting features. Mostly tweets consists of message along with usernames, empty spaces, special characters, stop words, emoticons, abbreviations, hash tags, time stamps, URL's ,etc. Thus to make this data fit for mining we pre-process this data by using various function of NLTK. In pre-processing we first extract our main message from the tweet, then we remove all

empty spaces, stop words (like is, a, the, he, them, etc.), hash tags, repeating words, URL's, etc. We then replace all emoticons and abbreviations with their corresponding meanings like :-), =D, =), LOL, Rolf, etc. are replaced with happy or laugh. Once we are done with it, we are ready with processed tweet which is provided to classifier for required results. A sample processed tweet is shown in Table 4.2

Table 4.2 Sample Tweet and Processed Tweet

Tweet Type	Result
Original tweet	@xyz I think Kejriwal is a habitual liar, even where he don't needs to lie he tells a lie >☹#AAP
Processed tweet	think, habit, lie, even, don't, need, tell, angry

Cleaning of Twitter data is necessary, since tweets contain several syntactic features that may not be useful for analysis. The pre-processing is done in such a way that data represented only in terms of words that can easily classify the class.

We create a code in Python in which we define a function which will be used to obtain processed tweet. This code is used to achieve the following functions:

- ∑ remove quotes - provides the user to remove quotes from the text
- ∑ remove @ - provides choice of removing the @ symbol, removing the @ along with the user name, or replace the @ and the user name with a word 'AT_USER' and add it to stop words
- ∑ remove URL (Uniform resource locator) - provides choices of removing URLs or replacing them with 'URL' word and add it to stop words
- ∑ remove RT (Re-Tweet) - removes the word RT from tweets
- ∑ remove Emoticons - remove emoticons from tweets and replace them with their specific meaning
- ∑ remove duplicates – remove all repeating words from text so that there will be no duplicates
- ∑ remove # - removes the hash tag class
- ∑ remove stop words – remove all stop words like a, he, the, and, etc which provides no meaning for classification

Table 4.3 shows the various types of contents that are included in tweets and also the actions performed on these contents. Some of the example of clean tweets is shown in Table 4.4

Table 4.3 Removed and modified content

CONTENT	**ACTION**
Punctuation (! ? , . ” : ;)	Removed
#word	Removed #word
@any_user	Remove @any_user or replaced with “AT_USER” and then added in stop words.
Uppercase characters	Lowercase all content
URLs and web links	Remove URLs or replaced with “URL” and then added in stop words
Number	Removed
Word not starting with alphabets	Removed
All Word	Stemmed all word (Converted into simple form)
Stop words	Removed
Emoticons	Replaced with respective meaning
White spaces	Removed

Table 4.4 Sample cleaned data

Raw data	**Clean data**
@jackstenhouse69 I really liked it, in my opinion it def is :)	Really, liked, opinion, def
:(\u201c@EW: How awful. Police: Driver kills 2, injures 23 at #SXSW http://t.co/8GmFiOuZbS\u201d	Sad, awful, police, driver, kills, injures

Once our data is cleaned and ready for processing our next step is to classify this cleaned data into different classes. For this we have to use supervise machine learning classifiers.

4.6 Classification

To classify tweets in different class (positive and negative) we build a classifier which consists of several machine learning classifiers. To build our classifier we used a library of Python called, Scikit-learn. Scikit-learn is a very powerful and most useful library in Python which provides many classification algorithms. Scikit-learn also include tools for classification, clustering, regression and visualization. To install Scikit-learn we simply use on line command in python which is 'pip install scikit-learn'.

In order to build our classifier, we use seven in-build classifiers which come in Scikit-learn library, which are:

- ∑ Naïve-Bayes Classifier
- ∑ MultinomialNB Classifier
- ∑ BernoulliNB Classifier
- ∑ Logistic Regression Classifier
- ∑ SGDC
- ∑ Linear SVC
- ∑ Nu SVC

The reason we are using seven classifiers, so that we can get the more reliable output. To use these classifiers, we write a script in Python, in which we first import the classifier and then we pass the training set to each classifier.

4.6.1 Feature Extraction

As we already discussed in Section 4.3.2, training and testing data is collected from NLTK corpus. We have round 5000 movie reviews each for positive and negative class. We take first 4000 reviews as training set and remaining 1000 as testing sets.

Both the training and testing data must be represented in same order for learning. One of the ways that data can be represented is feature-based. By features, it is meant that some attributes that are thought to capture the pattern of the data are first selected and the entire dataset must be represented in terms of them before it is fed to a machine learning algorithm. Different features such as n-gram presence or n-gram frequency, POS (Part of Speech) tags, syntactic features, or semantic features can be used. For example, one can use the keyword lexicons as features. Then the dataset can be represented by these features using either their presence or frequency.

Attribute selection is the process of extracting features by which the data will be represented before any machine learning training takes place. Attribute selection is the first task when one intends to represent instances for machine learning. Once the attributes are selected, the data will be represented using the attributes. So attributes are the features. Although we used the entire data set in our selection of attributes, the representation of the data must be done on a per instance (Twitter post) basis.

Feature vector plays a very important role in classification and helps to determine the working of the build classifier. Feature vector also help in predicting the unknown data sample. There are many types of feature vectors, but in this process we used unigram approach. Each tweet words are added to generate the feature vectors. The presence/absence of sentimental word helps to indicate the polarity of the sentences. We create a python script to extract the features from the training data. Code snippet for extracting features is shown in Figure 4.4

```
import nltk
import random
from nltk.corpus import movie_reviews

documents = [(list(movie_reviews.words(fileid)), category)
             for category in movie_reviews.categories()
             for fileid in movie_reviews.fileids(category)]

random.shuffle(documents)

all_words = []
for w in movie_reviews.words():
    all_words.append(w.lower())

all_words = nltk.FreqDist(all_words)
word_features = list(all_words.keys())[:4000]

def find_features(document):
    words = set(document)
    features = {}
    for w in word_features:
        features[w] = (w in words)

    return features
print((find_features(movie_reviews.words('neg/cv000_29416.txt'))))
featuresets = [(find_features(rev), category) for (rev, category) in documents]
```

Figure 4.4 Code for extracting features from tweets

Once we extract the features from training data, we are going to pass these in our build classifiers. A script in written in python which is used to pass training sets in classifier. Once, the classifier is trained we can also check the accuracy of each classifier by passing the testing set. Sample script of training and testing of classifier is shown in Figure 4.5

```
classifier = nltk.NaiveBayesClassifier.train(training_set)
print("Classifier accuracy percent:",(nltk.classify.accuracy(classifier, testing_set))*100)
print("NB Classifier accuracy percent:",(nltk.classify.accuracy(classifier, testing_set))*100)


MNB_classifier = SklearnClassifier(MultinomialNB())
MNB_classifier.train(training_set)
print("MNB_classifier accuracy percent:", (nltk.classify.accuracy(MNB_classifier, testing_set))*100)


BernoulliNB_classifier = SklearnClassifier(BernoulliNB())
BernoulliNB_classifier.train(training_set)
print("BernoulliNB_classifier accuracy percent:", (nltk.classify.accuracy(BernoulliNB_classifier, testing_set))*100)

LogisticRegression_classifier = SklearnClassifier(LogisticRegression())
LogisticRegression_classifier.train(training_set)
print("LogisticRegression_classifier accuracy percent:", (nltk.classify.accuracy(LogisticRegression_classifier, testing_set))*100)

SGDClassifier_classifier = SklearnClassifier(SGDClassifier())
SGDClassifier_classifier.train(training_set)
print("SGDClassifier_classifier accuracy percent:", (nltk.classify.accuracy(SGDClassifier_classifier, testing_set))*100)

LinearSVC_classifier = SklearnClassifier(LinearSVC())
LinearSVC_classifier.train(training_set)
print("LinearSVC_classifier accuracy percent:", (nltk.classify.accuracy(LinearSVC_classifier, testing_set))*100)

NuSVC_classifier = SklearnClassifier(NuSVC())
NuSVC_classifier.train(training_set)
print("NuSVC_classifier accuracy percent:", (nltk.classify.accuracy(NuSVC_classifier, testing_set))*100)
```

Figure 4.5 Sample code for training and testing build classifier

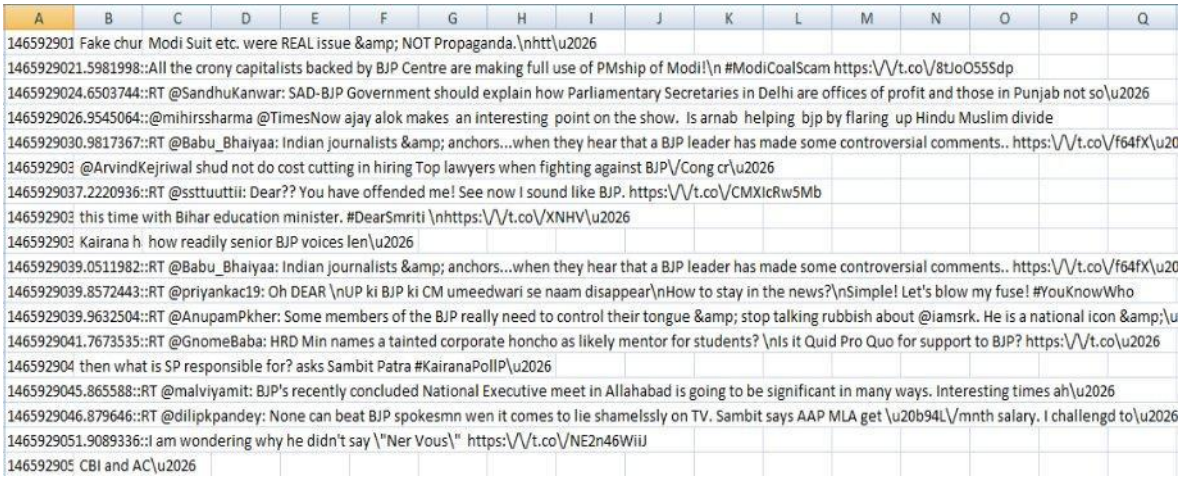
Chapter 5

Results and Analysis

In this chapter we are going to show various results that we have achieved in our implementation.

5.1 Tweets Collected

Tweets are collected with the help of Twitter API. When we ran the script shown in Figure 4.1 a .csv file is generated. A sample file for BJP tweets is shown in Figure 5.1



A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
146592901	Fake chur	Modi Suit etc. were REAL issue & NOT Propaganda.\nhtt\u2026														
1465929021.5981998::All the crony capitalists backed by BJP Centre are making full use of PMship of Modi!\n #ModiCoalScam https:\/\/t.co\/8tJoO55Sdp																
1465929024.6503744::RT @SandhuKanwar: SAD-BJP Government should explain how Parliamentary Secretaries in Delhi are offices of profit and those in Punjab not so\u2026																
1465929026.9545064::@mihirssharma @TimesNow ajay alok makes an interesting point on the show. Is arnab helping bjp by flaring up Hindu Muslim divide																
1465929030.9817367::RT @Babu_Bhaiyaa: Indian journalists & anchors...when they hear that a BJP leader has made some controversial comments.. https:\/\/t.co\/f64fX\u20																
146592903	@ArvindKejriwal shud not do cost cutting in hiring Top lawyers when fighting against BJP\/Cong cr\u2026															
1465929037.2220936::RT @ssttuuttii: Dear?? You have offended me! See now I sound like BJP. https:\/\/t.co\/CMXIcRw5Mb																
146592903	this time with Bihar education minister. #DearSmriti \nhttps:\/\/t.co\/XNHV\u2026															
146592903	Kairana h how readily senior BJP voices len\u2026															
1465929039.0511982::RT @Babu_Bhaiyaa: Indian journalists & anchors...when they hear that a BJP leader has made some controversial comments.. https:\/\/t.co\/f64fX\u20																
1465929039.8572443::RT @priyankac19: Oh DEAR \nUP ki BJP ki CM umeedwari se naam disappear\nHow to stay in the news?\nSimple! Let's blow my fuse! #YouKnowWho																
1465929039.9632504::RT @AnupamPkher: Some members of the BJP really need to control their tongue & stop talking rubbish about @iamsrk. He is a national icon &\u																
1465929041.7673535::RT @GnomeBaba: HRD Min names a tainted corporate honcho as likely mentor for students? \nIs it Quid Pro Quo for support to BJP? https:\/\/t.co\u2026																
146592904	then what is SP responsible for? asks Sambit Patra #KairanaPollP\u2026															
1465929045.865588::RT @malviyamit: BJP's recently concluded National Executive meet in Allahabad is going to be significant in many ways. Interesting times ah\u2026																
1465929046.879646::RT @dilipkpandey: None can beat BJP spokesmn wen it comes to lie shamelssly on TV. Sambit says AAP MLA get \u20b94L\/mnth salary. I challengd to\u2026																
1465929051.9089336::I am wondering why he didn't say \"Ner Vous\" https:\/\/t.co\/NE2n46WiiJ																
146592905	CBI and AC\u2026															

Figure 5.1 Sample Tweets Collected for BJP

Thus each such file for different political parties in stored in a directory as shown in Figure 4.3

5.2 Extracted Features

When we ran the script shown in Figure 4.4, it will extract the features from the training data and also apply a Boolean value to each attribute. The output for extracted features is shown in Figure 5.2

```
Python 3.4.4 (v3.4.4:737efcadf5a6, Dec 20 2015, 20:20:57) [MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: D:\NLTK_python\nltk12.py ==========
{'us': True, 'phoebe': False, 'monstrous': False, 'chronicling': False, 'unspeakable': False, 'wormer': Fals
e, 'poems': False, 'reports': False, 'pates': False, 'cinnamon': False, 'achiever': False, 'beryl': False, '
panel': False, 'prodigal': False, 'lubricants': False, 'subway': False, 'harbours': False, 'masters': False,
'wacked': False, 'cohesive': False, 'lamarr': False, 'suite': False, 'chancellor': False, 'brainy': False, '
defeating': False, 'rile': False, 'interments': False, 'multiplots': False, 'venice': False, 'poisonous': Fa
lse, 'turmoil': False, 'cho': False, '8': True, 'sacrificed': False, 'silvie': False, 'guano': False, 'detec
ts': False, 'evangelizing': False, 'billion': False, 'ignorantly': False, 'condescension': False, 'rard': Fa
lse, 'balrogs': False, 'vesco': False, 'minimum': False, 'molly': False, 'suitors': False, 'realized': False
, 'blazingly': False, 'plank': False, 'animations': False, 'amenities': False, 'slouched': False, 'empathize
s': False, 'embarass': False, 'cheeky': False, 'radioactive': False, 'unbeaten': False, 'juan': False, 'corr
upted': False, 'joyously': False, 'prostitutes': False, 'coil': False, 'resurrects': False, 'coeur': False,
'pronounces': False, 'demise': False, 'shandling': False, 'flowing': False, 'fired': False, 'vibrancy': Fals
e, 'topic': False, 'arguing': False, 'servant': False, 'undoubtful': False, 'pects': False, 'honda': False,
'unbeknownst': False, 'darryl': False, 'strict': False, 'whala': False, 'pedersen': False, 'retroactive': Fa
lse, 'demi': False, 'pixar': False, 'whol': False, '_urban_legend_': False, '84': False, 'infamously': False
, 'noises': False, 'economical': False, 'golan': False, 'nervously': False, 'extraneous': False, 'devirginiz
ed': False, 'aug': False, 'duveyrier': False, 'marcellus': False, 'sift': False, 'malachy': False, 'arrival'
: False, 'emanated': False, 'mcdormand': False, 'margins': False, 'see': True, 'accuser': False, 'reak': Fal
se, 'clamoring': False, 'murdoch': False, '2018': False, 'rectified': False, 'choosen': False, 'thing__about
': False, 'contaminated': False, 'nodded': False, 'fleiss': False, 'edges': False, 'verification': False, 'a
rrests': False, 'fest': False, 'jaunty': False, 'mira': False, 'plato': False, 'controversies': False, 'roma
n': False, 'wingers': False, 'revisitings': False, 'facade': False, 'dolly': False, 'duffel': False, 'razzie
': False, 'includes': False, 'heeled': False, 'celluloid': False, 'tearfully': False, 'errol': False, 'advan
tages': False, 'neilsen': False, 'discontented': False, 'explosion': False, 'wetting': False, 'belief': Fals
e, 'pricelessly': False, 'prouder': False, 'groove': False, 'bragg': False, '30m': False, 'punt': False, 'hi
```

Figure 5.2 Extracted features from training data

5.3 Classifier Accuracy for Training Data

Once we ran the script shown in Figure 4.5, we get the accuracy of each classifier for movie reviews training data. The output is shown in Figure 5.3

```
Python 3.4.4 (v3.4.4:737efcadf5a6, Dec 20 2015, 20:20:57
600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more info
>>>
========== RESTART: D:\NLTK_python\nltk15.py
==========
Original NB Classifier accuracy percent: 72.0
MNB_classifier accuracy percent: 79.0
BernoulliNB_classifier accuracy percent: 78.0
LogisticRegression_classifier accuracy percent: 76.0
SGDClassifier_classifier accuracy percent: 69.0
LinearSVC_classifier accuracy percent: 71.0
NuSVC_classifier accuracy percent: 78.0
```

Figure 5.3 Classifiers accuracy for training data

As, we can show from Figure 5.3, almost all classifiers are giving accuracy of average 75% and above. Thus our build classifier is fully trained and ready for sentiment analysis of twitter data.

5.4 Twitter Data Analysis

For our thesis, we collect tweets on daily basis for political parties BJP, AAP and INC in month of April 2016. We store all these tweets and pre-processed them so that they can be fit for mining. Once our data sets are ready we are going to pass each dataset for different political part to our classifier and check the accuracy for each party.

5.4.1 Analysis for BJP

```
========== RESTART: D:\NLTK_python\nltk15.py ==========
Original NB Classifier accuracy percent: 57.99999999999999
MNB_classifier accuracy percent: 80.0
BernoulliNB_classifier accuracy percent: 75.0
LogisticRegression_classifier accuracy percent: 72.0
SGDClassifier_classifier accuracy percent: 76.0
LinearSVC_classifier accuracy percent: 71.0
NuSVC_classifier accuracy percent: 72.0
>>>
```

Figure 5.4 Classification results when pass with Twitter data for BJP

5.4.2 Analysis for AAP

```
========== RESTART: D:\NLTK_python\nltk15.py ==========
Original NB Classifier accuracy percent: 63.0
MNB_classifier accuracy percent: 82.0
BernoulliNB_classifier accuracy percent: 79.0
LogisticRegression_classifier accuracy percent: 70.0
SGDClassifier_classifier accuracy percent: 68.0
LinearSVC_classifier accuracy percent: 70.0
NuSVC_classifier accuracy percent: 67.0
>>>
```

Figure 5.5 Classification results when pass with Twitter data for AAP

5.4.3 Analysis for INC

```
==================== RESTART: D:\NLTK_python\nltk15.py ====================
Original NB Classifier accuracy percent: 62.0
MNB_classifier accuracy percent: 78.0
BernoulliNB_classifier accuracy percent: 74.0
LogisticRegression_classifier accuracy percent: 69.0
SGDClassifier_classifier accuracy percent: 72.0
LinearSVC_classifier accuracy percent: 65.0
NuSVC_classifier accuracy percent: 71.0
>>>
```

Figure 5.6 Classification results when pass with Twitter data for INC

We analyze a pattern about each party, how people response to each party and their opinion about party leaders. Using live tweets as data set we got an accuracy of around 78% for MultinomialNB Classifier, which means classifier is working correctly. Other results, which we can find are confidence and precision. The classification we done are basically like a vote. Hence, we can apply a voting strategy to calculate maximum number of votes for a feature by using mode, and term it as confidence. It was found that classification of negative confidence is up to 100%, whereas for positive classification is 85% which is reliable. The precision of model was found out to be 70%, which means most of the results returned by classifier are relevant. Figure 5.7 shows the confidence score for positive and negative sentiments

```
Classification: neg Confidence %: 100.0
Classification: pos Confidence %: 100.0
Classification: pos Confidence %: 85.71428571428571
Classification: neg Confidence %: 100.0
```

Figure 5.7 Confidence Score for positive and negative

Finally we calculated the overall progress for each party. We calcute the number of positve over negative sentiment for each party. We take the mode of all positive as well as negative tweets for each party. The overall results from our implementation is shown in Table 5.1 and Figure 5.8

Table 5.1 Sentiment Analysis for BJP, APP and INC for April 2016

Party	Positive %	Negative %
BJP	55.5	45.5
AAP	42.56	57.44
INC	20.10	79.90

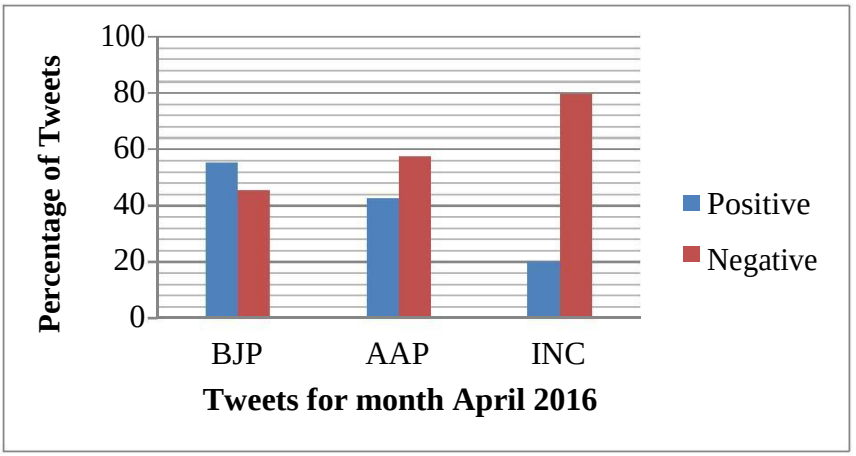


Figure 5.8 Sentiment Analyses for BJP, APP and INC for April 2016

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

Sentiment analysis is used to identifying people's opinion, attitude and emotional states. The views of the people can be positive or negative. Commonly, parts of speech are used as feature to extract the sentiment of the text. An adjective plays a crucial role in identifying sentiment from parts of speech. Sometimes words having adjective and adverb are used together then it is difficult to identify sentiment and opinion.

To do the sentiment analysis of tweets, the proposed system first extracts the twitter posts from twitter by user. The system can also computes the frequency of each term in tweet. Using machine learning supervised approach help to obtain the results.

Twitter is large source of data, which make it more attractive for performing sentiment analysis. We perform analysis on around 15,000 tweets total for each party, so that we analyze the results, understand the patterns and give a review on people opinion. We saw different party have different sentiment results according to their progress and working procedure. We also saw how any social event, speech or rally cause a fluctuation in sentiment of people. We also get to know which policies are getting more support from people which are started by any of these parties. It was shown that BJP is more successful political part in present time based on people opinion. It is not necessary that our classifier can only be used for political parties. It is general classifier. It can be used for any purpose based on tweets we collect with the help of keyword. It can be used for finance, marketing, reviewing and many more.

6.2 Future Scope

Some of future scopes that can be included in our research work are:

- ∑ Use of parser can be embedded into system to improve results.
- ∑ A web-based application can be made for our work in future.

- We can improve our system that can deal with sentences of multiple meanings.
- We can also increase the classification categories so that we can get better results.
- We can start work on multi languages like Hindi, Spanish, and Arabic to provide sentiment analysis to more local.

References

[1] H. Zang, “The optimality of Naïve-Bayes”, Proc. FLAIRS, 2004

[2] C.D. Manning, P. Raghavan and H. Schütze, “Introduction to Information Retrieval”, Cambridge University Press, pp. 234-265, 2008

[3] A. McCallum and K. Nigam, “A comparison of event models for Naive Bayes text classification”, Proc. AAAI/ICML-98 Workshop on Learning for Text Categorization, pp. 41-48, 1998

[4] M. Schmidt, N. L. Roux and F. Bach, “Minimizing finite Sums with the Stochastic Average Gradient”, 2002

[5] Y. LeCun, L. Bottou, G. Orr and K. Muller, “Efficient BackProp”, Proc. In Neural Networks: Tricks of the trade 1998.

[6] T. Wu, C. Lin and R. Weng, “Probality estimates for multi-class classification by pairwise coupling”, Proc. JMLR-5, pp. 975-1005, 2004

[7] “Support Vector Machines” [Online], http://scikit-learn.org/stable/modules/svm.html#svm-classification, Accessed Jan 2016

[8] P. Pang, L. Lee and S. Vaithyanathan, “Thumbs up? sentiment classification using machine learning techniques”, Proc. ACL-02 conference on Empirical methods in natural language processing, vol.10, pp. 79-86, 2002

[9] P. Pang and L. Lee, “Opinion Mining and Sentiment Analysis. Foundation and Trends in Information Retrieval”, vol. 2(1-2), pp.1-135, 2008

[10] E. Loper and S. Bird, “NLTK: the Natural Language Toolkit”, Proc. ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics ,vol. 1,pp. 63-70, 2002

[11] H. Wang, D. Can, F. Bar and S. Narayana, “A system for real-time Twitter sentiment analysis of 2012 U.S.presidential election cycle”, Proc. ACL 2012 System Demostration, pp. 115-120, 2012

[12] O. Almatrafi, S. Parack and B. Chavan, “Application of location-based sentiment analysis using Twitter for identifying trends towards Indian general elections 2014”. Proc. The 9th International Conference on Ubiquitous Information Management and Communication,2015

[13] L. Jiang, M. Yu, M. Zhou, X. Liu and T. Zhao, “Target-dependent twitter sentiment classification”, Proc. The 49th Annual Meeting of the Association

for Computational Linguistics: Human Language Technologies, vol. 1, pp. 151-160, 2011

[14] C. Tan, L. Lee, J. Tang, L. Jiang, M. Zhou and P. Li, "User-level sentiment analysis incorporating social networks", Proc. The 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 1397-1405, 2011

[15] A. Pak and P. Paroubek, "Twitter as a Corpus for Sentiment Analysis and Opinion Mining", vol. 10, pp. 1320-1326, 2010

[16] B. Sun and TY. V. Ng, "Analyzing Sentimental influence of Posts on Social Networks", Proc. The 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design, 2014

[17] A. Go, R. Bhayani and L. Huang, "Twitter sentiment classification using distant supervision", CS224N Project Report, Stanford, vol.1-12, 2009

[18] A. Barhan and A. Shakhomirov, "Methods for Sentiment Analysis of Twitter Messages", Proc.12th Conference of FRUCT Association, 2012

[19] T. Mitchell, "Machine Learning", McGraw Hill, 1997

[20] F. Jensen, "An Introduction to Bayesian Networks", Springer, 1996

[21] T. C. Peng and C. C. Shih, "An Unsupervised Snippet-based Sentiment Classification Method for Chinese Unknown Phrases without using Reference Word Pairs". IEEE/WIC/ACM Int. Conf. on Web Intelligence and Intelligent Agent Technology, vol. 3, pp. 243-248, 2010

[22] R. Feldman, "Techniques and applications for sentiment analysis", Proc. ACM, pp. 56-82, 2009

[23] N. Cristianini and J. Shawe-Taylor, " An Introduction to Support Vector Machines and Other Kernel-based Learning Methods", Cambridge University Press, March 2000

[24] " An Introduction to Python", v3.4.1, 2015 [Online], Available: https://docs.python.org

Video Presentation

https://www.youtube.com/channel/UCxfmif2BHNdya2sYx3FpLjA