A Major Project

Final-Report on

"**Sentiment Analysis**"

Submitted in Partial Fulfillment of the Requirements for the Degree

of Software engineering

Under Pokhara University.

Submitted by:

Aashish Neupane, 15701

Krishesh Shrestha, 15715

Bikal Bahadur Rokaya, 15709

Under the Supervision of:

(**Mr. Subash Manandhar**)

…………………….

Date:

(19th November, 2019)

Department of Software Engineering

NEPAL COLLEGE OF INFORMATION TECHNOLOGY

Balkumari, Lalitpur, Nepal

ACKNOWLEDGMENT

This project would not have been possible without the joint efforts of many individuals. It has been a pleasure for us to acknowledge the assistance and contributions that were very important and supportive throughout the project. We would like to extend our sincere thanks to all of them. We owe special thanks to a number of people who has devoted much of their time and expertise without which it would have been very difficult for us to complete our project entitled "**Sentiment Analysis**".

We are highly indebted to our Project Supervisor **Mr. Subash Manandhar**, for his valuable guidance throughout the project development period and for providing technical support with suggestions which helped our project to grow and foster to a certain level we didn't think of reaching in such a short period.

Last, but not the least, we would like to thank our teachers and colleagues who have been knowingly or unknowingly the part of this project and lent support and views during the entire development time.

We feel obliged in submitting this project as part of our curriculum.
Although, there may be still many who are unacknowledged in this humble vote of thanks, there are none who remain unappreciated.

Aashish Neupane
Krishesh Shrestha
Bikal B. Rokaya
Nepal College of Information Technology,
Balkumari, Lalitpur, Nepal

Abstract

People rely on other people's opinions to make decisions, especially if they belong to their circle of trust. In addition, there are lots of websites of recognized prestige that provide people opinions about different products and services, which are read by millions of people before making a decision. That is why systems for sentiment analysis are becoming increasingly important to automatically process the information and determine feelings of users. They analyze their written words, usually conditioned by the characteristics of microblogging platforms, in which a large number of messages are published every day, providing a great source of information, impossible to be managed manually.

The spread of social networks allows sharing opinions on different aspects of life and daily millions of messages appear on the web. This textual information can be divided in facts and opinions. Opinions reflect people's sentiments about products, personalities and events. Therefore this information is a rich source of data for opinion mining and sentiment analysis: the computational study of opinions, sentiments and emotions expressed in a text. Its main aim is the identification of the agreement or disagreement statements that deal with positive or negative feelings in comments or review.

In the past decade, new forms of communication, such as microblogging and text messaging have emerged and become ubiquitous. While there is no limit to the range of information conveyed by tweets and texts, often these short messages are used to share opinions and sentiments that people have about what is going on in the world around them.

Keywords: Sentiment analysis, Naïve Bayes, Microblogging, Machine Learning, Twitter-Data, Natural Language Processing

Table of Contents

List of Tables

List of Figures

1. INTRODUCTION

Huge volume of data is available on various websites where users are sharing & exchanging their ideas and opinion. With the increase in the use of Facebook, twitter and other social networking sites to express views on topics of interest/concern and having discussions on them is making such sites a pool of opinions. These opinions are also associated with sentiment of the user who is expressing the opinion.

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 Prediction of Elections; Stock Market etc. [1] Nasukawa & YI first introduced the term Sentiment Analysis & Opinion Mining in the year 2003. SA analyzes the user's thoughts/sentiments by determining the polarity (Positive, Negative and Neutral) from huge amount of data availability on Internet. According to researchers Opinion Mining is classified at three different levels as "Document Level", "Sentence Level" and "Aspect level". In this paper we did analysis on Sentence Level.

Textual information can be broadly categorized into two main types: facts and opinions. Facts are objective expressions about entities, events and their properties. Opinions are subjective expressions that describe an individual's sentiments, appraisals or feelings toward entities, events and their properties [1]. People express their opinions not only about products and services, but also about various topics and issues especially from social domains [3].

Twitter is popular online social networking service launched in March 2006.It enables users to send and read tweets with about 140 characters length. Currently Twitter acts as opinionated Data Bank with large amount of data available used for sentiment analysis. Twitter is very convenient for research because there are very large numbers of messages, Many of which are publicly available, and obtaining them is technically simple compared to scarping blogs from the web [4]

1.1. Problem Statement

- The problem is sentiment analysis is classifying the polarity of a given text at the document, sentence, or feature/aspect level.
- Whether the expressed opinion in a document, a sentence or an entity feature/aspect is positive, negative, or neutral.
- Tokenization, replacing sensitive data with unique identification symbols that retain all the essential information about the data.
- Phrase Level Sentiment Analysis in Twitter: Given a message containing a marked instance of a word or a phrase, determine whether that instance is positive, negative or neutral in that context.
- Sentence level Sentiment Analysis in Twitter: Given a message, decide whether the message is of positive, negative or neutral sentiment. For messages conveying both a positive and negative sentiment, whichever is the stronger sentiment should be chosen.

1.2. Project Objectives

- To implement an algorithm for automatic classification of text into positive, negative or neutral.
- Sentiment Analysis to determine the attitude of the mass is positive, negative or neutral towards the subject of interest.
- Graphical representation of the sentiment in form of Pie-Chart.

1.3. Significance of the study

Opinion Mining also called sentiment analysis is a process of finding user's opinion towards a topic or a product. Opinion mining concludes whether the user's view is positive, minus, or neutral about a product, issue, event, etc. Opinion mining and summarization process involve three primary steps, first is Opinion Retrieval, Opinion Classification and Opinion.

Summarization. Review Text is retrieved from review websites. Opinion text in blog, reviews, comments, etc. contains subjective information about the topic. Reviews classified as positive or negative review. Opinion summary is generated based on features opinion sentences by considering frequent features about a matter.

1.4. Project scope and Limitations

This section describes about the requirements of the project. The project should be developed in such a way that it meets all its objectives. Sentiment analysis is a useful tool for any organization or group for which public sentiment or attitude towards them is important for their success - whichever way that success is defined.

1.5. Project Scopes

Social media monitoring Brand monitoring

Voice of customer (VoC) Customer service

Workforce analytics and voice of employee Product analytics

Market research and analysis Agent monitoring

1.6. Project Limitations

Sentiment analysis tools can identify and analyses many pieces of text automatically and quickly. But computer programs have problems recognizing things like sarcasm and irony, negations, jokes, and exaggerations - the sorts of things a person would have little trouble identifying. And failing to recognize these can skew the results.

Disappointed' may be classified as a negative word for the purposes of sentiment analysis, but within the phrase “I wasn't disappointed", it should be classified as positive.

We would find it easy to recognize as sarcasm the statement "I'm really loving the enormous pool at my hotel!", if this statement is accompanied by a photo of a tiny swimming pool; whereas an automated sentiment analysis tool probably would not, and would most likely classify it as an example of positive sentiment.

With short sentences and pieces of text, for example like those you find on Twitter especially, and sometimes on Facebook, there might not be enough contexts for a reliable sentiment analysis. However, in general, Twitter has a reputation for being a good source of information for sentiment analysis, and with the new increased word count for tweets it's likely it will become even more useful.

So, sentiment analysis tools do a really great job of analyzing text for opinion and attitude, but they're not perfect.

2. LITERATURE STUDY/REVIEW

This section summarizes some of the scholarly and research works in the field of Machine Learning and data mining to analyze sentiments on the Twitter and preparing prediction model for various applications.

Sentiment analysis of in the domain of micro-blogging is a relatively new research topic so there is still a lot of room for further research in this area. Decent amount of related prior work has been done on sentiment analysis of user reviews, documents, web blogs/articles and general phrase level sentiment analysis. These differ from twitter mainly because of the limit of 140 characters per tweet which forces the user to express opinion compressed in very short text. The best results reached in sentiment classification use supervised learning techniques such as Naive Bayes and Support Vector Machines, but the manual labelling required for the supervised approach is very expensive. Some work have been done unsupervised and semi-supervised as well, and there is lot of room for improvements.

As the available social platforms are shooting up, the information is becoming vast and can be extracted to turn into business objectives, social campaigns, marketing and other promotional strategies as explained in [13]. The benefit of social media to know public opinions and extract their emotions are considered by authors in [12] and explained how twitter gives advantage politically during elections. Further, the concept of the hashtag is used for text classification as it conveys emotion in few words. They suggested how previous research work suffered from lack of training set and misses some features of target data. They opted two stage approach for their framework- first preparing training data from twitter using mining conveying relevant features and then propounding the Supervised Learning Model to predict the results of Elections held in USA in 2016. After collecting and preprocessing the tweets, training data set was created first by manual labelling of hashtags and forming clusters, next by using online Sentimental Analyzer VADER which outputs the polarity in percentage. This approach reduced the number of tweets or training set and further they applied Support VectorMachine and Naive Bayes classification algorithm to determine the polarity of tweets. Multistage Classification approach was used where an entity classifier receives general class of tweets and categorise them with respect to individual candidates for comparison.

The common approach found in almost all relevant research works constitutes data collection using Twitter API, preprocessing of data, filtering of data then approaches in feature extraction, classification and pattern analysis makes the distinction. In another work [14] of Sentiment Analysis and Influence Tracking on Twitter, authors also predicted the polarity – positive, negative or neutral of tweets by creating a classifier. In addition, they used multiple algorithms and methods to determine the influence of active entity on the tweet patterns of users exhibiting certain emotions. They mined tweets only at the entity level i.e. brand, product, celebrity elements present in tweets rather than the whole sentence in the tweets posted by users. The approach they followed using algorithms to extract features and track the impact and influence made their work different from rest of the literature. The feature extraction process after preprocessing included constructing n grams along with POS taggers taking care of negation part and improving accuracy of classification.

The bag-of-words model is one of the most widely used feature model for almost all text classification tasks due to its simplicity coupled with good performance. The model represents the text to be classified as a bag or collection of individual words with no link or dependence of one word with the other, i.e. it completely disregards grammar and order of words within the text. The simplest way to incorporate this model in our classifier is by using unigrams as features. Generally speaking n-grams is a contiguous sequence of "n" words in our text, which is completely independent of any other words or grams in the text. So unigrams is just a collection of individual words in the text to be classified, and we assume that the probability of occurrence of one word will not be affected by the presence or absence of any other word in the text. This is a very simplifying assumption but it has been shown to provide rather good performance (for example in [10] and [9]). One simple way to use unigrams as features is to assign them with a certain prior polarity, and take the average of the overall polarity of the text, where the overall polarity of the text could simply be calculated by summing the prior polarities of individual unigrams. One way of using prior polarities is we construct our own polarity lexicion but not necessary from our training data, so we don't need to have labelled training data. One way of doing this as proposed by Turney et al. is to calculate the prior semantic orientation (polarity) of a word or phrase by calculating it's mutual information with the word "excellent" and subtracting the result with the mutual information of that word or phrase with the word "poor" [15]. They used the number of result hit counts from online search engines of a relevant query to compute the mutual information.

3. TEAM MEMBERS AND DIVIDED ROLES

Name	Roles	Responsibilities
Krishesh Shrestha	Back-End System Developer	Backend development, API Implementation, Application Workflow Designing
Bikal B Rokaya	UI Designer	Define and design system User Interfaces and User Interaction in the web application.
	End User Documentation	Participate in testing Develop Documentation
Aashish Neupane	Front-End Developer	Develop and create User Interfaces as per the UI Design. Getting data from backend API(REST API) to create fully functioning smooth GUI.

Table 1 : Team Members and Divided Roles

4. METHODOLOGY

We have planned to work following these methodologies for the application of knowledge, skills, tools and techniques to a broad range of activities in order to meet the requirements of our project, Sentiment Analysis. This section presents a detailed information about the software development process, project approach and the tool that we used for our project. 4.1 SOFTWARE DEVELOPMENT LIFECYCLE The framework that we planned to incorporate for developing this project is Incremental model. This model combines linear sequential model with the iterative prototype model. New functionalities will be added as each increment is developed. The phases of the linear sequential model are: Analysis, Design, Coding and Testing. The software repeatedly passes through these phase in iteration and an increment is delivered with progressive changes. [11]

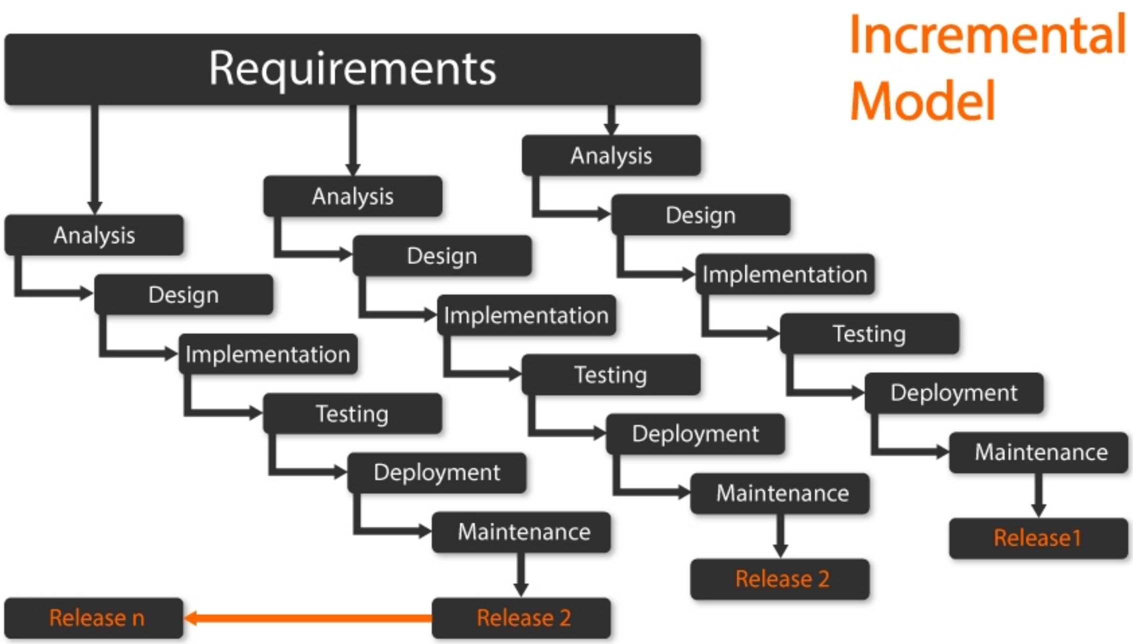


Figure 1: Incremental Model

4.1. ANALYSIS PHASE

In this phase, analysis was performed in order to find out the requirements of the system. The outcome of this phase would be a SRS which is an acronym for “System Requirement Specifications”.

4.2. DESIGN PHASE

In this phase the SRS would be translated into the system’s design. Context Diagram, DFD, ER – Diagram, Use Case Diagram and Class Diagram will be developed.

4.3. CODING PHASE

In this phase, coding would be done according to the design and a working system is achieved/ developed by the end of this process.

4.4. TESTING PHASE

In this phase, the system would be tested. With each testing a list of changes to the system developed, is suggested and the changes will be applied to the software and the software would be delivered as a successive increment until a satisfying system is achieved.

4.5. MANAGING INCREMENTS

Each stage of incremental model adds some functionality to the product and passes it on to the next stage. The first increment (generally known as a core product) was used for a detailed evaluation. This process resulted in creation of a plan for the next increment. The iteration process, which includes the delivery of the increments to the user, continues until the software is completely developed, i.e. iteratively enhance the requirements until the final software is implemented. Our project which implements the Incremental Model, comprises of three increments which are discussed as below:

- INCREMENT 1: Develop Full-fledged Web Application

In this phase we focused on analysis and design of our system with the help of the objectives of our project. This helped us to figure out every aspects of the project and take them into consideration. A full-fledged web application was developed in this phase. We developed an initial project plan to help us in our future increments. The system architecture which is an essential part was developed during this initial iteration. The artifacts to be produced in this phase are:

  - Actors and Use cases
  - Project Boundary
  - System Modules (Web Application Modules)
  - Initial System Architecture
  - Feasibility Study
  - Risk Assessment
  - Domain model
  - ER Diagram
  - Context Diagram
  - Data Flow Diagram
  - Software Architecture Document
  - Cost and Schedule Estimates
  - Activity Diagram

In this phase we worked on integrating API to our system as we have to create a three-tier architecture for our system. REST API is mostly used in cases where client doesn’t need to do much effort and most of the tasks are defined within server itself. Rather than developing a system from the scratch we will use Web AsynHTTPClients like LoopJ and Volley for the two-way communication between Web and the Web Interface.

- INCREMENT 2**:** Recommendation and Collaborative Filtering Algorithm Integration

In this phase we worked on validating the system architecture with our back end. Some conclusions from the previous iteration were helpful in the further development of the system. Here, in this phase we seek to develop a platform that seek to predict the “rating” or “preference” that a user would give to an item, specifically the event. Recommendation Systems typically produce a list of recommendation in one of two ways- through Collaborative and Content-based filtering or the personality-based approach. We seek to build a model from a user’s past behavior as well as similar decisions made by other users. However, due to time-constraint and insufficient resources, we couldn’t implement Recommendation Algorithm in this version.

- INCREMENT 3: Sentiment Analysis Assistant Development and System Deployment

In this phase, we worked on finalizing our deployment of the Web system for the initial part then we’ll go through the entire system deployment which brings many challenges to the system. We’ll also made few changes to our system architecture as per the need. We had to make few other changes to our artifacts.

4.6. PROS OF INCREMENTAL MODEL

Generates working software quickly and early during the software life cycle.

- More flexible – less costly to change scope and requirements.
- Easier to test and debug during a smaller iteration.
- Easier to manage risk because risky pieces are identified and handled during its iteration.
- Each iteration is an easily managed milestone.
- Handling functionality during iteration process

4.7. WHY DID WE CHOOSE INCREMENTAL MODEL?

- The requirements of the complete system were clearly defined and understood.
- Major requirements were defined (however, some details could evolve with time)
- Various backend APIs needed to be integrated and implemented

5. TOOLS AND TECHNOLOGIES

The tools and technologies used for documentation, designing and developing UI/UX, testing are listed below in table

TOOLS	PURPOSE
Python	High Level Programming language used for overall project development
NLTK	NLTK used in Sentiment analysis process are Tokenization, Stop Word removal, Stemming and tagging.
Numpy	Natural Language Processing
GitHub	Manage Source Code/Version Control System
TextBlob	Tool for data analysis and polarity generation from twitter data.
HTML/SCSS	Frontend Development languages.
Typescript	Language used to handle APIs and data flow from backend to frontend.
Django and Django REST Framework	Backend Development for Sentiment Analysis
Angular 8	Frontend Development for Sentiment Analysis

Table 2: Tools and technologies used

6. SYSTEM DESIGN

Designing according to the requirement specification, we have made an attempt to make sure that the system design actually confirms the user requirements of the system.

6.1. MACHINE LEARNING METHODS

We have used Baseline method and in-built classifiers from NLTK: Naive Bayes, maximum entropy.

6.1.1 Naive Bayes

Naive Bayes is a simple model which works well on text categorization. We use a multinomial Naive Bayes model. Class 0' is assigned to tweet d, where

$$c^{\text{¿}} = argmac_c \, P_{NB}\left(\frac{c}{d}\right)$$

$$P_{NB}\left(\frac{c}{d}\right) = \frac{\left(P(c)\right)\sum_{i=0}^{m} P\left(\frac{f}{c}\right)^{n_i(d)}}{P(d)}$$

```
class NaiveBayesAnalyzer(BaseSentimentAnalyzer):
    """Naive Bayes analyzer that is trained on a dataset of movie reviews.
    Returns results as a named tuple of the form:
    ``Sentiment(classification, p_pos, p_neg)``

    :param callable feature_extractor: Function that returns a dictionary of
        features, given a list of words.
    """
    kind = DISCRETE
    #: Return type declaration
    RETURN_TYPE = namedtuple('Sentiment', ['classification', 'p_pos', 'p_neg'])
    def __init__(self, feature_extractor=_default_feature_extractor):
        super(NaiveBayesAnalyzer, self).__init__()
```

```
        self._classifier = None
        self.feature_extractor = feature_extractor

    @requires_nltk_corpus
    def train(self):
        """Train the Naive Bayes classifier on the movie review corpus."""
        super(NaiveBayesAnalyzer, self).train()
        neg_ids = nltk.corpus.movie_reviews.fileids('neg')
        pos_ids = nltk.corpus.movie_reviews.fileids('pos')
        neg_feats = [(self.feature_extractor(
            nltk.corpus.movie_reviews.words(fileids=[f])), 'neg') for f in neg_ids]
        pos_feats = [(self.feature_extractor(
            nltk.corpus.movie_reviews.words(fileids=[f])), 'pos') for f in pos_ids]
        train_data = neg_feats + pos_feats
        self._classifier = nltk.classify.NaiveBayesClassifier.train(train_data)

    def analyze(self, text):
        """Return the sentiment as a named tuple of the form:
        ``Sentiment(classification, p_pos, p_neg)``
        """
        # Lazily train the classifier
        super(NaiveBayesAnalyzer, self).analyze(text)
        tokens = word_tokenize(text, include_punc=False)
        filtered = (t.lower() for t in tokens if len(t) >= 3)
        feats = self.feature_extractor(filtered)
        prob_dist = self._classifier.prob_classify(feats)
        return self.RETURN_TYPE(
            classification=prob_dist.max(),
            p_pos=prob_dist.prob('pos'),
            p_neg=prob_dist.prob("neg")
        )
```

6.1.2 Sentiment Analyzers

Sentiment analysis implementations.

class textblob.en.sentiments.NaiveBayesAnalyzer(*feature_extractor=<function _default_feature_extractor>*)

Naive Bayes analyzer that is trained on a dataset of movie reviews. Returns results as a named tuple of the form: Sentiment (classification, p_pos, p_neg)

Parameters:	feature_extractor (*callable*) – Function that returns a dictionary of features, given a list of words.

RETURN_TYPE

Return type declaration

alias of Sentiment

analyze(*text*)[source]

Return the sentiment as a named tuple of the form: Sentiment(classification, p_pos, p_neg)

train(***kwargs*)[source]

Train the Naive Bayes classifier on the movie review corpus.

class textblob.en.sentiments.PatternAnalyzer

Sentiment analyzer that uses the same implementation as the pattern library. Returns results as a named tuple of the form:

Sentiment(polarity, subjectivity, [assessments])

where [assessments] is a list of the assessed tokens and their polarity and subjectivity scores

RETURN_TYPE

alias of Sentiment

analyze(*text*, *keep_assessments=False*)[source]

Return the sentiment as a named tuple of the form: Sentiment(polarity, subjectivity, [assessments]).

6.1.3 Front-end

Frontend is developed in ANgular 8. Typescript for data manipulation and API handling and HTML/SCSS for UI designing. BackendURL provided by backend , which is responsible for providing Sentiment percentages and Tweets, is set as environment variables , to get datas from JSON response as returned by backend REST API. Angular services are used for normal data handling and communication between different components. Ng-Charts, an angular package for charts , is used for Bar Chart and Pie-Chart, Datas from Rest-API is dynamically provided as real time data to represent them in Charts as analytics. Bootstrap 4 is used for UI decoration and smooth and better user interface. Angular services are used for normal data handling and communication between different components.

7. BUDGET ESTIMATES

7.1. LINE OF CODE

LOC (Lines of Code) is a simple and straight forward way of counting the productivity of a programmer in a given time period.

Using Lines of Code metric, the project size is estimated by counting the number of source instructions in the developed program.

Estimated LOC = 4850
Average Productivity = 200 LOC/pm
Labor Rate = Rs 15,000 per month

Now,
Estimated Project Cost = Estimated LOC * Cost per LOC
= 4850 *

= 4850 * 15000/200

≈ Rs. 3,63,750

8. TESTING

We wanted to make sure that all the elements of the developed worked functioned properly. For this, we created a test cases for our work, in which elements such as validation, reliability and user acceptance will be tested. The system will be tested for normal condition, primarily.

8.1. TESTING TABLE

Each unit of the system was tested for its correct and proper functionality. The unit testing of each components is illustrated in the table below.

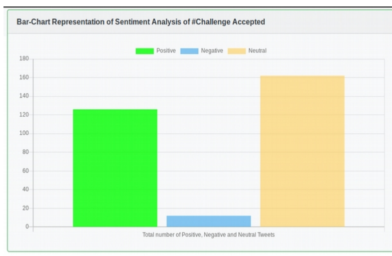
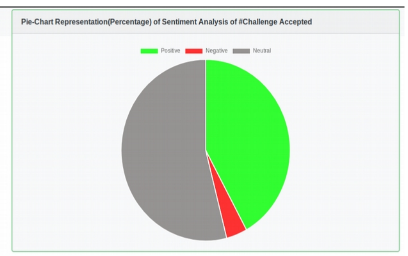
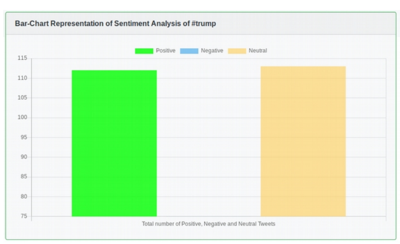
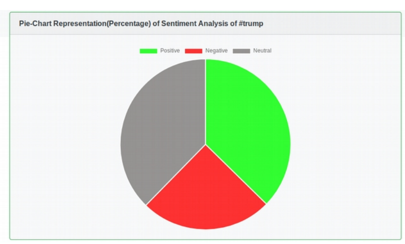
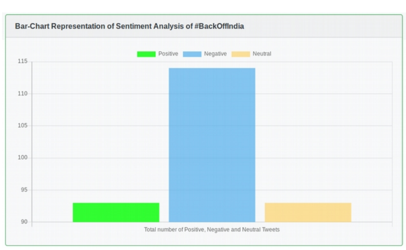
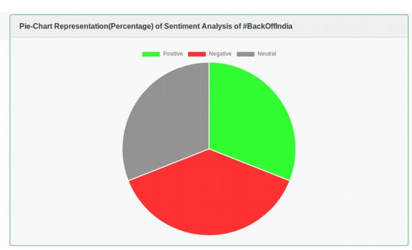
Test No.	Keywords	Results	Remarks
1.	#ChallengeAccepted	 	Successful
2.	#Trump	 	Successful
3.	#BackOffIndia	 	Successful

Table 3: Testing

9. CONCLUSION AND FUTURE EXTENSIONS

Our experiments on twitter sentiment analysis show that part-of-speech features may not be useful for sentiment analysis in the microblogging domain. More research is needed to determine whether the POS features are just of poor quality due to the results of the tagger or whether POS features are just less useful for sentiment analysis in this domain. Features from an existing sentiment lexicon were somewhat useful in conjunction with microblogging features, but the microblogging features (i.e., the presence of intensifiers and positive/negative/neutral emoticons and abbreviations)were clearly the most useful.

Using hashtags to collect training data did prove useful, as did using data collected based on positive and negative emoticons. However, which method produces the better training data and whether the two sources of training data are complementary may depend on the type of features used. Our experiments show that when microblogging features are included, the benefit of emoticon training data is lessened.

The Sentiment Analysis app is now at the initial phase with its beta version having most of the basic functionalities. All the modules have been working after integrating and are ready for the demo. As the features adding up the level of complexity has been increasing as well. However it is not complete with the ideas we have put through and might need more improvisation in the coming days as well. This makes us think about the future extensions that we are going to implement in this application. Some of the extensions we have planned of are:

- Costumized search bar for user to select how many tweets they want to view.
- Application Rating and Feedback to the developers via the application.

10. REFERENCES

[1] Vu Dung Nguyen, Blesson Vaghese, “Royal Birth of 2013:Analysing and Visualising Public Sentiment in the UK using Twitter,” Research Gate, 2013

[2] Anant Arora, Chinmay Patil, Stevina Correia, ”Opinion Mining: An Overview,” International Journal of Advance Research in Computer and Communication Engineering, 2015

[3] Mostafa Karamibekr,Ali a.Ghorbani, ”A structure for opinions in Social Domains,” Social Computing (SocialCom), International Conference, 2013

[4] Brendan O’ Connor, Ramnath Balasubramanyan, ”From Tweets to polls:Linking Text Sentiment to Public opinion time Series,” Proceedings of the fourth International AAAI Conference on Weblogs and Social Media, 2010.

[5] Efthymios Kouloumpis, TheresaWilson, Johns Hopkins University, USA, Johanna Moore, School of Informatics University of Edinburgh, Edinburgh, UK in a paper on Twitter Sentiment Analysis

[6] Hassan Saif, Yulan He and Harith Alani, Knowledge Media Institute, The Open University, United Kingdom in a paper Semantic Sentiment Analysis of Twitter in Nov 2012

[7] Subhabrata Mukherjeel, Akshat Malul, Balamurali A.R.12, Pushpak Bhattacharyya1, 1Dept. of Computer Science and Engineering, IIT Bombay, 2I1TB-Monash Research Academy, IIT Bombay on a paper on TwiSent

[8] Isaac G. Councill, Ryan McDonald, Leonid Velikovich, Google, Inc., New York on a paper on What's Great and What's Not: Learning to Classify the Scope of Negation for Improved Sentiment Analysis

[9] Alec Go, Richa Bhayani and Lei Huang. Twitter Sentiment Classification using Distant Supervision. Project Technical Report, Stanford University, 2009.

[10] Efthymios Kouloumpis, Theresa Wilson and Johanna Moore. Twitter Sentiment Analysis: The Good the Bad and the OMG! In Proceedings of AAAI Conference on Weblogs and Social Media (ICWSM), 2011.

[11] Roger S. Pressman, Ph.D., Seventh Edition, Software Engineering: A Practitioner’s Approach, McGraw Hill, 2010

[11] Peter D. Turney. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL), 2002.

[12] J. Ramteke, S. Shah, D. Godhia, and A. Shaikh, “Election result prediction using Twitter sentiment analysis,” in Inventive Computation Technologies (ICICT), International Conference on, 2016, vol. 1, pp. 1–5.

[13] Alexander Pak and Patrick Paroubek. "Twitter as a corpus for sentiment analysis and opinion mining". In Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC’10), may 2010.

[14] R. Mehta, D. Mehta, D. Chheda, C. Shah, and P. M. Chawan, “Sentiment analysis and influence tracking using twitter,” International Journal of Advanced Research in Computer Science and Electronics Engineering (IJARCSEE), vol. 1, no. 2, p. pp–72, 2012.

[15F] Peter D. Turney. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL), 2002.

APPENDIX

SYSTEM SNAPSHOT

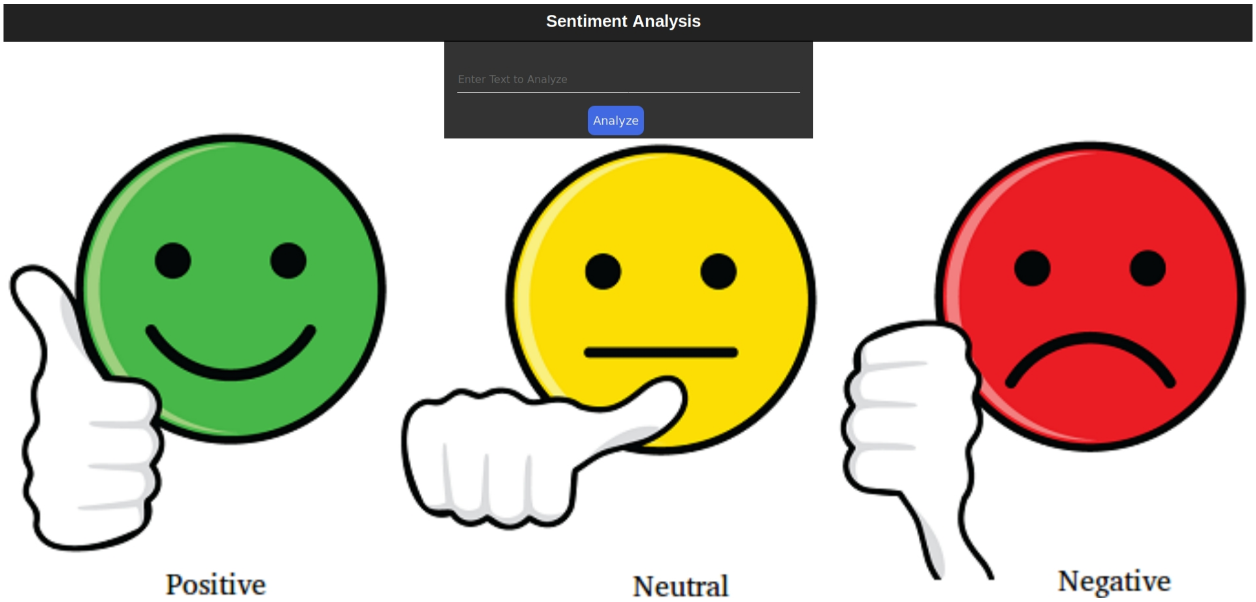


Figure 2 Landing Page

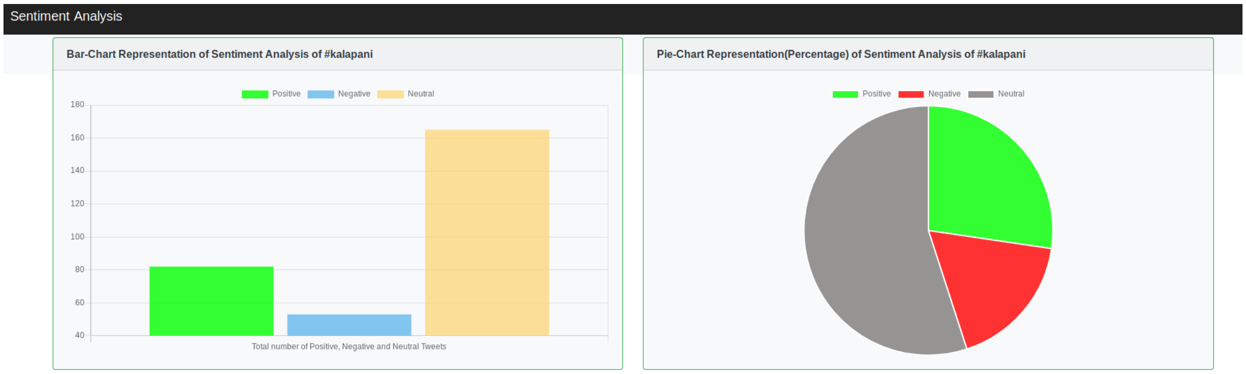


Figure 3 Analytic of Sentiment

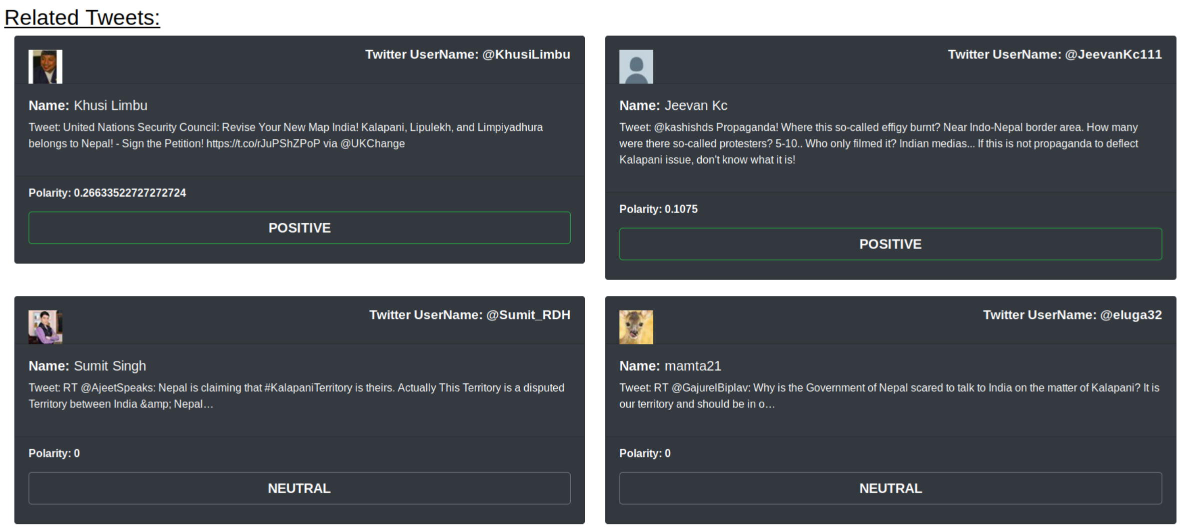


Figure 4 Tweets related to the searched keywords