**SENTIMENT ANALYSIS PERFORMED ON TWEETS TO DETECT EMOTIONS BEHIND AMBIGUOUS PHRASES**

Submitted in partial fulfilment of the requirements

of the degree of

**B. E. Computer Engineering**

By

**Prerna Parmeshwaran 60004120073**

**Mihika Shah 60004120100**

**Dyuti Shukla 60004120110**

Guide:

**Prof. Kiran Bhowmick**

Assistant Professor

|  |  |  |
| --- | --- | --- |
| D:\Admission\Admissions_1011\wwwroot\hdrlogo.gif | Department of Computer Engineering  D. J. Sanghvi College of Engineering  Mumbai – 400 056 | E:\new logo.JPG |

University of Mumbai

2015-2016

**CERTIFICATE**

This is to certify that the project entitled **“Sentiment Analysis Performed on Tweets to Detect Emotions behind Ambiguous Phrases”** is a bona fide work of **Prerna Parmeshwaran (60004120073), Mihika Shah (60004120100) and Dyuti Shukla (60004120110)** submitted to the University of Mumbai in partial fulfilment of the requirement for the award of the degree of B.E. in Computer Engineering

**(Prof Kiran Bhowmick)**

**Internal Guide**

**(Dr. N. M. Shekokar) (Dr. Hari Vasudevan)**

**Head of Department Principal**

**Project Report Approval for B.E.**

This project report entitled **Sentiment Analysis Performed on Tweets to Detect Emotions Behind Ambiguous Phrases** by ***Prerna Parmeshwaran (60004120073), Mihika Shah (60004120100), Dyuti Shukla (60004120110)*** is approved for the degree of ***B.E. in Computer Engineering.***

Examiners

1.---------------------------------------------

2.---------------------------------------------

Date:

Place:

Declaration

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

----------------------------

(Prerna Parmeshwaran

60004120073.)

----------------------------

(Mihika Shah

60004120100.)

----------------------------

(Dyuti Shukla 60004120110.)

Date:

**ABSTRACT**

Historically, humans have always ventured into public forums to air their opinions about subjects that affected them personally, or evoked strong emotions. With the advent of the internet and social networking sites such as Facebook, Twitter and Tumblr, public opinion is now recorded for posterity and displays a wealth of information about what affects the world today, what they adore or dislike, and can be even used as reliable measures of public sentiment for large scale projects and policies.

The potential for the use of such public websites as indicators of general feeling is humongous, especially since social media is geared towards exchange of opinions and ideas on a large scale. As we move to an era of a global community, we can no longer afford to ignore the impact the web plays in the daily lives of the people worldwide. Today, the combined opinion of the internet may even force large companies to release product versions that expressly fix the problems that were decried by their consumers in the first place, or political parties to launch schemes that are preferred by the public in general.

Global sentiment is a powerful weapon that can be harnessed by the use of Sentiment Analysis techniques. Knowing how the general populace perceives a product or an idea is crucial to the survival of any business. Our project focuses on extracting such opinion from sentences, looking beyond sentence polarity to give its users clearer perspective of a review or a tweet. We hope to be able to look beyond simply declaring sentences as “positive” and “negative”, but to be able to tell the emotions that were at the forefront of the consumer’s mind when he wrote a particular statement, such as “joy”, “anger”, “disappointment” and even “pleasure”.

In conclusion, as we move towards a world where opinion decides what happens next, knowledge of what emotions the product evokes is extremely important. Our project is an attempt to discern emotions behind tweets, which can be used to gauge public opinion behind any important event.

**Contents**

|  |  |  |
| --- | --- | --- |
| Chapter | Contents | Page Number |
| 1 | INTRODUCTION | 1 |
|  | 1.1 Description | 1 |
|  | 1.2 Problem Formulation | 2 |
|  | 1.3 Motivation | 3 |
|  | 1.4 Proposed Solution | 3 |
|  | 1.5 Scope of the Project | 5 |
| 2 | REVIEW OF LITERATURE | 6 |
| 3 | SYSTEM ANALYSIS | 16 |
|  | 3.1 Functional Requirements | 16 |
|  | 3.2 Non-Functional Requirements | 16 |
|  | 3.3 Specific Requirements | 17 |
|  | 3.4 Use Case Diagrams | 17 |
| 4 | ANALYSIS MODELLING | 19 |
|  | 4.1 Database Modelling | 19 |
|  | 4.2 Activity Diagram | 20 |
|  | 4.3 Functional Modelling | 22 |
|  | 4.4 Timeline Chart | 23 |
| 5 | DESIGN | 25 |
|  | 5.1 Architectural Design | 25 |
|  | 5.2 User Interface Design | 26 |
| 6 | IMPLEMENTATION | 28 |
|  | 6.1 Algorithms Used | 28 |
|  | 6.2 Working of the Project | 28 |
| 7 | Testing | 33 |
|  | 7.1 Test Cases | 33 |
|  | 7.2 Types of Testing Used | 34 |
| 8 | RESULTS AND DISCUSSIONS | 35 |
| 9 | CONCLUSION AND FUTURE SCOPE | 37 |
| 10 | REFERENCES | 38 |

Appendix

Publications

Acknowledgements

**List of figures**

|  |  |  |
| --- | --- | --- |
| Figure Number | Figure Caption | Page Number |
| 1-1 | Model of the proposed system | 4 |
| 2-1 | Russell’s circumplex model | 14 |
| 2-2 | Plutchik’s emotion wheel | 14 |
| 3-1 | Use case diagram | 17 |
| 4-1 | Activity diagram | 21 |
| 4-2 | Level 0 DFD | 22 |
| 4-3 | Level 1 DFD | 22 |
| 4-4 | Level 2 DFD | 22 |
| 4-5 | Work breakdown structure for the project | 23 |
| 4-6 | Timeline chart for the project | 24 |
| 5-1 | Architectural design | 25 |
| 5-2 | Main Visualization of Emotion Extraction | 26 |
| 5-3 | Level one of the wheel, showing tweets classified as the emotion “rage” | 27 |
| 5-4 | Level 2 of the Wheel, showing the emotions present in a single tweet | 28 |

**List of tables**

|  |  |  |
| --- | --- | --- |
| Table Number | Table Caption | Page Number |
| 1-1 | Examples of ambiguous tweets | 2 |
| 7-1 | Test Cases | 33 |
| 8-1 | Summary of the results for different keywords | 35 |
| 8-2 | Sample of the Sharapova data set | 36 |

# Chapter 1

# Introduction

## Description

Sentiment analysis refers to the use of computational techniques such as natural language processing, classification techniques, computational linguistics and so on to extract information related to sentiment and emotions from text. This has a lot of applications ranging from marketing to customer service where the opinion of the end users of various products matter a great deal to companies. Sentiment analysis techniques are popularly applied on social media and reviews so as to detect the attitude of the writer.

Due to the large number of Twitter users worldwide, it is now a gold mine for anybody who wishes to study sentiments of the users. Twitter users tweet about a myriad of topics ranging from politics to newly launched electronic products, so there would be no dearth of opinions expressed about any topic on Twitter.

The problem with text analysis on Twitter is that there is a lot of ambiguity in the statements made by the users. Due to the 140 character limit imposed on Twitter users, they are forced to use internet slang and wrong/non-standard spellings to express their views. In addition to this, there is a lot of sarcasm present online these days. All this lends ambiguity to the text and hence, automated classification of tweets based on sentiments becomes quite difficult. We attempt to address some of these issues related to ambiguous phrases in our project.

Another key feature of the project is the extraction of the actual emotion expressed in the tweets. Most existing sentiment analysis tools available online only give us the polarities of the input statements, i.e. whether the given statement expresses a positive or a negative opinion. For example, if one considers the sentence “The slow touch screen is very irritating”, popular sentiment analysis tools will only inform us that the overall sentiment conveyed is negative.

However, we wish to go further and classify this sentence as “angry” or perhaps “dissatisfied”. Such an analysis of tweets will make the study of opinions more detailed and informative.

The table shown below gives some examples of ambiguous tweets.

Table 1-1: Examples of ambiguous tweets

|  |  |
| --- | --- |
| **Tweet** | **Reason for ambiguity** |
| #ICYMI Russia to conduct air strikes on theatres showing Shahid Kapoor- Alia Bhat starrer Shaandaar | Sarcasm |
| Had a pretty sick comeback win against Ninja and friends on Empire TS while streaming last night. Uploading later tonight #hoaXfiles #Halo5 | Words with multiple meanings |
| Absolutly @tarsem\_insan bro #BlockBusterMSG2 is gr8 source f inspiration 2 all @Gurmeetramrahim proved dat simplicity cn b a BLOCKBUSTER!!!! | Slang, incorrect spellings |
| even though I am working I will be able to watch #Shaandaar cause 22nd is a holiday !!! whoooo @shahidkapoor I loveeeeeee you :\* | Exaggerated words |
| @TheXFactor @4thImpactMusic I'm so glad they made it through the live shows... They deserve it. Their performance was a terribly awesome. ;\* | Words of multiple polarities |

## Problem Formulation

In recent years, the vast growth of social networking sites has provided users with a large platform to express their thoughts about various products, events and people. The advantages of doing this are twofold. Firstly, this enables the entities such as companies, PR teams and political parties to judge public opinion about their activities. Secondly, this provides potential customers an overview of the various products available in the market, along with their drawbacks and advantages. With the fast growing data, it becomes impossible for people to identify user’s opinions manually, and it has become imperative to automate the task of extracting people’s sentiments.

While there are various existing tools to perform sentiment analysis on tweets, none of them actually focus on extracting the actual emotion behind the user’s statements, or look at the ambiguity existing in any statement. Most of the tweets contain ambiguity mainly because of the presence of slang, context-dependent words, negation and so on.

Our main goal in this project is to focus on understanding the sentiment of a person even though there is some vagueness in their statement. We first subject the text to some pre-processing, where slang is detected and replaced by something easily understandable. We then apply the Plutchik’s model on these sentences with which we can find out the approximate emotion of the user. This project will make it possible to identify clearly what a user has to say about a product, and it will prove extremely useful to product developers as they will be able to keep in mind exactly what their customer wants and expects from them.

## Motivation

Due to the enormous viability of Sentiment Analysis in consumer feedback, we have chosen to work on it as our project. However, the existing techniques focus primarily on determining the polarity of the sentences in question, which does not provide adequate information on the subject at hand. Knowing the polarity of a sentence as “positive” or “negative” gives no discernible information about the emotion expressed, requiring the reviewers to read the statement in question and manually decipher the sentence.

With these failures in mind, we have broadened our scope to actual extraction of the emotions that pervade any given sentence. This allows for more accurate understanding and filtering of responses that can be mined at any given point. Classifying sentences based on the emotion behind them will provide accurate data to any company about what was liked and what was not, and would make grievance redressal easier.

## Proposed Solution

The steps for the processes used by us are described below:

1. Retrieval of Data: Using the existing Twitter API, the twitter data is mined for data extraction. The tweets are selected based on a particular keyword. We have elected to use the Twitter API due to ease of data extraction.
2. Preprocessing: In the data preprocessing stage, information like Twitter handles, timestamps of the message, links and videos are removed. This information need to be removed as it is largely irrelevant and may faulty results by our system.
3. Tweet Correction: Tweets often contain slang, misspellings and other irrelevant data. Thus we correct the misspellings in the sentences and replace the slang with words from Standard English that are equivalent to the slang. Since we are only concerned with English Tweets, all non-English tweets are also filtered out.
4. Emotion Extraction: For the purposes of our system, we consider the “Modified Plutchik’s Wheel of Emotion” which divides all emotions into an eight-point wheel. It consists of an inner wheel and an outer wheel, where the inner wheel represents the more intense emotion as compared to the outer wheel. We also consider three additional emotions outside the wheel.
   1. Creating the dictionary: We create a dictionary by calculating word distances between common English words and the 19 emotions present in the wheel of emotion. We use the UMBC Semantic Similarity Service for this, which computes the semantic similarity between the two words.
   2. Extraction of emotion: Now, the process of emotion extraction starts. Each tweet is broken down word by word, which is compared with the distance matrix present in the dictionary. Common negation words and their effect on the rest of the sentence are also taken into consideration. The average of the various word distances for each word of the tweet is then computed. The target emotion vector is compared with the resultant vector using cosine similarity, and the resultant emotion is then computed.

5. The results are displayed graphically.

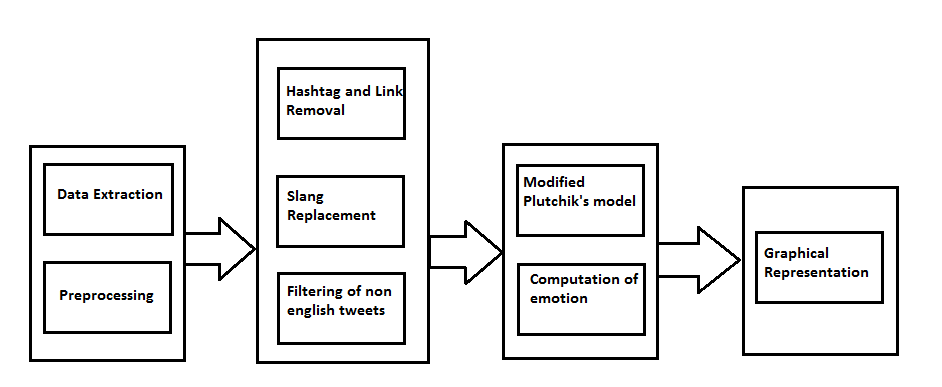
****

Fig 1-1: Model of the proposed system

## Scope of the Project

The current scope of the project is limited to Tweets in English only. This is because most tweets are in English. We also focus on removing ambiguity caused by the use of slang and abbreviations. While extracting the emotions from the tweets, we consider the presence of common words that induce negation, and the effect that these words have on the rest of the tweet.

# Chapter 2

# Review of Literature

## 2.1 Methods for Sentiment Analysis

Sentiment analysis methods are traditionally classified into two broad types: lexicon based and machine learning based techniques. A third kind of method, that is a hybrid of the above is increasingly being used in recent times, because it tends to perform better. In addition to these three methods, there is a completely different kind of approach called concept level sentiment analysis. Each of these techniques is described below.

*2.1.1 Lexicon based methods*

As the name suggests, lexicon based methods [1] make use of a word dictionary, or a lexicon, to be able to classify statements. An opinion word is one that conveys some sentiment to the reader. For example, ‘happy’ is an opinion word that conveys a positive sentiment, but ‘apple’ is not. Opinion words and their related emotions are collected in a sentiment dictionary. The standard steps in the lexicon based sentiment analysis technique are:

1. Pre-process the text and remove redundant information like HTML tags.
2. Initialize the overall sentiment score (s) to 0.
3. Tokenize the text. Go through each token in the sentence and match it with the entries in the sentiment dictionary. If a match is found:
4. If the associated sentiment is positive, increment s.
5. Else if the associated sentiment is negative, decrement s.
6. After the entire statement has been parsed, s is compared to a pre-defined threshold value t (typically 0). If s is greater than t, the overall sentiment is positive. If s is less than t, the overall sentiment is negative.

The lexicon based method can be called an ‘unsupervised’ method, because a separate labeled data set is not required for training the classifier. This is a huge advantage of the lexicon based method because for training in the machine learning methods, a large training dataset is needed.

Turney [2] proposed a lexicon based method where ‘excellent’ and ‘poor’ are used as seed words, and the semantic direction of phrases are calculated with respect to these words.

The lexicon based approach has some limitations:

1. In ambiguous phrases, the meaning of the words used is usually not the standard meaning. Therefore, this approach will not work well for classifying ambiguous statements.
2. In tweets, the context in which words are used is very important to gauge the sentiment behind them. The lexicon based approach does not take context into consideration.
3. Constructing a detailed lexicon is a tedious process.

*2.1.2 Learning based methods*

Machine Learning methods use classification techniques to classify text. Most machine learning methods use supervised learning to classify the text, and training and test sets are needed. The training set is used to identify the various characteristics of the text. The different techniques used in this category are Naive Bayes (NB), Support Vector Machines (SVM), and Maximum Entropy (ME) [2].

Feature selection is a very important aspect of machine learning based methods, as they can tell us how documents are represented. The features which are commonly used for sentiment analysis are [1, 3]:

* 1. Term presence and their frequency: This consists of unigrams, bigrams, term frequency and its presence.
  2. Part of speech tagging: As the name suggests, part of speech tagging marks a word according to its category (i.e. part of speech like nouns, adjectives and so on), based on both context and its definition. POS tagging is a major factor in identifying the sentiment of a text.
  3. Negations: Since negations can substantially alter the meaning of the statement, it has to be taken into account to identify the sentiment.
  4. Opinion words and phrases: Statistical-based or lexicon-based approaches are used to express sentiments as either positive or negative.

*2.1.2.1 Naive Bayes*

Naive Bayes Classifier [4] is a probabilistic classifier which uses Bayes Theorem. The Bayes Rule which forms the basis of this classifier can be given by:

where P(c) is the prior probability of the class and P(d) is the prior probability of the document.

The basic assumption of this classifier is that the features are assumed independent. This model can be combined with a decision rule, a common rule being the maximum a posteriori model or the MAP model. Using this rule, a document d can be classified into class:

There are a few variations [4] to the Naive Bayes Model, two of which are the Bernoulli Model and the Multinomial Model. The Bernoulli Model is a Bayesian Network with no word dependencies and binary word features. It generates a Boolean indicator for every term in the vocabulary based on its presence or absence. It also takes into account words that are absent in the document. This has been found to perform well with small vocabulary sizes. The Multinomial Model is a unigram language model with integer word counts. This is used when the frequency of the word that occurs in the statement plays a key role in classifying text. It performs much better than the Bernoulli Model for large vocabulary sizes, providing on average a 27% reduction in error. A modified version of the Multinomial model is the binarized model, which does not take into account the frequency of the word, just the fact that it occurs in the statement is enough. It has been used effectively on a large scale for sentiment analysis.

The advantage with Naive Bayes is that it is extremely simple to implement, but since features are assumed to be independent of each other, POS cannot be used.

*2.1.2.2 Support Vector Machines*

The main idea for SVM for sentiment classification [4] is to find a hyperplane that divides the documents as per the sentiment, and the margin between these classes should be as high as possible. The principle of SVM is Structural Risk Minimization. The objective is to find a hypothesis h for which the error is the lowest. If we symbolize the hyper plane as h and the tweet as t, and represent the classes into which the tweet has to be classified as Cj∈ {1,-1} corresponding to the sentiment of the tweet, the solution can be written as:

The texts that have α>0 are the ones which contribute in finding the hyperplane.

SVMs have the ability to handle large feature spaces with high number of dimensions. As long as the text classification is linearly separable, SVM does not have any issue with the number of features in the feature space. Moreover, SVM does not assume any features to be irrelevant, which sometimes leads to a loss of information. However, the main problem with SVM is that it is difficult to identify which features are more important for classification.

*2.1.2.3 Maximum Entropy*

The main principle of maximum entropy [4] is that a uniform model is preferred to satisfy the given constraints. It can be used to estimate any probability distribution. In maximum entropy, the training data is used to set constraints on the conditional distribution. Each constraint should express a characteristic of the training data that should also be present in the learned distribution.

The first task in maximum entropy is to identify a set of feature functions that will be useful for classification. Then, for each feature, its expected value is measured over the training data and this is taken as a constraint for the model distribution.

It can be shown that the distribution is always of the exponential form.

Where f(c,d) is a feature, Ai is a parameter to be estimated and Z(d) is just a normalization function. An advantage of Maximum entropy is that it does not have any independence assumptions. As a result of this, bigrams and phrases can be added as features for classification. However, overfitting is one of the drawbacks of Maximum Entropy.

Machine learning methods have a big advantage over lexicon based methods as they are able to handle large collections of data.

*2.1.3 Hybrid methods*

Hybrid methods incorporate features of both lexicon and learning based methods. They have been observed to have higher accuracy than lexicon and learning methods used alone [1]. Some hybrid techniques that have been used to perform sentiment analysis are as follows:

*2.1.3.1    Feature based sentiment analysis*

In this approach [5], an attempt is made to identify various features in a sentence and identify the sentiment associated with each of them. A sentiment dictionary is used for feature extraction and machine learning algorithms are used to train the classifier. The steps involved are:

1. Feature and opinion extraction: A tokenized sentence is passed as the input to this step. The output is a list of feature words and a list of opinion words that were present in that sentence.
2. Anaphora resolution: Examples of anaphora include the usage of pronouns in a sentence so as to avoid the repetitive usage of a noun. Usage of anaphora makes the sentence ambiguous because it is difficult for the computer to map opinion words to the corresponding feature. Hence, backtracking is used to resolve the usage of anaphora, so as to map the opinions to the correct features.
3. Feasibility analysis: Extraneous words that are not related to the process of sentiment analysis are eliminated in this step.
4. Statistical features identification: A set of positive and negative seed words already exists in the form of the sentiment dictionary. In this step, the correlation of the opinion words with either of these sets is calculated.
5. Sentiment determination: Now that the nature of the opinion words has been calculated, the overall sentiment expressed in the sentence is determined using various machine learning algorithms. Usually, supervised ML techniques are used for this purpose.

*2.1.3.2    Identifying polarities of words using emoticons*

This technique [6] is especially important for sentiment analysis on Twitter because most users tend to use a lot of emoticons along with words to express themselves. It uses both lexical and machine learning techniques.

The underlying assumption of this method is that the overall orientation of the emoticons used in a sentence is the same as the overall sentiment expressed by the words in that sentence. In simpler terms, the emotions expressed using words and emoticons in one particular sentence would be the same.

A list of annotated emoticons (i.e. a list of emoticons and their related sentiments) has to be used for this task. Also, a lexicon containing words and their sentiments has to be used. The steps involved in this method of sentiment analysis are:

1. Identify the emotions conveyed by the emoticons. When multiple emoticons are used in a single sentence, it is assumed that they will be of the same orientation.
2. Create a model for the words used in the sentence. A vector consisting of the words is created, in conjunction with the emoticons used. Dimension reduction is done by using principal component extraction.
3. Machine learning techniques are used to classify the sentence using the features identified above. In this technique, an SVM is used for the classification.

The assumption about the orientations of the emoticons and the words in the sentence makes it quite easy to implement, because it is usually correct. However, in ambiguous phrases, this assumption may not hold true and this method would fail in such a case.

*2.1.4 Concept Level Sentiment Analysis*

Unlike lexical and keyword spotting methods, Concept-Level sentiment Analysis [7] does not depend wholly on lexical and keyword spotting methods. Concept level methods can be used to determine emotions that are expressed subtly, as long as they can be connected or related to concepts that are present in the sentences. Concept level techniques depend on large semantic dictionaries that act as a repository for semantic words and concepts. Sentences in concept level dictionaries are parsed into concepts that display tangible relationships with each other.

Thesaurus and Commonsense dictionaries are two commonly used propagation methods, but the use of Thesauruses are generally limited to lexical dictionaries only. A thesaurus is used to map syntactically similar words by means of some defined syntax rules.

Commonsense dictionaries on the other hand map relations between concepts. Each concept aggregates all possible surface words and phrases. These dictionaries when employed in sentiment analysis have higher recall than word based match techniques with greater variety of relations among the elements.

Concepts in such commonsense networks may be broken down into word based relations or parsing the entire sentence to find assertions in sentences. Assertions comprised of two concepts, and words such as “Used For” “IsA” are used to demonstrate the relation between them. For comparison, parsing breakdown of a few sentences is given below.

1. *Banana is a fruit.*

Here, the sentence banana is a fruit can be parsed into assertions thus: Banana/ isA /Fruit. The concepts mentioned here are “Banana” and “Fruit” with “IsA” specifying the relation between them.

1. *A saxophone is used for jazz.*

Here, the concepts are “Saxophone” and “jazz”, which are related by the use of the words “Used For”, signifying that concept “Saxophone” is required to play Jazz.

1. *Microprocessors are a concept in computing science*.

“Microprocessors” and “Computing Science” are the concepts gained by delineating the assertions. The correct relation here is given by “are a concept in”, which means that the first concept is related to the second by the virtue of having a context in the second concept.

The division of such concepts has tremendous value in sentiment analysis projects, where sentiments can be related to each other on the basis of the concepts given in them. Assumptions can be made on the fact that semantically related words have similar sentiments related to them.

The approaches taken by various semantic dictionaries are given below.

1. ANEW and SenticNet [7, 8, 9]: ANEW and SenticNet are exploited for propagating sentiment values based on the assumption that semantically related concepts share common sentiment. Polarity accuracy, Kendall distance, and average-maximum ratio are

used, instead of mean error, to better evaluate sentiment dictionaries. A similar approach is adopted in which presents a methodology for enriching SenticNet concepts with affective information by assigning an emotion label to them.

1. ISEAR and SentiWordNet [7, 10, 11]: Various features can be extracted from ISEAR, as well as similarity measures that rely on the polarity data provided in SenticNet (those based on WordNet-Affect) and ISEAR distance-based measures, including point-wise mutual information, and emotional affinity.  Works which propose the re-evaluation of objective words in SentiWordNet by assessing the sentimental relevance of such words and their associated sentiment sentences may also be used to provide effective methods of sentiment analysis.

## 2.2 Methods for Emotion Extraction

Classifying a statement as a positive, negative or a neutral opinion will not suffice if one wants to gather more information about the user’s mood while tweeting. Identifying the basic emotion of the user will help organizations understand how the user really perceives his product. Performing emotion extraction on text is an uphill task because human emotions are very complex and subjective in nature. However, with the help of emotional models, one can broadly classify a piece of text into a predefined set of human emotions. Some significant emotional models are as shown below:

1. Russell’s circumplex model: James Russell [12] conceptualized emotions as being distributed in a two dimensional space, where the X axis represents valence emotions and the Y axis represents the arousal. Valence is related to the polarity, i.e. the attractiveness or aversiveness towards a particular subject or an event. Arousal deals with the intensity of the emotions felt. There exists an online twitter sentiment visualization tool [13] which makes use of dictionaries such as ANEW to determine the valence and arousal values of the tweets, and then determines the emotion using the Russell’s model.

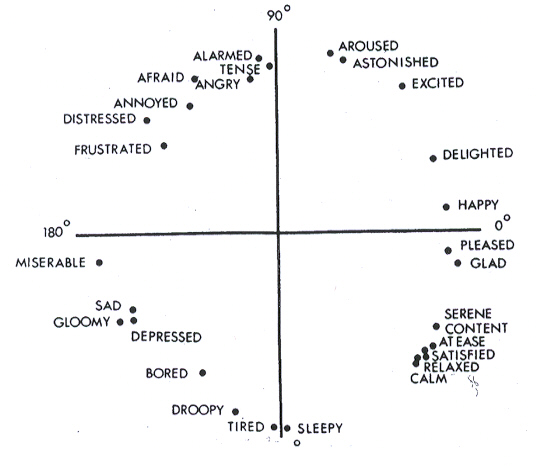


Figure 2-1: Russell’s circumplex model

1. Plutchik’s wheel of emotions: Robert Plutchik’s ‘Wheel of Emotions is yet another popular emotion model. It takes into consideration eight basic emotions, and the emotions obtained by varying the intensities of the basic ones. Additionally, various emotions in this model can be combined to obtain more complex emotions. The Plutchik’s model has been used in extracting opinions from text [14].

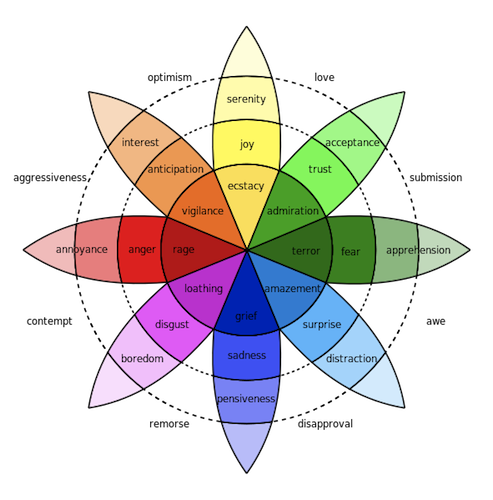


Figure 2-2: Plutchik’s emotion wheel

1. OCC Model: The OCC model [15] is similar in sense to many existing cognitive models in which it compares effect of an action (valence) on the desirability of the consequence of the action. The model separates and quantifies emotions based on their underlying strategic patterns of appraisal such as the consequences one would consider applying in any situation.  Such responses are derived from a person’s primary focus at the time, what affects them and how they react in response to the subject in question. Using these psychological models, we can map the text that we extract from the tweets, obtain the core emotion and also the intensity and decide which emotion is most significantly expressed in the tweet.

# Chapter 3

# System Analysis

## 3.1 Functional Requirements

The functional requirements for our project are as stated:

1. Emotion extraction and classification: The system must be able to accurately represent the emotions in the sentences based on our 8 point scale without returning false results for sentences with conflicting emotions. All emotions present in the sentence must be considered for the final result without ignoring the emotions that have little impact on the emotional impact of the sentence as a whole.
2. Elimination of slang: There are a large number of statements contain slang words, which are not directly understood by the computer. Hence, such slang words must be detected and replaced with something more easily understood.
3. Considering negation: Common negation words like ‘no’, ‘not’, ‘never’ etc. can change the meaning of the entire statement and hence must be considered while extracting emotions.

## 3.2 Non-Functional Requirements

The non-functional requirements of the system are as shown below:

1. Reliability: The system must be able to recover from failures. The system must not compromise the integrity of any other process.
2. Stability: The system must be stable without taxing the platform it is running on for maximum resource utilization.
3. Security: The system must not compromise the public data of the people who are mined for the information that is used as a sample set for the project. The system should also prevent changes to the sentences after pre-processing.
4. Performance: The time required to run the software should be low.

## 3.3 Specific Requirements

1. Hardware requirements: Since it is a purely software oriented project, there are no explicit hardware requirements. The project will be made to run on a standard computer. No additional hardware such as chips and other devices are required.
2. Software requirements: The software does not need any specific software configuration to run. However, some software and files need to be installed on the user’s machine for it to run.
3. The user must have a reliable internet connection and a browser because the input data will be pulled from Twitter using Twitter APIs.
4. Since the project would be coded in Python, the user must have a Python interpreter installed.
5. The Python Natural Language Toolkit (NLTK) should be downloaded on the user’s machine.

## 3.4 Use Case Diagram

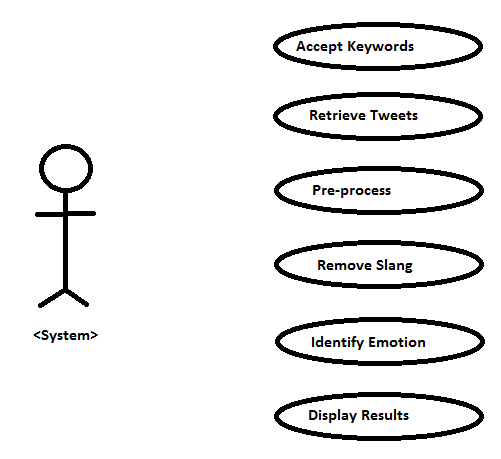


Fig 3-1: Use Case Diagram

The use case diagram above shows the main functions of the system along with the user's interaction with the system.

Each use case represents an action or a series of actions performed in order to achieve the task. The use cases described in our use case model are:

* Accepting the keywords entered by the user
* Retrieving relevant tweets from Twitter using the keywords and the Twitter APIs
* Pre-processing the tweets to remove extra information.
* Identification and replacement of slang words
* Extraction of emotions using Plutchik’s model
* Graphical representation of results

# Chapter 4

# Analysis Modeling

## 4.1 Data Modeling

We use CSV files in the implementation of the project. They are given as:

1. Tweets: We will be retrieving tweets from the internet using the Twitter APIs based on the keywords entered by the user. The tweets first need to be pre-processed, because it is important to remove extraneous information and also correct misspellings, slang and so on. Then, the polarity and the emotion expressed in the tweets will be extracted. Therefore, a separate file is created for each stage of the tweet collection and pre-processing:
   1. raw\_tweet: This column will contain the tweets as extracted by the Twitter API, without any changes.
   2. processed\_tweet: This column contains the tweet after pre-processing, spell checking and slang correction.
2. Slang: Due to the 140 character limit of Twitter, users tend to use a lot of slang and contractions while tweeting. These slang words cannot be excluded during the process of sentiment analysis, as they are often valuable indicators of sentiment. Hence, while cleaning the tweet, it is important to take these slang words into consideration and convert them into regular English so that it is feasible for the computer to process them. The attributes of this table are:
   1. slang\_word: This column holds the slang word.
   2. english\_word: This column contains the corresponding English form of the slang word under consideration.

We would also require the help of several word dictionaries. However, they need not be a part of our database, as exhaustive dictionaries are already available online.

## 4.2 Activity Diagram

The activity diagram shown below depicts the steps that need to be followed by the system to achieve the desired result. Each step that has to be carried out in the proposed model is shown in the activity diagram.

**Description of Activity Diagram**

* First, tweets are recovered from Twitter using Twitter API.
* Only English Tweets are recovered, tweets of any other language are discarded.
* A slang dictionary is used to eliminate slang and replace it by similar English words of equivalent meaning.
* A modified version of Plutchik’s model is implemented, which takes into account 8 basic, 8 complex and 3 outlier emotions.
* Each word in the tweet is then compared with a dictionary containing word similarity values. If the value is not present, then the word is compared with the UMBC Semantic Dictionaries by the use of their API.
* The final value of all the emotions is computed using cosine distances.

Contains slang?

Extract tweets

Remove slang

Store tweets

Import semantic dictionaries

Parse tweet word by word

Type of word

Ignore stop word

Compute semantic score

Invoke negation handling methods

Compute vector for emotion score

Compare with target vector

Are all tweets analyzed

Visualization

Yes

No

Stop word Normal word Negation

No

Yes

Yes

Fig 4-1: Activity Diagram

## 4.3 Functional Modeling

Keywords

Raw twitter data

Emotion classified tweets

Fig 4-2: Level 0 DFD

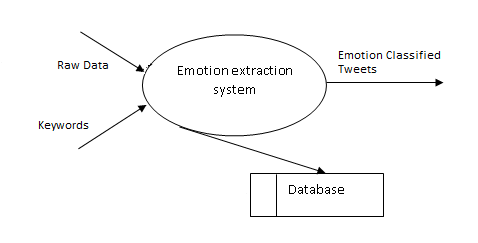


Fig 4-3: Level 1 DFD

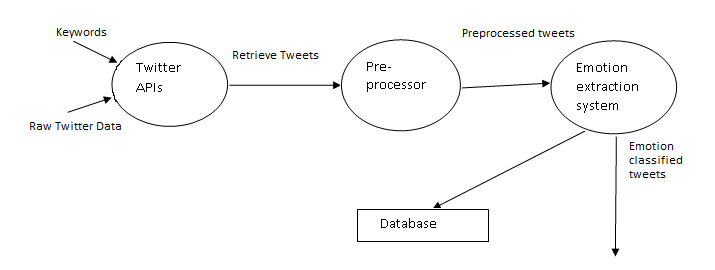


Fig 4-4: Level 2 DFD

The above data flow diagrams show how the data traverses through the system. Each level of the DFD gives a more detailed description about the system, its modules and the flow of the data across modules.

In the most abstracted level, the input is a set of keywords and the output is a classified set of related tweets. As we delve deeper, we can see the various modules, their individual inputs and outputs.

## 4.4 Timeline Chart

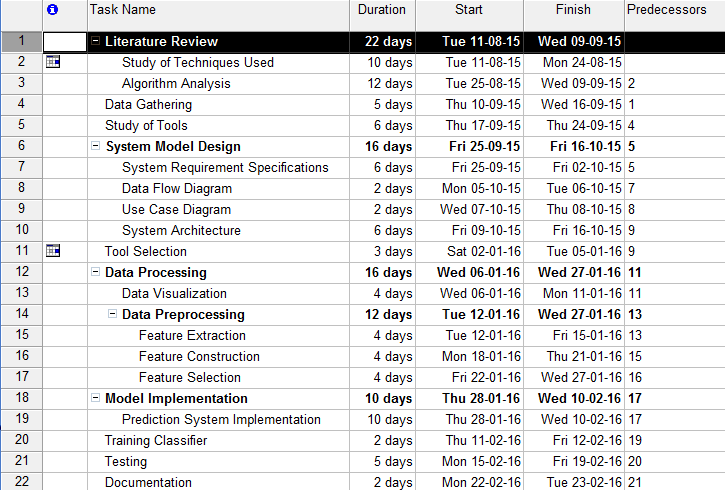


Figure 4-5: Work breakdown structure for the project.

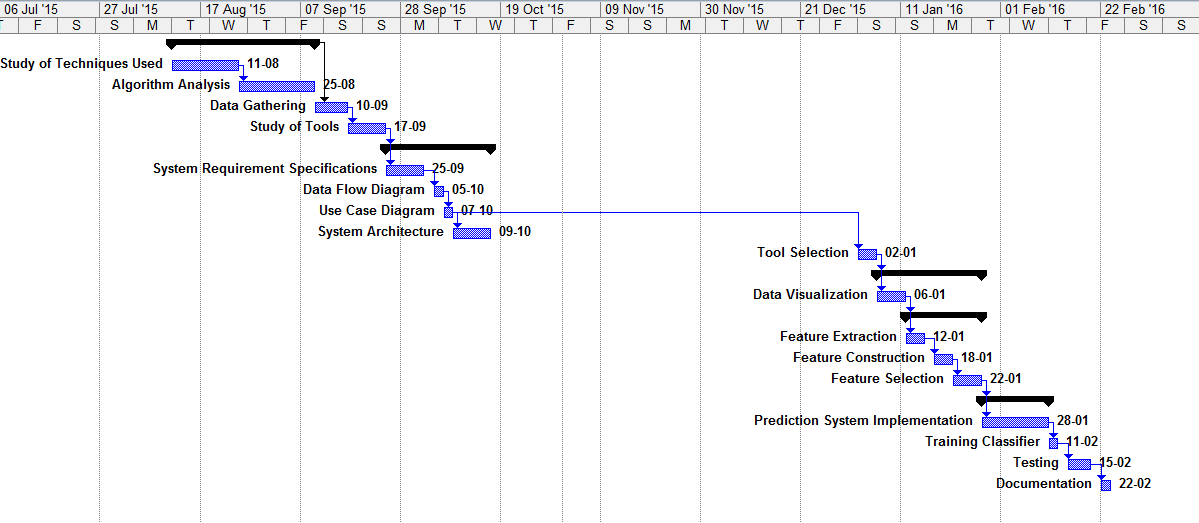


Fig 4-6: Timeline Chart

The figure above shows the Gantt chart of the project. It shows the most basic tasks involved, and the approximate time required to complete them along with the start and ending dates.

# Chapter 5

# Design

## 5.1 Architectural Design

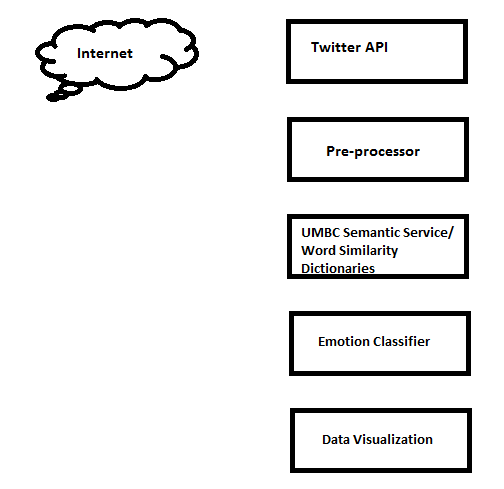


Fig 5-1: Architectural Design

The proposed model consists of the following blocks:

* The Twitter APIs are used for obtaining the data from Twitter, based on the keywords entered by the user.
* The tweets collected by the APIs need to be cleaned because they contain a lot of extraneous user, location and identification information that are irrelevant to us. The tweets also contain URLs, user handles and slang words that must be removed. That is done in the pre-processor block.
* The tweets are parsed word by word and the words are compared to the nineteen target emotions using the UMBC semantic service or our word similarity dictionaries.
* The cosine similarities between the target matrices and our resultant matrices are computed and the most dominant emotion is identified.
* Finally, the results are represented graphically.

## 5.2 User Interface Design

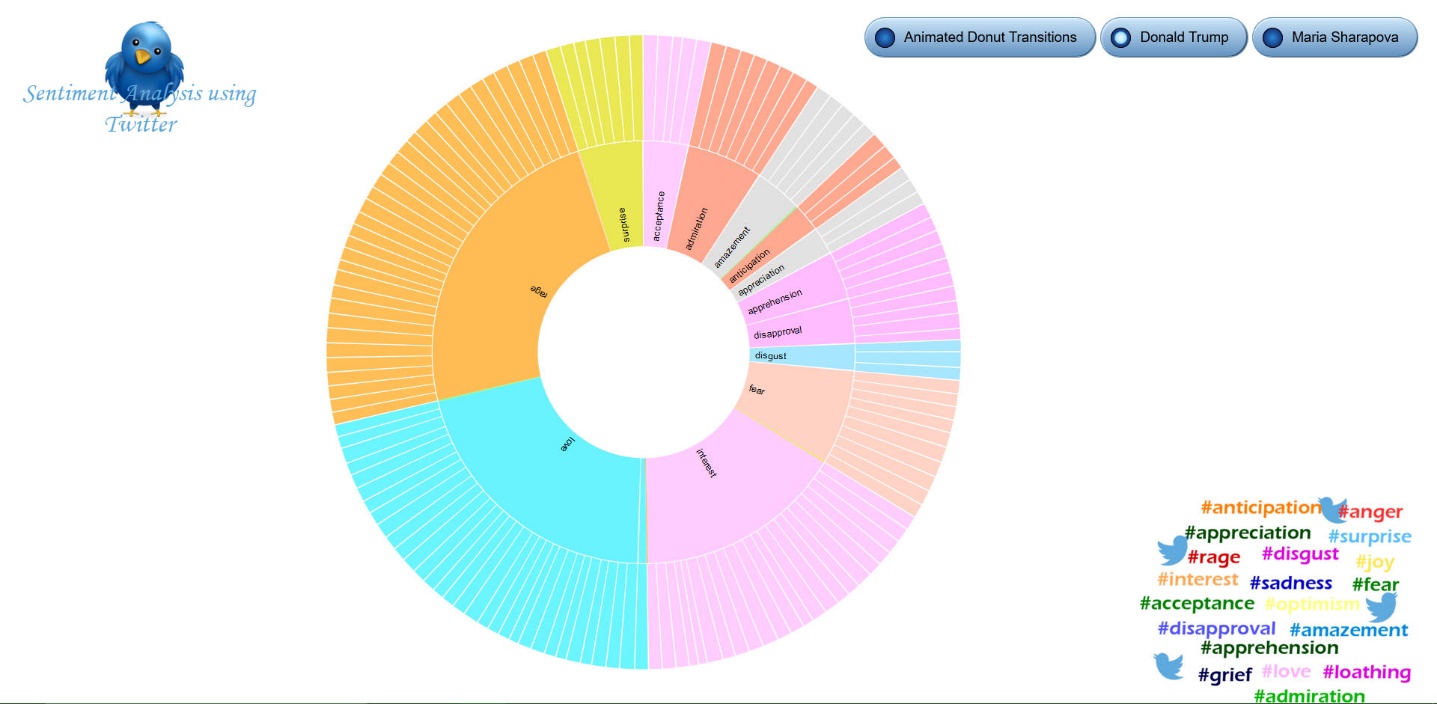


Fig 5-2: Main Visualization of Emotion Extraction

The figure 5-2 above shows the main representation of the results. The central wheel depicts each of our target emotions. The tweets that are of a similar final emotion are grouped together and are shown on the outer wheel.

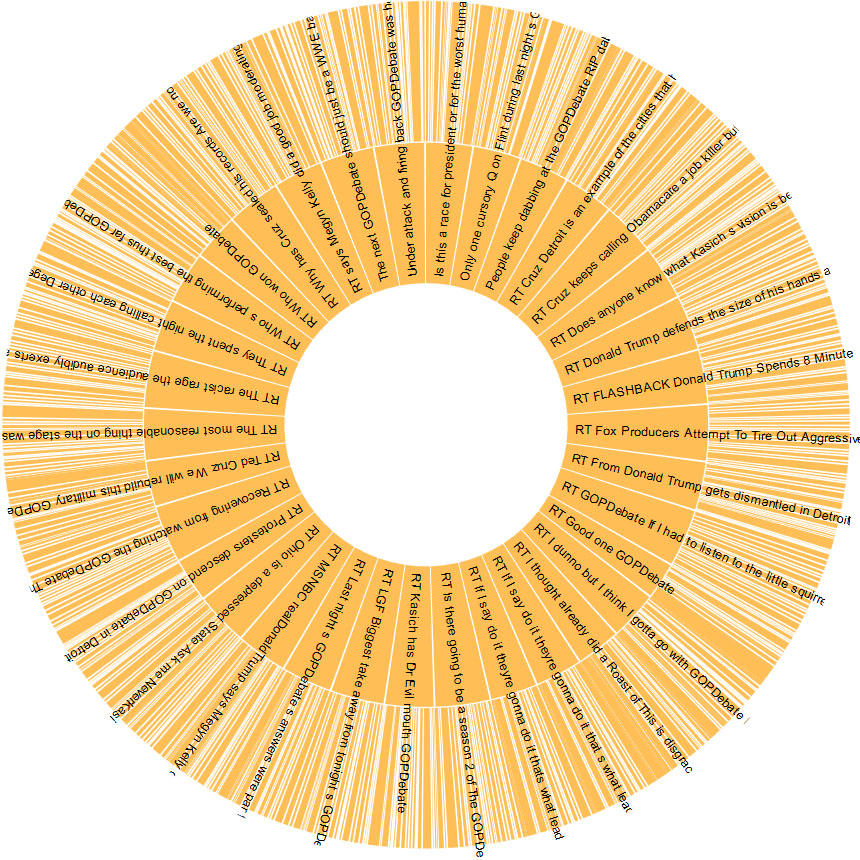


Fig 5-3: Level one of the wheel, showing tweets classified as the emotion “rage”.

Figure 5-3 shows us a more detailed breakdown of all the tweets contained within one resultant emotion.

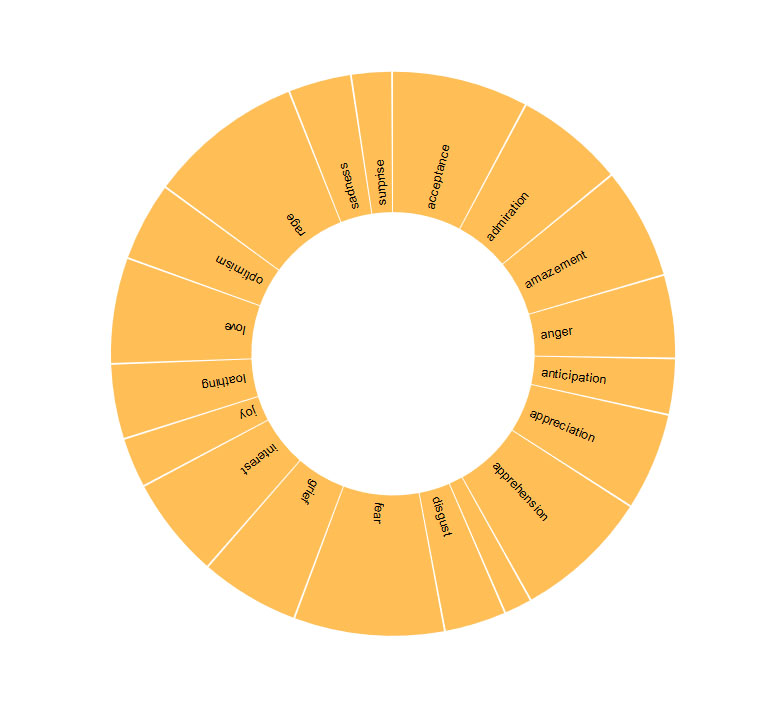


Fig 5-4: Level 2 of the Wheel, showing the emotions present in a single tweet.

Figure 5-4 is the most detailed representation of the tweets. Each section in the when shows us the magnitude of each emotion present. The section with the largest area is our final resultant emotion for that tweet.

# Chapter 6

# Implementation

## 6.1 Algorithms Used

The main algorithm used in the project is the Cosine Similarity Measure to determine the similarity between two vectors. Since there are nineteen target emotions, there are nineteen different target vectors, each corresponding to one emotion. On parsing the sentence word by word, we compare each word (if it is not a stop word) with the target emotions using the UMBC semantic Similarity Service available freely online. Then, we add the emotion-wise scores for all the relevant words in the tweet and store it in a vector A. This vector A is compared with all the nineteen target emotion vectors using the Cosine Similarity Measure. The emotion corresponding to the target vector for which the similarity with A is the maximum is our final extracted emotion for that tweet.

## 6.2 Working of the Project

The tweepy API is used to obtain data from Twitter. To use this, we first need to register our application with Twitter. We will then get an access token, an access token secret, a consumer key and a consumer secret key. Twitter supports the OAuth authentication mechanism and these tokens and keys are required for the authentication to be successful. The code for the authentication and tweet streaming is given below:

def getTweets(self):

l = StdOutListener()

s=Streamer()

auth = OAuthHandler(s.consumer\_key, s.consumer\_secret)

auth.set\_access\_token(s.access\_token, s.access\_token\_secret)

stream = Stream(auth, l)

#prompt user to enter a keyword of his/her choice

str=raw\_input("Enter a keyword: ")

stream.filter(track=[str],languages=['en'])

Tweepy provides a mechanism for getting tweets based on user entered keywords. It can also filter tweets by languages, as is shown above. The tweets are collected in the JSON format, with a lot of extra information such as date, time location, tweet ID, retweet information etc.

We need to first extract the tweet text from the JSON file because only that information is relevant to our project. Then, we clean the tweet to remove unwanted information from the tweet text itself. Such information includes user handles, URLs and extra punctuation.

Once this is done, we observe that tweets have a lot of slang present in them. Such slang is not understood by the computer. Hence, the slang must be replaced by their English equivalents which are better understood. Slang cannot be eliminated as they are also important sources of emotional information. For the slang replacement module, we have created a file in the CSV format which maps popular slang words to their English equivalents. We then parse the tweets word by word. If we encounter a slang words present in our list, we replace it using the equivalent found in our list. The code for the same is given below:

with open(filename) as slang\_file:

for line in slang\_file:

words = line.lower().split()

replaced = []

for y in words:

if y in self.data:

replaced.append(self.data[y])

else:

replaced.append(y)

text = " ".join(map(str,replaced))

We now create the final text file by removing duplicate tweets. The output of this process is the input to our emotion extraction phase.

In the next phase, the tweets are considered one at a time. Each tweet is parsed word by word. If a stop word is encountered, it is ignored. Else, a word is compared with the target emotion using either our pre-constructed dictionary or the UMBC semantic similarity service. If a common negation word is encountered, we tackle the effect of negation using one of the two approaches given below:

* Use the antonym of the word after the negation: Suppose the phrase is ‘not happy’, we include the effect of negation by using the antonym of happy, i.e. ‘sad’ for our distance matrix. The list of common words and their antonyms are generated using NLTK-WordNet using the following code:

from nltk.corpus import wordnet as wn

for i in wn.all\_synsets():

if i.pos() in ['a', 's']:

for j in i.lemmas():

if j.antonyms():

print j.name(), j.antonyms()[0].name()

* Use the contrasting emotion from the Plutchik’s wheel: In the Plutchik’s wheel of emotion that we are using, the emotion diagonally opposite to the one that we are considering it is its opposite. We observe that ‘joy’ is opposite ‘sadness’, ‘admiration’ is opposite ‘loathing’ and so on. Thus, on encountering a negation word, the word after it is mapped to the opposite emotion and stored in the result vector. For clarity, consider the following example. If the only two emotions we consider are ‘joy’ and ‘sadness’,
  + Target vector for joy=[1 0]
  + Target vector for sadness=[0,1]
  + Result vector of a word is [word similarity with joy, word similarity with sadness]
  + Suppose the tweet is “I am not happy.” Here, ‘happy’ occurs after a negation.
  + If the similarity of ‘happy’ with ‘joy’ is 0.8 and with ‘sadness’ is 0.1, the resultant vector becomes [0.1 0.8]
  + Therefore, this vector is more similar to the sadness target vector and hence would be classified as ‘Sadness’.

The emotion values of each word are then stored in a vector, which is added up and averaged when the entire tweet is parsed. This results in an emotional score vector per tweet, which is then compared to each of the target vector using the cosine similarity measure. The cosine similarity measure is used because it gives us the structural similarity between two vectors, rather than the distance between them in space. For each tweet, the resultant emotion is the target vector for which the cosine distance measure is the maximum. The tweet, resultant emotion and the cosine similarity values for each target emotion is stored in a CSV file.

Now, we need to represent our results in a graphical format. We use the bi-level partition graph provided by D3.js under the BSD license. For this, our data must be in the JSON format. The first level consists of our target emotions and the tweets grouped within them. The next level contains each tweet and their corresponding emotion values. Therefore, we go through the entire CSV file and create a corresponding data file in the JSON format.

Finally, we launch the result page using a python method, and the bi-level partition graph for the corresponding data set is generated.

# Chapter 7

# Testing

## 7.1 Test Cases

The following are the test cases we have used:

Table 7-1: Test cases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Number | Description | Expected Output | Actual Output | Result |
| 1. | The streaming program was given the keyword “Donald Trump” | A JSON file containing tweets related to Donald Trump | A JSON file containing tweets related to Donald Trump | Pass |
| 2. | The pre-processing program was given raw tweets | A file containing the tweet text with URLs and user handles removed | A file containing the tweet text with URLs and user handles removed | Pass |
| 3. | The slang removal program was given a set of tweet texts containing slang. | A text file that contains tweets with slang replaced with their English equivalents. | A text file that contains tweets with slang replaced with their English equivalents. | Pass |
| 4. | The emotion extraction program is given cleaned tweets as input for classification. | A CSV file containing the tweet, the main emotion and the cosine similarity values with each emotion. | A CSV file containing the tweet, the main emotion and the cosine similarity values with each emotion. | Pass |
| 5. | The visualization program is given the CSV file as input. | An HTML page with the results in a graphical format is generated. | An HTML page with the results in a graphical format is generated. | Pass |

## 7.2 Types of Testing Used

*Unit Testing:*

Each module of the project was created independently and is capable of functioning as a standalone program. Before the final integration testing, the modules were independently used to collect, process and evaluate data. The first module on data collection would simply collect information based on the keywords provided. In order to check the progress of the program, we would allow it to dynamically store the data into a text file that would constantly update as new information was found.

*Integration Testing:*

Integration testing is used when different modules are combined and it has to be tested as a whole. It is used to expose faults in the interaction between the integrated units. It is performed after unit testing, and after the various modules are integrated. We used the bottom up approach to integration testing.

Integration testing was required by us because of the various modules present in our application – for example, tweet retrieval, tweet preprocessing, and slang correction had to be integrated and tested to check whether all the modules are working correctly independently as well as together. Without performing integration testing, there could have been errors in the functioning of the entire program, even though the code worked properly as a stand-alone.

*Black Box Testing:*

Black box testing, also called Behavioral Testing, is a testing method in which the internal structure of the item being tested is not known to the tester. Incorrect/missing functions, data structure errors, performance errors and initialization and termination errors are identified by black box testing.

*White Box Testing:*

White box testing us a testing method in which the internal structure of the item is known to the tester. The tester chooses inputs through the various paths in the code and determines the appropriate output. White box testing goes beyond the user interface and focuses on the details of a system.

# Chapter 8

# Results and Discussions

To test the accuracy of the system, we have tried using various keywords entered by the user. Varying number of tweets, ranging from 50 to 500, has been collected per keyword and the emotion extraction was done on these tweets.

We have tabulated some of our results below:

Table 8-1: Summary of the results for different keywords

|  |  |  |
| --- | --- | --- |
| **Keyword with brief description** | **Number of tweets analyzed** | **Accuracy (%)** |
| Keyword: Donald Trump  Consists of tweets collected about Donald Trump after the GOP Debate held in March | 133 | 53.47 |
| Keyword: JNU  Consists of tweets collected after JNU made headlines in India in early March | 73 | 50.68 |
| Keyword: #indvsnz  Consists of tweets collected after India lost the T20 match to New Zealand during the World T20 Championship | 394 | 56.09 |
| Keyword: Maria Sharapova  Consists of the tweets collected after news reports of Sharapova indulging in substance abuse emerged | 68 | 72.05 |

Some of the tweets from the Maria Sharapova dataset are given below, with the final emotion.

Table 8-2: Samples of the Sharapova data set

|  |  |
| --- | --- |
| **Tweet** | **Emotion Detected** |
| Tennis to nominate a three person panel to discuss Sharapova s fate . Penalty depends on degree of fault . | Interest |
| Sharapova fights back in doping case Maria Sharapova denies she had been warned five times about the impending ban on meldonium and . . | Disapproval |
| Serena Williams Surprised amp Shocked By Maria Sharapova s Announcement | Surprise |
| RT You re our Maria and there s nothing to add LetMariaPlay IStandWithMaria | Admiration |
| RT Reaction to yesterday s GOP debate then Maria Sharapova s failed drug test . audio podcast | Disapproval |
| RT Once Maria Sharapova Asked Who is Sachin Tendulkar . She Was Banned From Tennis After Failing In Drug Test. Never Mess Wi | Disapproval |
| RT My theory about why there have been so many meldonium positives . Sharapova is one of 99 so far . | Interest |

The program gives a good accuracy when the tweets are more or less straightforward. It is largely able to handle the effects of negation. However, the accuracy dips when there is sarcasm, or when the words may have multiple meanings in different contexts.

# Chapter 9

# Conclusion and Future Scope

Sentiment Analysis is too broad a field to ever give us 100% accuracy, simply because human emotions are complex and multifaceted. There is always a certain margin of error based on interpretation and intent of the tweet, because every individual perceives information differently. This project is a small attempt at understanding how humans process emotions as well as understanding how this information could be incorporated in our technology in order to create a deeper feeling of “immersion”. Work in this direction is still underway, with new methods in development. Automatic emotion detection will continue to be challenged by slang, wrong spellings, references, emoticons, etc.

Information that can be gained through such analysis provides large avenues of public sentiment towards products, politics and events, making it a valuable source of consumer feedback. A small example of this can be gained from our use of this project to understand the views of people with respect to major events such as Donald Trump at a debate or a match between two cricket teams. The wide variety of opinions that we have unearthed show how people react to any experience, which is invaluable given the large domains it could fit into like finance, politics, products among others. Future challenges for this project include the detection of sarcasm, which is often employed to great effect on Twitter, along with higher accuracy. In future, it may also be extended to provide support for languages that are not English and learning from the information it collects.

# Chapter 10

# References

[1] S.M. Vohra and J.B. Teraiya, “A Comparative Study of Sentiment Analysis Techniques”, Journal of Information, Knowledge and Research in Computer Engineering, Volume 2- Issue 2, 2012

[2] P. Turney, “Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews”, Proceedings of the Association of Computational Linguistics, 2002

[3] Neethu M.S. and Rajasree R., “Sentiment Analysis in Twitter Using Machine Learning Techniques”, Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 2013

[4] Bhuta S., Doshi A., Dochi U. and Narvekar M., “A Review of Techniques for Sentiment Analysis of Twitter Data”, International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014

[5] Ahmad Kamal and Muhammad Abulaish, “Statistical Features Identification for Sentiment Analysis”, International Symposium on Computational and Business Intelligence, 2013

[6] Shuigui Huang, Wenwen Han, Xirong Que and Wendong Wang, “Polarity Identification of Sentiment Words based on Emoticons”, Ninth International Conference on Computational Intelligence and Security, 2013

[7] Erik Cambria, “An Introduction to Concept-Level Sentiment Analysis”, Advances in Soft Computing and its Applications, Volume 8266 of the series Lecture Notes in Computer Science, 2013

[8] Bradley, M.M. and Lang, P.J., “Affective norms for English words (ANEW): Stimuli, instruction manual and affective ratings”, Technical report C-1, Gainesville, FL. The Center for Research in Psychophysiology, University of Florida, 1999

[9] Cambria E., Havasi C. and Hussain A., “SenticNet 2: A Semantic and Affective Resource for Opinion Mining and Sentiment Analysis”, Proceedings of FLAIRS, Marco Island, 2012

[10] Bazzanella, C., “Emotions, language and context”, Weigand, E. (ed.) Emotion in Dialogic Interaction. Advances in the complex, pp. 59–76, Benjamins, Amsterdam, 2004

[11] Andrea Esuli and Fabrizio Sebastiani, “SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining”, LREC, 2006

[12] James Russell, “A Circumplex Model of Affect”, Journal of Personality and Social Psychology, Vol. 39 No. 6, 1980

[13] Healey and Ramaswamy, 2013. http://www.csc.ncsu.edu/faculty/healey/tweet\_viz/

[14] Dhanashri Chafale and Amit Pimpalkar, “Sentiment Analysis on Product Reviews Using Plutchik’s Wheel of Emotions with Fuzzy Logic”, ABHIYANTRIKI An International Journal of Engineering & Technology (AIJET), 2014

[15] Naseem Ahmadpour, “OCC model: application and comparison to the dimensional model of emotion”, International Conference on Kansei Engineering and Emotion Research, 2014.

# Appendix

## Credits

The following are the programs used for the Sentiment Analysis project. They have been credited with their original licenses included.

1. Scikit-learn: Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Édouard Duchesnay. Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research, 12, 2825-2830 (2011). Licensed under BSD License.
2. Scipy: Jones E, Oliphant E, Peterson P, et al. SciPy: Open Source Scientific Tools for Python, 2001-, http://www.scipy.org/ [Online; accessed 2016-04-11]. Copyright © 2003-2013 SciPy Developers. ® All rights reserved.
3. Tweepy: Copyright © 2001, 2002 Enthought, Inc. All rights reserved.
4. NLTK: Bird, Steven, Edward Loper and Ewan Klein (2009), Natural Language Processing with Python. O’Reilly Media Inc.
5. WordNet: Princeton University "About WordNet." WordNet. Princeton University. 2010. <http://wordnet.princeton.edu>
6. UMBC Semantic service: Lushan Han, Abhay L. Kashyap, Tim Finin, James Mayfield and Johnathan Weese, UMBC\_EBIQUITY-CORE: Semantic Textual Similarity Systems, Proc. 2nd Joint Conf. on Lexical and Computational Semantics, Association for Computational Linguistics, June 2013.
7. D3JS: Library released under BSD license. Copyright 2015 Mike Bostock.

# Publications

[1] Dyuti Shukla, Mihika Shah, Prerna Parmeshwaran and Prof. Kiran Bhowmick, “A Proposed Solution for Sentiment Analysis on Tweets to Extract Emotions from Ambiguous Statements”, International Journal of Engineering Research & Technology, Volume. 4 - Issue. 11 , November – 2015. E-ISSN: 2278-0181, DOI: http://dx.doi.org/10.17577/IJERTV4IS110185

# Acknowledgements

A final year project is a very important component of any student’s life and to perform a task of such a magnitude, a lot of support from teachers and mentors is required. Therefore, we are extremely grateful to our principal **Dr. Hari Vasudevan** and the head of the department of Computer Engineering, **Dr. Narendra M. Shekokar** for their support and for providing us with all the facilities required by us to work on our final year project.

We would also like to acknowledge our project guide **Prof. Kiran Bhowmick** for her help. Her vast knowledge and experience have been instrumental in helping us start our project in the best possible manner.

Prerna Parmeshwaran

Mihika Shah

Dyuti Shukla