Chapter 1

Introduction

Twitter data analysis for current events, companies, products and people thus, leading a way to shape history. Data modeling and analysis is the application of natural language processing, computational linguistics, and text analytics to identify and extract subjective information (based on emotion as opposed to reason) in source materials (unstructured texts). The sentiment found within comments, feedback or critiques provide useful indicators for many different purposes. These sentiments can be categorized either into two categories: positive and negative; or into an n-point scale, e.g., very good, good, satisfactory, bad, very bad. In this respect, a sentiment analysis task can be interpreted as a classification task where each category represents a sentiment. With the growing availability and popularity of opinion-rich resources such as online review sites and personal blogs, new opportunities and challenges arise as people now can, and do, actively use information technologies to seek out and understand the opinions of others. The sudden eruption of activity in the social networking domain has drawn the attention of analysts, social media as well as general public to area of sentiment analysis to extract invaluable information from public opinion. Twitter is a pioneer in the social networking domain. It is a popular online micro-blogging service launched in 2006. Twitter has specifically 517 million accounts as of July 1, 2012, positioning it as the second-biggest social networking site. Almost 300 million new tweets are generated every day (Paris-based analyst group Semiocast). Users on Twitter write tweets up to 140 characters to tell others about what they are doing and thinking. Because almost all tweets are public, these rich data offer new opportunities for doing research on data mining and natural language processing. Thus, data miners use Twitter as the source of its opinionated data. One way to perform Twitter data analysis is to directly exploit traditional Sentiment Analysis methods (Pang and Lee 2007). However, tweets are quite different from other text forms like product reviews and news articles. Firstly, tweets are often short and ambiguous because of the limitation of characters. Secondly, there are more misspelled words, slang, modal particles and acronyms on Twitter because of its casual form. Thirdly, a huge amount of unlabeled or noisy labeled data can be easily downloaded through

Twitter API. We propose a method for modeling the data and analyzing it to solve the complex real world problem solving.

1.1 Motivation

Companies are mining the social web to build dossiers on people. Information posted publicly on blogs, Facebook, Twitter, forums and other sites is being used for analysis. Polonetsky, the famous American lawyer and internet privacy expert, said aggregators like “Rapleaf Inc” will collate information about individuals and sell it to companies that want to learn about those customers and what they do online [1]. Entities such as airlines, politicians, and even non-profits can use this data for finding new customers or targeting products to existing ones. Financial services companies such as banks and lenders are also using the same data mining services for marketing purposes and to make lending decisions.

1.2 Aims and Objectives

Real Time Data is analyzed and modeled. The Analysis made can be represented using graphical tools like Pie Charts, Graphs, Nodes. The system will be intuitive and easy enough to use so that users with minimal instruction can use it and analyze the real time data. The system can be integrated for multiple real time problems which uses following steps [2].

- Association - looking for patterns where one event is connected to another event
- Sequence or path analysis - looking for patterns where one event leads to another later event
- Classification - looking for new patterns
- Clustering - finding and visually documenting groups of facts not previously known
- Forecasting - discovering patterns in data that can lead to reasonable predictions about the future

1.3 Problem Definition

This project is based on analysis and modeling of real time data obtained from social networking sites. The data is in various formats which is processed to find out the real time topics of discussion. Such topics are displayed in the form of simple graphs. This helps to find out the reaction of people to various events and topics using which business tasks can be conducted.

Example:

If a company wants to review its recently launched product, then the application will first collect the tweets which are related to the company and the product. All the tweets apart from company name and product are irrelevant for the application. Then we will divide the tweets so that we will get the number of people talking in favor and number of people talking against that product. Depending upon these numbers we can generate a pie chart and show the result in percentage about the product's popularity.

Chapter 2

Literature Review

Tree kernel model for analysis of Twitter Data

The Tree Kernel model is a new approach to analysis of tweets proposed in the form of a paper from the Computer Science department of Columbia University.

The following figure shoes the tree which is for analysing a particular tweet mentioned below it.

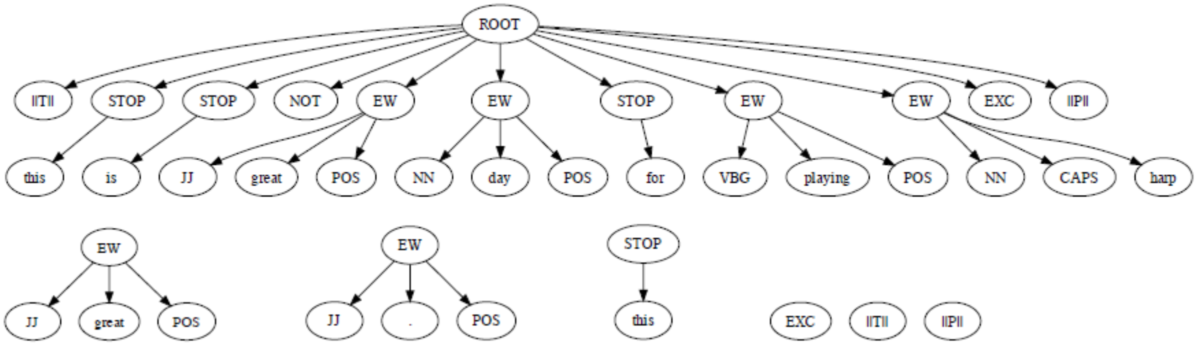


Figure 1: Tree kernel for a synthesized tweet: "@Fernando this isn't a great day for playing the HARP! :)"

The tree is starts with the root and every word in the sentence is classified into categories as follows:

||T|| - target, ||P|| - positive emoticon, ||N|| - negative emoticon,

POS – positive sentiment, STOP – stop words, EW – English word, NN – noun, VBG – verb

The paper presents the results of experiments on various models like Unigram Model, Tree Kernel model and feature based model as well as a combination of these models to achieve a better accuracy of analysis.

The Tree Kernel model suggested a way of sorting different types of data differently. Emoticons are classified into 5 categories from Extremely Positive to Extremely Negative and rest of the words into 3 categories. They have used Dictionary of Affect in Language (DAL) (Whissel, 1989) and extend it using WordNet to calculate polarity of the words. Words with polarity less than 0.5 are taken as negative, higher than 0.8 as positive and the rest as neutral.

Another set of Senti-features is also added to the Unigram model and Kernel model baseline to improve its performance. This upscales its average accuracy by about 3 to 4% over the traditional base models. Thus, a significant improvement is achieved.

This method of classification can be very useful for our purpose of classification of tweets. We aim at classifying them primarily based on topic. Whether the topic and its effect or reaction from the people is positive or negative is at the discretion of the user [3].

Related work

As stated, much emphasis is on the public reaction or public sentiment of a current trending topic. Such trends are used to make strategic and business decisions. One way to perform Twitter data analysis is to directly exploit traditional Sentiment Analysis methods (Pang and Lee 2007). However, tweets are quite different from other text forms like product reviews and news articles. Firstly, tweets are often short and ambiguous because of the limitation of characters. Secondly, there are more misspelled words, slang, modal particles and acronyms on Twitter because of its casual form. Thirdly, a huge amount of unlabeled or noisy labeled data can be easily downloaded through Twitter API. This paper proposes a method for modeling the data and analyzing it to solve the complex real world problem solving.

The methods used specifically for the analysis of Twitter data that we have referred are the papers Sentiment Analysis of Twitter Data and End-To-End Sentiment Analysis of Twitter Data. The first paper introduces the Tree-Kernel model for the analysis of data. It shows a way of cleaning the raw data, processing it to assign a polarity to keywords which suggest the emotion behind it and results of this new model. This was a major basis for the method of classification that is used in this paper. The second one compares PNP and an Objective PNP model to find the better one in classification of tweets into Positive, Negative, Neutral and Objective tweets.

Chapter 3

Analysis

The Unified Modeling Language (UML) is a general-purpose modeling language in the field of software engineering, which is designed to provide a standard way to visualize the design of a system.

3.1 Requirements

The requirements are primarily characterized in two following classes:

Functional Requirements:

- Extract data from the social networking website Twitter
- Evaluate the tweets and filter them to remove redundancies
- Clean the tweets to remove metadata
- Perform good sentiment analysis
- Provide graphical solution from extracted data

Non Functional Requirements:

- Quick response or low response time
- Accuracy of the response given to the user
- Correct answer in readable manner
- Minimum errors
- User friendly and easy user interface

3.2 Use Case Diagram

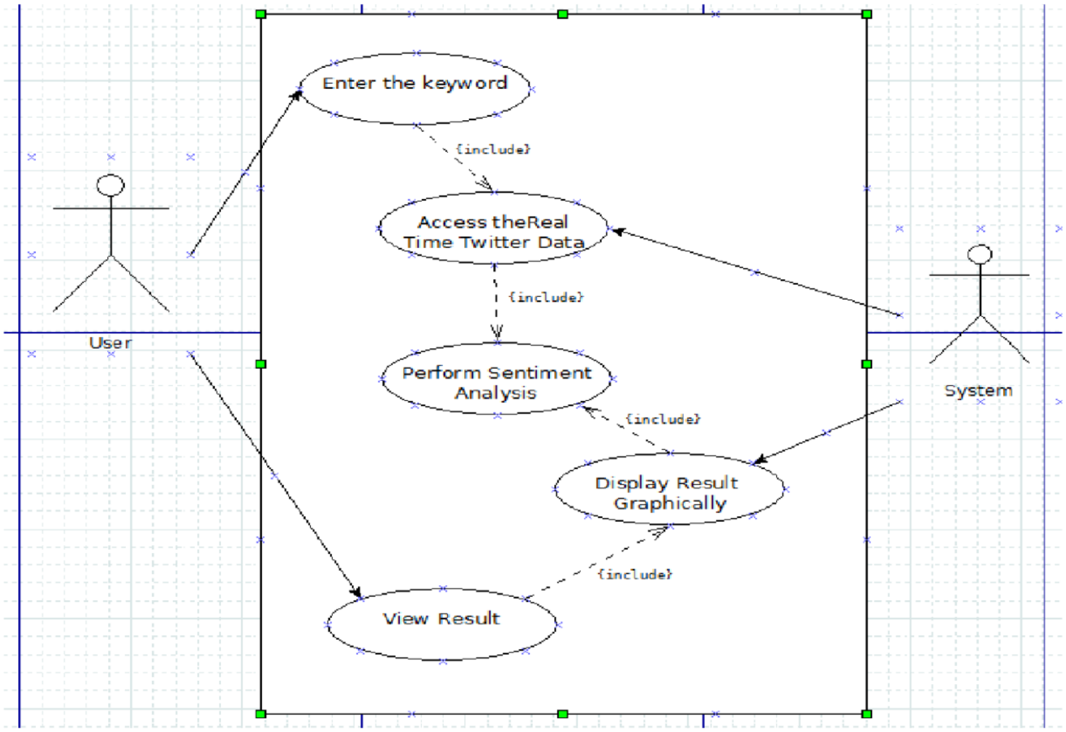

Enter the keyword
{include}
Access theReal Time Twitter Data
{include}
Perform Sentiment Analysis
{include}
Display Result Graphically
{include}
View Result
User
System

Fig. 2 Use Case Diagram

The above diagram is a Use Case diagram for Real Time data Analyzer and Modeller based on Twitter data. There are two actors in this system, the user and the system itself.

The primary function of the user is to provide the system with a keyword. The keyword will be searched in the Twitter data obtained by using the Twitter APIs. The user can then view the output which will be represented in a graphical way.

The system performs operations by searching the keyword in the Twitter data. All the tweets containing the keyword is obtained. The data obtained is then cleaned. After cleaning the data, sentiment analysis is carried out on the data. On the basis of analysis, the system classifies the data into different categories. These categories are then represented in a graphical way.

3.3 Sequence Diagram

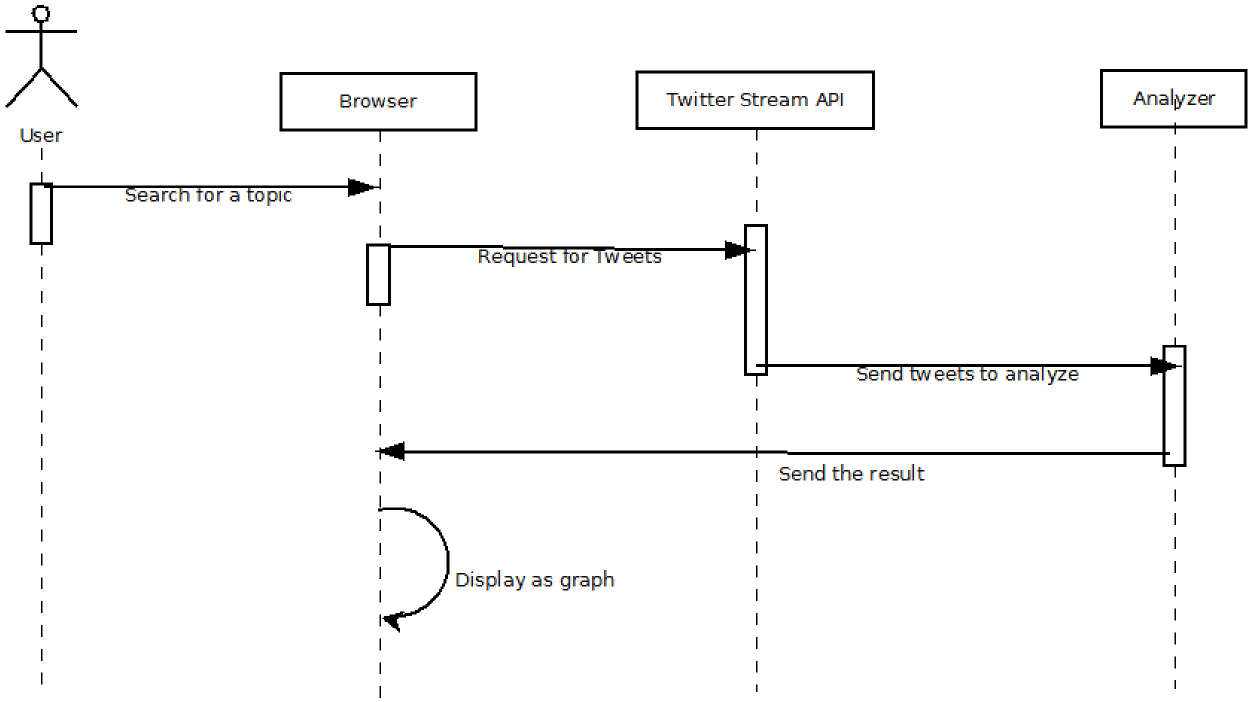


Fig. 3 Sequence diagram

A sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. A sequence diagram shows object interactions arranged in time sequence.

The above diagram shows the sequential working of the project. The user who wishes to search for a particular topic inputs the word into the browser. This is the keyword for the Twitter API which searches its tweets in real time to find the relevant tweets by all its users. These tweets are returned by the API and they are stored in a CSV file. They are in the JSON format. It contains the tweeted text, user name, location of the tweet, retweets, followers etc. Some of this data is ignored and only the important part, that is, the text of the tweet is taken for analysis. These input tweets are tested according to the training data for their positive or negative sentiment. The sentiment that emerges is shown in the form of a graph in the browser.

hapter 4

Proposed Methodology

We propose to implement the Modeler by analyzing Real Time Twitter Data using sentiment analysis. The system will accept an input from the user. Using the keyword entered by the user, the system will search for the tweets containing that keyword. The data received may contain huge amount of unlabeled or noisy labeled data. Therefore, it is necessary to clean the data before applying sentiment analysis on it. The output is the graphical representation of the data classified on the basis of sentiment.

As mentioned above, almost 300 million new tweets are generated every day. The tweets are based on different topics. At the same time the sentiment behind the tweet is different. The words in the tweets help us to determine whether it can be classified as a good tweet or bad tweet. For example; “Luck is a dividend of sweat. The more you sweat, the luckier you get” is classified as a good tweet as it encourages a person to make hard efforts to be successful. At the same time, “Insults are pouring down on me as thick as hail” is classified under a bad tweet as it hurts a person’s sentiment. Thus, it is necessary for the system to classify between the good and bad tweet by recognizing the idea behind the tweet.

The process can be mainly divided into 4 steps.

1) Data Gathering: Twitter data can be gathered through twitter streaming API. We continuously collect the tweets from the Twitter depending on the given Keyword and in the specified time span. We can also specify the language in which we want the tweets.

2) Data Processing: In this we clean the data by removing irrelevant and unwanted data. If we specify a keyword it is possible that the keyword is used in some other context which is not required for us. Cleaning of data is followed by sentiment analysis on the tweets. After sentiment analysis we can retrieve knowledge from the tweets.

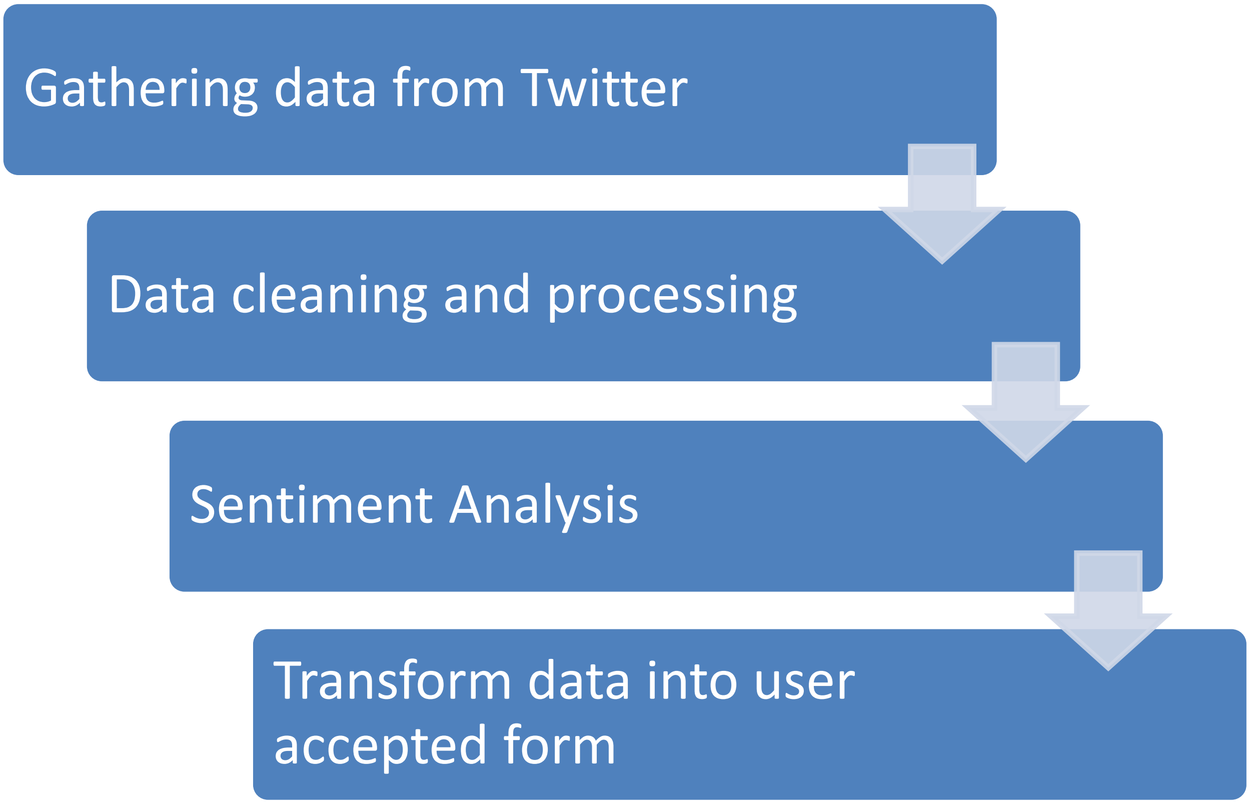

Gathering data from Twitter
Data cleaning and processing
Sentiment Analysis
Transform data into user accepted form

Fig 4. Proposed System

3) Sentiment Analysis: Sentiment analysis on the processed tweets involves categorizing the topic into various emotions. We are classifying the tweets into positive and negative tweets using NLTK (Natural Language Tool Kit) algorithm.

4) Transformation of data into user accepted form: Instead of showing text message we can display the message in the graphical format which is visually pleasing and helps user to understand quickly. We are presenting the final output in 3D donut graph and it keeps on changing after getting the streaming data from Twitter.

Chapter 5

Study of Technologies to Be Used

5.1 HTML

HyperText Markup Language (HTML) is the main markup language for displaying web pages and other information that can be displayed in a web browser. HTML is written in the form of HTML elements consisting of *tags* enclosed in angle brackets (like <html>), within the web page content. HTML tags most commonly come in pairs like <h1> and </h1>, although some tags, known as *empty elements*, are unpaired, for example <img>. The first tag in a pair is the *start tag*, the second tag is the *end tag* (they are also called *opening tags* and *closing tags*). In between these tags web designers can add text, tags, comments and other types of text-based content.

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts in languages such as JavaScript which affect the behavior of HTML webpages.

Web browsers can also refer to Cascading Style Sheets (CSS) to define the appearance and layout of text and other material. The W3C, maintainer of both the HTML and the CSS standards, encourages the use of CSS over explicit presentational HTML markup.

5.2 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to style web pages and interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. CSS is a cornerstone specification of the web and almost all web pages use CSS style sheets to describe their presentation.

CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for tableless web design).

CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. While the author of a document typically links that document to a CSS file, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified. However if the author or the reader did not link the document to a specific style sheet the default style of the browser will be applied.

CSS specifies a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called cascade, priorities or weights are calculated and assigned to rules, so that the results are predictable.

5.3 Javascript

Javascript (sometimes abbreviated JS) is a prototype-based scripting language that is dynamic, weakly typed and has first-class functions. It is amulti-paradigm language, supporting object-oriented, imperative, and functional programming styles.

Javascript is a scripting language that will allow you to add real programming to your webpages.
You can create small application type processes with javascript, like a calculator or a primitive game of some sort.

However, there are more serious uses for javascript:

- Browser Detection

  Detecting the browser used by a visitor at your page. Depending on the browser, another page specifically designed for that browser can then be loaded.

- Cookies

  Storing information on the visitor's computer, then retrieving this information automatically next time the user visits your page. This technique is called "cookies".

- Control Browsers

  Opening pages in customized windows, where you specify if the browser's buttons, menu line, status line or whatever should be present.

- Validate Forms

  Validating inputs to fields before submitting a form.

  An example would be validating the entered email address to see if it has an @ in it, since if not, it's not a valid address.

5.4 Python

Python is a widely used general-purpose, high-level programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C. The language provides constructs intended to enable clear programs on both a small and large scale.

Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard library.

Like other dynamic languages, Python is often used as a scripting language, but is also used in a wide range of non-scripting contexts. Using third-party tools, such as Py2exe, or Pyinstaller, Python code can be packaged into standalone executable programs. Python interpreters are available for many operating systems.

Python is a multi-paradigm programming language: object-oriented programming and structured programming are fully supported, and there are a number of language features which support functional programming and aspect-oriented programming (including by metaprogramming[29] and by magic methods). Many other paradigms are supported using extensions, including design by contract and logic programming.

Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. An important feature of Python is dynamic name resolution (late binding), which binds method and variable names during program execution.

The design of Python offers only limited support for functional programming in the Lisp tradition. The language has map(), reduce() and filter() functions, comprehensions for lists, dictionaries, and sets, as well as generator expressions.[34] The standard library has two modules (itertools and functools) that implement functional tools borrowed from Haskell and Standard ML.

The core philosophy of the language is summarized by the document "PEP 20 (The Zen of Python)", which includes aphorisms such as:

- Beautiful is better than ugly
- Explicit is better than implicit
- Simple is better than complex
- Complex is better than complicated
- Readability counts

Rather than requiring all desired functionality to be built into the language's core, Python was designed to be highly extensible. Python can also be embedded in existing applications that need a programmable interface. This design of a small core language with a large standard library and an easily extensible interpreter was intended by Van Rossum from the very start because of his frustrations with ABC (which espoused the opposite mindset).

5.5 JSON (JavaScript Object Notation)

JSON or **JavaScript Object Notation** is a text-based open standard designed for human-readabledata interchange. It is derived from the JavaScript scripting language for representing simple data structures and associative arrays, called objects. Despite its relationship to JavaScript, it is language-independent, with parsers available for many languages.

The JSON format is often used for serializing and transmitting structured data over a network connection. It is used primarily to transmit data between a server and web application, serving as an alternative to XML.

Why JSON over XML?

Extensible Markup Language (XML) is a text format derived from Standard Generalized Markup Language (SGML). Compared to SGML, XML is simple. HyperText Markup Language

(HTML), by comparison, is even simpler. Even so, a good reference book on HTML is an inch thick. This is because the formatting and structuring of documents is a complicated business.

Most of the excitement around XML is around a new role as an interchangeable data serialization format. XML provides two enormous advantages as a data representation language:

1. It is text-based.
2. It is position-independent.

These together encouraged a higher level of application-independence than other data-interchange formats. The fact that XML was already a W3C standard meant that there wasn't much left to fight about (or so it seemed).

Unfortunately, XML is not well suited to data-interchange, much as a wrench is not well-suited to driving nails. It carries a lot of baggage, and it doesn't match the data model of most programming languages. When most programmers saw XML for the first time, they were shocked at how ugly and inefficient it was. It turns out that that first reaction was the correct one. There is another text notation that has all of the advantages of XML, but is much better suited to data-interchange. That notation is JavaScript Object Notation (JSON).

The most informed opinions on XML (see for example xmlsuck.org) suggest that XML has big problems as a data-interchange format, but the disadvantages are compensated for by the benefits of interoperability and openness.

JSON promises the same benefits of interoperability and openness, but without the disadvantages.
Let's compare XML and JSON on the attributes that the XML community considers important.

- Simplicity

XML is simpler than SGML, but JSON is much simpler than XML. JSON has a much smaller grammar and maps more directly onto the data structures used in modern programming languages.

- Extensibility

JSON is not extensible because it does not need to be. JSON is not a document markup language, so it is not necessary to define new tags or attributes to represent data in it.

- Interoperability

JSON has the same interoperability potential as XML.

- Openness

JSON is at least as open as XML, perhaps more so because it is not in the center of corporate/political standardization struggles.

Chapter 6

Design

Design is the creation of a plan or convention for the construction of an object or a system. Software design is the process by which an agent creates a specification of a software artifact, intended to accomplish goals, using a set of primitive components and subject to constraints.

Designing Principles:

- **The design should be traceable to the analysis model.** Because a single element of the design model often traces to multiple requirements, it is necessary to have a means for tracking how requirements have been satisfied by the design model.

- **The design should exhibit uniformity and integration.** A design is uni- form if it appears that one person developed the entire thing. Rules of style and format should be defined for a design team before design work begins. A design is integrated if care is taken in defining interfaces between design components.

- **The design should be structured to accommodate change.** The design concepts discussed in the next section enable a design to achieve this principle.

- **Design is not coding, coding is not design.** Even when detailed procedural designs are created for program components, the level of abstraction of the design model is higher than source code. The only design decisions made at the coding level address the small implementation details that enable the procedural design to be coded.

6.1 Analyzer Training

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning.

Natural Language Processing with Python provides a practical introduction to programming for language processing. Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analyzing linguistic structure, and more. A new version with updates for Python 3 and NLTK 3 is in preparation.

Positive Tweets:

```
('I love this car', 'positive'),
 ('This view is amazing', 'positive'),
 ('I feel great this morning', 'positive'),
 ('I am so excited about the concert', 'positive'),
 ('He is my best friend', 'positive'),
 ('Success is nothing without someone you love to share it with', 'positive'),
 ('And there the grass grows soft and white And there the sun burns crimson bright', 'positive'),
 ('Staying positive is all in your head', 'positive'),
 ('Everyday is a new opportunity to make someone smile', 'positive'),
 ('A dream is the bearer of a new possibility the enlarged horizon the great hope', 'positive'),
 ('The best way to destroy an enemy is to make him a friend', 'positive'),
 ('Find something you love to do and you’ll never have to work a day in your life', 'positive'),
 ('Whatever you are be a good one', 'positive'),
 ('The future depends on what we do in the present', 'positive'),
 ('Never regret If it’s good it’s wonderful If it’s bad it’s experience', 'positive'),
 ('Everyday is a gift that’s why they call it the present', 'positive'),
 ('God never shuts a door without opening a window', 'positive'),
 ('If you are breathing you have hope', 'positive'),
 ('There are always flowers for those who want to see them', 'positive'),
 ('I have found that if you love life life will love you back', 'positive'),
 ('Our greatest glory is not in never falling, but in rising every time we fall', 'positive'),
 ('The secret to success is to start from scratch and keep on scratching', 'positive'),
 ('Blessed are those who can give without remembering and take without forgetting', 'positive'),
 ('To a brave man good and bad luck are like his left and right hand He uses both', 'positive'),
 ('Even if you’re on the right track you’ll get run over if you just sit there', 'positive'),
 ('The door to happiness opens outward', 'positive'),
 ('Optimists are nostalgic about the future', 'positive'),
 ('Hope is faith holding out its hand in the dark', 'positive'),
 ('Once you choose hope anything\'s possible', 'positive'),
 ('You must start with a positive attitude or you will surely end without one', 'positive'),
 ('Some days there won\'t be a song in your heart Sing anyway', 'positive'),
 ('I\'m doing very good today How are you', 'positive'),
 ('Toughness is in the soul and spirit not in muscles', 'positive'),
 ('I have never met a man so ignorant that I couldn\'t learn something from him', 'positive'),
 ('I like rising sun', 'positive'),
 ('Positive anything is better than negative thinking', 'positive'),
 ('What a beautiful ride it was', 'positive'),
 ('The dinner was delicious', 'positive'),
 ('Happy is he who dares courageously to defend what he loves', 'positive'),
 ('Take time to laugh it is the music of the soul', 'positive'),
 ('Politeness is to human nature what warmth is to wax', 'positive'),
 ('Make your optimism come true', 'positive'),
 ('Happiness is an attitude we either make ourselves miserable or happy and strong', 'positive'),
 ('You are such a genius glamorous and genuine at the same time', 'positive'),
 ('A positive attitude may not solve all your problems', 'positive'),
 ('Hard work toil and patience are key to success', 'positive'),
 ('Everything is going to be alright', 'positive'),
 ('Today is gonna be a good day', 'positive'),
 ('Tomorrow will be better', 'positive'),
 ('It\'s nice to watch the friendship developing between both of you', 'positive'),
 ('I love when someone holds the door for me', 'positive'),
 ('Smile it keeps people wondering', 'positive'),
 ('Congratulations Great job', 'positive'),
 ('live happy live free', 'positive'),
 ('All of us are born with a good amount of courage', 'positive'),
 ('When I becomes We even illness becomes wellness', 'positive'),
 ('love yourself', 'positive'),
 ('listen to Inspirational and motivational stories', 'positive'),
 ('Live out of your imagination not your history', 'positive'),
 ('I admire your courage and loyalty', 'positive'),
 ('you have reached the summit', 'positive'),
 ('my best wishes are always with you', 'positive')]
```

Negative Tweets:

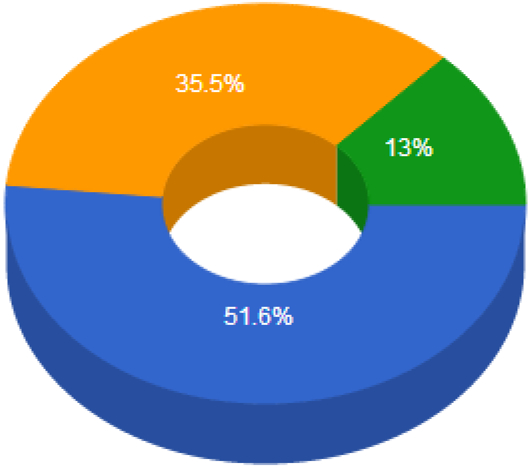
```
[('I do not like this car', 'negative'),
('This view is horrible', 'negative'),
('I feel tired this morning', 'negative'),
('I am not looking forward to the concert', 'negative'),
('He is my enemy', 'negative'),
('This band is awful', 'negative'),
('This is terrible', 'negative'),
('They make me feel bad', 'negative'),
('This band are not that great', 'negative'),
('This is absolutely atrocious', 'negative'),
('I dislike them', 'negative'),
('This band are the worst', 'negative'),
('What the hell is this crap', 'negative'),
('This band should die', 'negative'),
('I hate this band', 'negative'),
('This band make me want to die', 'negative'),
('My ears are dying', 'negative'),
('They suck they are terrible', 'negative'),
('In a closed mouth flies do not enter', 'negative'),
('it shits me', 'negative'),
('my poor little angel', 'negative'),
('teddy in the case BITCHES', 'negative'),
('I bet you get bullied a lot', 'negative'),
('I don\'t even like you we can stop pretending to be friends now', 'negative'),
('Acting like a slut does not make you cute', 'negative'),
('If you guys do not want me to be mean to you', 'negative'),
('DO YOU NOT REALIZE WE ALL WANT TO MURDER YOU RIGHT NOW', 'negative'),
('I mean this in the best way possible, but I legitimately can not stand you', 'negative'),
('You probably get left when you go out with your friends because they all hate you', 'negative'),
('You are not as bad as people say.  You are much worse', 'negative'),
('Please keep talking  Yawning means we are interested', 'negative'),
('I don\'t get bitter just better', 'negative'),
('I have looked every where but I just can\'t find anyone who cares', 'negative'),
('Is that meant to hurt my feelings You better try harder next time', 'negative'),
('I may look calm but in my head I have killed you three times', 'negative'),
('You are only young once but you will be immature forever', 'negative'),
('If you are going to be two faced at least make one of them pretty', 'negative'),
('Im glad to see you are not letting your education get in the way of your ignorance', 'negative'),
('I see you looking my direction, you are going to envy my perfection', 'negative'),
('I don\'t know what is making you so dumb, but it is working', 'negative'),
('People like you make people like me look even better', 'negative'),
('Such a shame when a skinny body is wasted on an ugly face', 'negative'),
('we all actually hate you', 'negative'),
('You have your butt cheeks out in OHill', 'negative'),
('Who in their right mind would go near those teeth', 'negative'),
('REALLY bad headache this morning', 'negative'),
('Join our dog days of summer sale', 'negative'),
('we are finally no longer at the top of the list of social media screw-ups', 'negative'),
('I\'ve heard of getting screwed by an airline but this is ridiculous', 'negative'),
('Just be thankful it wasn\'t a model', 'negative'),
('we\'re sorry to hear that you did not have a good trip, Mr. Armstrong', 'negative'),
('I always get stuck sitting near a crying baby', 'negative'),
('I\'m so sorry I\'m scared now', 'negative'),
('I\'m just a fangirl pls I don\'t have evil thoughts and plus I\'m a white girl', 'negative'),
('stocks are falling', 'negative'),
('He is my enemy', 'negative')
```

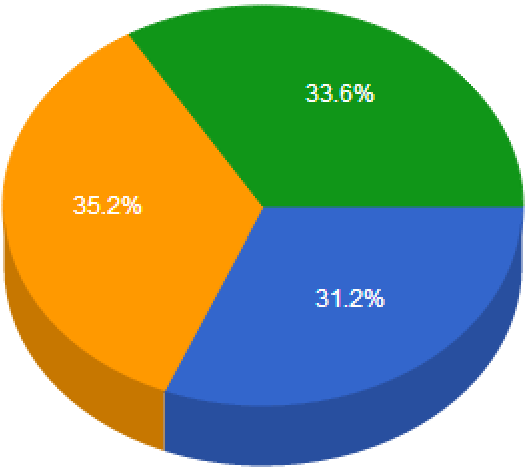
6.2 Output Graph Design

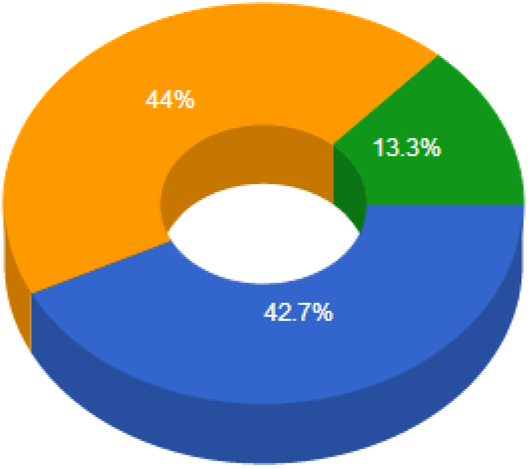
Output uses JavaScript library for manipulating documents based on data. JavaScript and D3 library helps us bring data to life using HTML, SVG and CSS. D3's emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation.

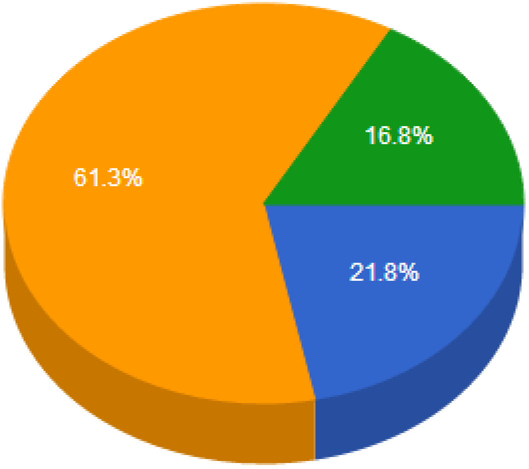
The graphs are easy to understand for the users and they are quicker than the text output to gather the knowledge from the information.

Graphs used: 3D Donut graph









Chapter 7

Applications

The most common application of sentiment analysis is in the area of reviews of consumer products and services. Another important domain for sentiment analysis is the financial markets. There are numerous tweets about each public company. A sentiment analysis system can use these various sources to find articles that discuss the companies and aggregate the sentiment about them as a single score that can be used by an automated trading system.

1) Applications in Business Intelligence

It is very difficult to survey customers who didn't buy the company's laptop. Instead, we could use our application to search the web for opinions and reviews of this and competing products.

2) Cross domain applications

Insights and applications from SA have been useful in other areas

– Politics/political science

– Law/policy making

– Sociology

– Psychology

In general, Humans are subjective creatures and opinions are important. Being able to interact with people on that level has many advantages for information systems.

Chapter 8

Conclusion

Initially, we have analyzed the major approaches to analysis of Twitter data. We referred some technical papers and publications for the same and also a lot of general information on the internet. We gathered various data on the following:

- Data sets (Twitter, Facebook, Blogger)
- Various sentiment analysis methods
- Various Technologies (Python, JSON, JavaScript, HTML, CSS)

Based on our research and observations, we have proposed the technologies and methods that we will be using for designing the tool which will analyze the data. We finalized Twitter stream data because it has the information in the form of Tweets with maximum 140 characters. It is easy to gather the knowledge and perform sentiment analysis on this data.

We plan to categories the data from Twitter which consists of tweets from users based on the topic of discussion. The user will use the tool to search the topics of his choice and the tool will show him the extent of discussion, popularity, reaction and related topics to it. The tweets are categorized into positive and negative using NLTK algorithm for training the data set using the existing set of positive and negative tweets. After this the trained model is compared with the tweets gathered. This will generate the total number of positive and negative tweets. Using the count we will generate the final graph for the user. This graph will undergo transition because the data in streaming real time. The change in the sentiment will automatically displayed in the graph.

APPENDIX 1: Code for Extracting Twitter Stream Data

```
from tweepy import Stream

from tweepy import OAuthHandler

from tweepy.streaming import StreamListener

import time

ckey='C9afrqfHZprKGmg3gNdUg'

csecret='SFHb24Uin1b1bGd3Qc7OwLT44c22jpTTWmdlWsBAI'

atoken='2358376346-n2Z2vDLxyD1nyltUMkQ8qM87IaTdcFWmKnDIiFr'

asecret='GLRGzdxmELUbY7VWueXTPzLdnxxtGqPDX4QwaOfyaRSSF'

input = raw_input('Enter the Keyword : ')
class listener(StreamListener):

     def on_data(self, data):
          try:
               tweet =
data.split(',"text":"')[1].split('","source')[0]
               print tweet
               saveFile = open('twitDB2.csv','a')
               saveFile.write(tweet)
               saveFile.write('\n')
               saveFile.close()
               return True
          except BaseException, e:
               print 'failed ondata,',str(e)


     def on_error(self, status):
          print status


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

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

```
# Open/Create a file to append data
csvFile = open('result.csv', 'a')
#Use csv Writer
csvWriter = csv.writer(csvFile)

csvWriter.writerow([tweet.created_at, tweet.text.encode('utf-
8')])
print (tweet.created_at, tweet.text);
csvFile.close()
```

APPENDIX 2: Code for Training the Classifier

```
import nltk

from nltk.tokenize import word_tokenize

from nltk.probability import FreqDist

pos_tweets = [('I love this car', 'positive'),
              ('This view is amazing', 'positive'),
              ('I feel great this morning', 'positive'),
              ...
              ('my best wishes are always with you',
'positive')]

neg_tweets = [('I do not like this car', 'negative'),
              ('This view is horrible', 'negative'),
              ('I feel tired this morning', 'negative'),
              ...
              ('stocks are falling', 'negative'),
              ('He is my enemy', 'negative')]

tweets = []

for (words, sentiment) in pos_tweets + neg_tweets:
    words_filtered = [e.lower() for e in words.split() if len(e)
>= 3]
    tweets.append((words_filtered, sentiment))

#!/usr/bin/python

print('Content-type: text/html\r\n\r')

def get_words_in_tweets(tweets):
    all_words = []
    for (words, sentiment) in tweets:
     all_words.extend(words)
    return all_words

def get_word_features(wordlist):
    wordlist = nltk.FreqDist(wordlist)
    #wordss = nltk.tokenize.word_tokenize(wordlist)
    #wordlist = nltk.FreqDist(wordss)
    word_features = wordlist.keys()
    return word_features
```

```
word_features  = get_word_features(get_words_in_tweets(tweets))
print (word_features )



def extract_features(document):
    document_words = set(document)
    features = {}
    for word in word_features:
     features['contains(%s)' % word] = (word in document_words)
    return features

training_set = nltk.classify.apply_features(extract_features,
tweets)

classifier = nltk.NaiveBayesClassifier.train(training_set)
```

APPENDIX 3: Code to analyze the Sentiments

```
import nltk

from nltk.tokenize import word_tokenize

from nltk.probability import FreqDist

import time

x=1
old=0
positive=0;
negative=0;
irrelevant=0;

classifier = nltk.NaiveBayesClassifier.train(training_set)

while x==1:
     fo=open('twitDB2.csv', 'rb')

     fo.seek(old)

     for line in fo:
          #print line
          tweet = line+""
          if
classifier.classify(extract_features(tweet.split()))== positive
:
               positive++
          if
classifier.classify(extract_features(tweet.split()))== negative
:
               negative++
          else:
               irrelevant++

          old=fo.tell()

     time.sleep(0.5)

     fo.close()

classifier = nltk.NaiveBayesClassifier.train(training_set)
```

APPENDIX 4: Graph generation Code

Index.html

```
<!DOCTYPE html>

<meta charset="utf-8">

<style>

body {
  font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;
  width: 960px;
  height: 500px;
  position: relative;
}
path.slice{
	stroke-width:2px;
}
polyline{
	opacity: .3;
	stroke: black;
	stroke-width: 2px;
	fill: none;
}
svg text.percent{
	fill:white;
	text-anchor:middle;
	font-size:12px;
}

</style>

<body>

<button onClick="changeData()">Change Data</button>
<script src="http://d3js.org/d3.v3.min.js"></script>
<script src="I:\Donut\donut.js"></script>
<script>

var salesData=[
	{label:"Positive", color:"#3366CC"}, //blue
	{label:"Negative", color:"#FF9900"},  // orange
	{label:"Irrelevant", color:"#109618"}, //green
];
```

```
var svg =
d3.select("body").append("svg").attr("width",700).attr("height",
300);

svg.append("g").attr("id","salesDonut");

svg.append("g").attr("id","quotesDonut");

Donut3D.draw("salesDonut", randomData(), 150, 150, 130, 100, 30,
0.4);
Donut3D.draw("quotesDonut", randomData(), 450, 150, 130, 100,
30, 0);
     /*
function changeData(){
     Donut3D.transition("salesDonut", randomData(), 130, 100,
30, 0.4);
     Donut3D.transition("quotesDonut", randomData(), 130, 100,
30, 0);
}
*/

function randomData(){
     return salesData.map(function(d){
          return {label:d.label, value:1000*Math.random(),
color:d.color};});
}
</script>
</body>
```

Dounut.js

```
!function(){
     var Donut3D={};

     function pieTop(d, rx, ry, ir ){
          if(d.endAngle - d.startAngle == 0 ) return "M 0 0";
          var sx = rx*Math.cos(d.startAngle),
               sy = ry*Math.sin(d.startAngle),
               ex = rx*Math.cos(d.endAngle),
               ey = ry*Math.sin(d.endAngle);

          var ret =[];
          ret.push("M",sx,sy,"A",rx,ry,"0",(d.endAngle-
d.startAngle > Math.PI? 1: 0),"1",ex,ey,"L",ir*ex,ir*ey);
          ret.push("A",ir*rx,ir*ry,"0",(d.endAngle-d.startAngle
> Math.PI? 1: 0), "0",ir*sx,ir*sy,"z");
          return ret.join(" ");
     }

     function pieOuter(d, rx, ry, h ){
          var startAngle = (d.startAngle > Math.PI ? Math.PI :
d.startAngle);
          var endAngle = (d.endAngle > Math.PI ? Math.PI :
d.endAngle);

          var sx = rx*Math.cos(startAngle),
               sy = ry*Math.sin(startAngle),
               ex = rx*Math.cos(endAngle),
               ey = ry*Math.sin(endAngle);

               var ret =[];
               ret.push("M",sx,h+sy,"A",rx,ry,"0 0
1",ex,h+ey,"L",ex,ey,"A",rx,ry,"0 0 0",sx,sy,"z");
               return ret.join(" ");
     }

     function pieInner(d, rx, ry, h, ir ){
          var startAngle = (d.startAngle < Math.PI ? Math.PI :
d.startAngle);
          var endAngle = (d.endAngle < Math.PI ? Math.PI :
d.endAngle);

          var sx = ir*rx*Math.cos(startAngle),
               sy = ir*ry*Math.sin(startAngle),
               ex = ir*rx*Math.cos(endAngle),
               ey = ir*ry*Math.sin(endAngle);
               var ret =[];
               ret.push("M",sx, sy,"A",ir*rx,ir*ry,"0 0 
1",ex,ey, "L",ex,h+ey,"A",ir*rx, ir*ry,"0 0 0",sx,h+sy,"z");
               return ret.join(" ");
     }

     function getPercent(d){
          return (d.endAngle-d.startAngle > 0.2 ? 
                    Math.round(1000*(d.endAngle-
d.startAngle)/(Math.PI*2))/10+'%' : '');
     }

     Donut3D.transition = function(id, data, rx, ry, h, ir){
          function arcTweenInner(a) {
            var i = d3.interpolate(this._current, a);
            this._current = i(0);
            return function(t) { return pieInner(i(t), rx+0.5, 
ry+0.5, h, ir);  };
          }
          function arcTweenTop(a) {
            var i = d3.interpolate(this._current, a);
            this._current = i(0);
            return function(t) { return pieTop(i(t), rx, ry, 
ir);  };
          }
          function arcTweenOuter(a) {
            var i = d3.interpolate(this._current, a);
            this._current = i(0);
            return function(t) { return pieOuter(i(t), rx-.5, 
ry-.5, h);  };
          }
          function textTweenX(a) {
            var i = d3.interpolate(this._current, a);
            this._current = i(0);
            return function(t) { return 
0.6*rx*Math.cos(0.5*(i(t).startAngle+i(t).endAngle));  };
          }
          function textTweenY(a) {
            var i = d3.interpolate(this._current, a);
            this._current = i(0);
            return function(t) { return 
0.6*rx*Math.sin(0.5*(i(t).startAngle+i(t).endAngle));  };
          }
          var _data =
d3.layout.pie().sort(null).value(function(d) {return
d.value;})(data);

          d3.select("#"+id).selectAll(".innerSlice").data(_data)
               .transition().duration(750).attrTween("d",
arcTweenInner);

          d3.select("#"+id).selectAll(".topSlice").data(_data)
               .transition().duration(750).attrTween("d",
arcTweenTop);

          d3.select("#"+id).selectAll(".outerSlice").data(_data)
               .transition().duration(750).attrTween("d",
arcTweenOuter);


     d3.select("#"+id).selectAll(".percent").data(_data).transit
ion().duration(750)

     .attrTween("x",textTweenX).attrTween("y",textTweenY).text(g
etPercent);
     }

     Donut3D.draw=function(id, data, x /*center x*/, y/*center
y*/,
               rx/*radius x*/, ry/*radius y*/, h/*height*/,
ir/*inner radius*/){

          var _data =
d3.layout.pie().sort(null).value(function(d) {return
d.value;})(data);

          var slices =
d3.select("#"+id).append("g").attr("transform", "translate(" + x
+ "," + y + ")")
               .attr("class", "slices");


     slices.selectAll(".innerSlice").data(_data).enter().append(
"path").attr("class", "innerSlice")
               .style("fill", function(d) { return
d3.hsl(d.data.color).darker(0.7); })
               .attr("d",function(d){ return pieInner(d,
rx+0.5,ry+0.5, h, ir);})
               .each(function(d){this._current=d;});
     slices.selectAll(".topSlice").data(_data).enter().append("p
ath").attr("class", "topSlice")
               .style("fill", function(d) { return d.data.color;
})
               .style("stroke", function(d) { return
d.data.color; })
               .attr("d",function(d){ return pieTop(d, rx, ry,
ir);})
               .each(function(d){this._current=d;});


     slices.selectAll(".outerSlice").data(_data).enter().append(
"path").attr("class", "outerSlice")
               .style("fill", function(d) { return
d3.hsl(d.data.color).darker(0.7); })
               .attr("d",function(d){ return pieOuter(d, rx-
.5,ry-.5, h);})
               .each(function(d){this._current=d;});


     slices.selectAll(".percent").data(_data).enter().append("te
xt").attr("class", "percent")
               .attr("x",function(d){ return
0.6*rx*Math.cos(0.5*(d.startAngle+d.endAngle));})
               .attr("y",function(d){ return
0.6*ry*Math.sin(0.5*(d.startAngle+d.endAngle));})

     .text(getPercent).each(function(d){this._current=d;});

     }

     this.Donut3D = Donut3D;
}();
```

REFERENCES

[1] Leah Betancourt (2010 , March 02) [online] available: http://mashable.com/2010/03/02/data-mining-social-media/

[2]Margaret Rouse (2008, December) [online] available: http://searchsqlserver.techtarget.com/definition/data-mining

[3] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, Rebecca Passonneau, ‘Sentiment Analysis of Twitter Data’ [Technical Paper] available: Proceedings of the Workshop on Language in Social Media (LSM 2011), pages 30–38, Portland, Oregon, 23 June 2011. c 2011 Association for Computational Linguistics

[4] Fayyad, Usama; Piatetsky-Shapiro, Gregory; Smyth, Padhraic (1996). "From Data Mining to Knowledge Discovery in Databases". Retrieved 17 December 2008.

[5] "Data Mining Curriculum". ACM SIGKDD. 2006-04-30. Retrieved 2011-10-28.

[6] “Graph generation” http://bl.ocks.org/NPashaP/9994181

[7] Sentiment Analysis of Twitter Data, Department of Computer Science, Columbia University, 2010

[8] End-to-End Sentiment Analysis of Twitter Data, Columbia University, 2008

[9] Wang, W., Chen, L., Thirunarayan, K., Sheth, A., ” Harnessing 4 Twitter Big Data for Automatic Emotion Identification,” In the proceedings of International Conference on Social Computing , 2012.

[10] Lisa Pearl; Mark Steyvers, “C’mon – You Should Read This”: Automatic Identification of Tone from Language Text, International Journal of Computational Linguistics (IJCL), 2012

[11] Alec Go, Richa Bhayani, Lei Huang, “Twitter Sentiment Classification using Distant Supervision,” Stanford Digital Library Technologies Project, 2009

[12] “Joshthecoder” https://github.com/tweepy/tweepy

ACKNOWLEDGEMENT

We would like to thank our internal guide Dr. Dhananjay R. Kalbande for his overwhelming support during the entire duration of our project and for being a great mentor. He has helped refine our project idea and give direction to our project at every step.

It is only right to express our sincere gratitude to the BE Computer batch of 2013-14, our classmates, whose help and support was instrumental in the development and successful completion of this phase of our project. Also, the teaching staff and non-teaching staff of the Computer Department who have helped us from time to time with necessary resources and otherwise, deserve a special mention.

Ultimately, we would like to thank the open-source community, world over, who have taken painstaking efforts and provided us with useful information available on the Internet.