**Chapter 3**

**High Level Design**

The software development usually follows Software Development Life Cycle (SDLC). The design stage involves two sub stages namely High level design and detailed level design. High level design gives an overview of how the system works and top level components comprising the system.

**3.1 Design Considerations**

While designing the program, certain design considerations were taken into account. These are as follows:

**3.1.1 General Constraints**

The general constraint on the system as of now is Twitter search API’S cant able to give accurate tweets. Our training data is not confined to a specific domain. So it affects the accuracy of the results. The text in the query box should be limited to English.

**3.1.2 Development Methods**

**3.1.2.1 Retrieval:**

The company Twitter offers different methods to access their data. Two types of application programming interfaces (API's) are offered to communicate with Twitter. A REST API, which provides a simple interface to the Twitter functionalities, and a Streaming API which is a powerful real-time API. The access to the public data of Twitter is extremely limited with the REST API, and less so for the Twitter Streaming API. The Twitter Streaming API is designed for the more inventive uses of the Twitter API. Neither of the two provides full access to the public posted tweets. For full access to the Twitter data, commercial packages can from a third-party can be used. Twitter4J is an open source Java package which provides full access to the real time tweets. The non-full access of the Twitter Streaming API still provides a huge amount of sampled data for scientific purposes. Twitter4J is an unofficial Java library for the Twitter API, with Twitter4J, you can easily integrate your Java application with the Twitter service. Twitter4J is an unofficial library.

Twitter4J features:

* 100% Pure Java - works on any Java Platform version 5 or later
* Zero dependency : No additional jars required
* Built-in OAuth support
* Out-of-the-box gzip support
* 100% compatible with Twitter API 1.1

**3.1.2.1 Preprocessing:**

The Twitter language model has many unique properties.

1. Usernames Users often include Twitter usernames in their tweets in order to direct their messages. A de facto standard is to include the @ symbol before the username (e.g. @alecmgo). An equivalence class token (USERNAME) replaces all words that start with the @ symbol.
2. Usage of links Users very often includes links in their tweets. An equivalence class is used for all URLs. That is, we convert a URL like “http://tinyurl.com/cvvg9a" to the token \URL.”
3. Tweets contain very casual language. Example, if you search \hungry" with an arbitrary number of u's in the middle (e.g. huuuungry, huuuuuuungry, huuuuuuuuuungry) on Twitter, there will most likely be an on empty result set. We use preprocessing so that any letter occurring more than two times in a row is replaced with two occurrences.

**3.1.2.2 Analysis:**

**Machine Learning Methods**

There are many machine learning algorithms available, some of them are Naive Bayes, SVM and Maximum Entropy.

**Naïve Bayes Method**:

Naive Bayes is a simple model which works well on text categorization. We use a multinomial Naive Bayes model. Here tweet d is considered as a list of words x1 to xn.  Class cj is assigned to tweet d if the computed probability result using the below formula is highest for class cj.



In this formula, xi represents a particular feature of tweet d. CNB is the class to which the tweet d is assigned. Parameters P(cj) and P (xi |cj) are obtained through maximum likelihood estimates, and Laplace add-1 smoothing is used for unseen features.

P(cj) (prior) is computed using the below formula



where p(cj) is the probability of the occurrence of document cj in a give list of documents, Ndoc is the count of total number of documents.

P (xi |cj) (Likelihood) is calculated using the below formula



where  is the total number of occurrences of the feature xi in document c. V is vocabulary(that is the total count of the distinct set of words in training set).

**3.2 Architectural Strategies**

This section explains the approach that was used and also the procedure for future

enhancements of this project.

**3.2.1 Programming Language**

The programming language that was used to develop the entire application was Java.

The back end consisted of preprocessing of twitter data and Naïve Bayes implementation that was purely developed in java. Front end was developed using JSP, HTML and JavaScript.

**3.2.2 Future Plans**

The future plans for this project are to increase the accuracy of the results by probably applying new machine learning techniques and using good training data. Develop a mobile app, so that user can access this application easily.

**3.2.3 User Interface Paradigm**

The User Interface is completely designed using JSP, HTML and Java script. The user will need to enter the keyword in the textbox and press the submit button in order for the twitter data to be retrieved and analyzed. The analyzed tweets will be displayed in the three columns i.e., positive, negative and neutral based on the class assigned for that particular tweet.

**3.2.4 Data Storage Management**

The Data Management in this project was handled in the best way possible. We are storing the data retrieved from twitter in a text file, we analyze this stored data line by line by using training data which is also stored in three other text files. The results are stored in another text file. Web page accesses this result file to display the results.

**3.3 System Architecture**

System architecture of the project is shown in figure 3.1. There are following steps

* Twitter: In this step the data is accessed using Twitter APIs.
* Reference database: Twitter database with tweets is reference database for this project.
* Retrieval: This can be done by giving keyword as a input here accessing tweets takes place by reference database.
* Preprocessing: In this step ambiguities are removed from Twitter data.
* Analysis: This is done by Naïve Bayes machine learning algorithm. It produces classified data as positive, negative and neutral.
* Web Application: Here dynamic web application is hosted in a server to demonstrate the application.

**Reference**

**Database**

**Machine Learning Techniques**

**Preprocessing**

Tweets

**Twitter**

**Retrieval**

**Analysis**

**Web Application user interface**

Figure 3.1 System Architecture

**3.4 Data Flow Diagrams**

**3.4.1 LEVEL 0:**

The level 0 data flow diagram consists of only one process called the Sentiment Analysis and Classification (1). The input to this module is the tweets and the training data. The output is the classified data that is the tweets classified into 3 groups positive, negative or neutral. Figure 3.1 explains the DFD level 0.

Tweets Result

**Classified data**

**Get the data from Twitter**

Training Data

Figure 3.2 DFD level 0

**3.4.2 Level 1:**

In the level 1 data flow diagram as explained in figure 3.2, Sentiment Analysis And Classification is shown in modules namely retrieval of tweets module (1.1), preprocessing module (1.2), and Sentiment Analysis module(1.3). The keyword is given as input to the retrieval of tweets module, it retrieves from the twitter database based on given input keyword (1.1). The output of this module is given to preprocessing module (1.2); here the unwanted data in tweets is removed. This module output is given to sentiment analysis module (1.3), it uses the naïve bayes implementation technique to train the data which is retrieved from twitter. The output is the classified data.

**User entering a keyword**

Twitter data

Keyword

Tweets

Training data

Machine leaning

Text file with Tweets

**Preprocessing**

**(1.2)**

Processed file

Analyzed Tweets

**Classified data**

Figure 3.3 DFD level 1

**3.4.3 Level 2:**

In the level 2 data flow diagram, Preprocessing module shown in level 1 is further divided into two components namely Processing (1.2.1), Feature reduction (1.2.2) . The twitter data is given as the input to the pre-processor module. The twitter data free form unwanted URL’s obtained after pre-processing (1.2) are given as the input to the feature reduction module (1.2.2). This module generates normalized extracted twitter data and feeds them as input to the sentiment analysis module (1.3).The sentiment analysis module is further divided into three sub modules. Machine Learning Algorithm (1.3.2), Training Model (1.3.1), and Evaluation Model (1.3.3) the final output is the classified data. Figure 3.3 explains the DFD level 2.

Twitter database

Tweets

**User enter the keyword**

Keyword

Retrieved Tweets

Processed Data

Training Data

Feature reduced Data

**Machine Learning**

**Algorithm**

**(1.3.1)**

**Evaluation Model**

**(1.3.3)**

**Training Model**

**(1.3.2)**

Analyzed data

**Classified data**

Figure 3.4 DFD level 2