**EmoTweets**

A Project Presented to

The Faculty of the School of IT and Computing

De La Salle University – Science & Technology Complex

In Partial Fulfillment

of the Requirements for the Degree of

Bachelor of Science in Computer Science

by

Blanquera, Kimberly D.

Dayupay, Mike I.

Professor Ma. Christine Gendrano

July <#>, 2015

1. **Company Background**
   1. **Brief History**

131dB started informally as three developers working on contract projects from previous clients and referrals as early as 2008. The group worked on telephony systems and integration of services to such systems.In 2010, projects became regular at a rate that the group needed to incorporate. The group continued working as a startup, meeting regularly in coffee shops while renting a virtual office in Makati City.After maintaining steady growth in 2012, 131dB opened their headquarters in Alabang, Muntinlupa City. Over time, 131dB gained a portfolio of clients and customers in various industries such as BPO, finance and telecommunications.

* 1. **Products/Services** 
     1. VoIP Services

131dB aims to give clients excellent services and reliable connections at a highly affordable cost. They offer a wide range of services of VoIP and IP-BPX development and consultation. They cater to companies in the United States and in 60 countries around the world.

**Services 131dB provides:**

* + - * VoIP Trunking – make outbound calls from around the globe at minimum standard cost.
      * Direct Inverse Dialing (DDD) – receive inbound calls from the US and other international countries, including toll free.
      * IP-BPX Development and Consultation services – 131dB helps you cut business by shifting to IP- sited networks without changing their existing setup using our unified communication platform. Increase productivity, monitor performance and tune your workforce with their customizable tools.
    1. E-Pass Outlet
    2. Software solutions to businesses
  1. **Market**
  2. **Organizational Structure (including IT Department)**

1. **Project Background**
   1. **Current Situation**

Twitter has drawn millions of users who use their services, eventually creating a large pool of data. Users, 134 million and growing, send and post messages, called *tweets*, at a rate of 1382% in the year 2008 alone. Twitter data has been used to gather news, customer views about various brands or products, and public opinion regarding politicians. This shows the tangibility of applying natural language processing to Twitter data to gain insight on human emotions. Increased usage and future technology can render services like Twitter as a source of better understanding of the human condition: opinions, feelings and lives.

* 1. **Project Description**

EmoTweets performs sentiment analysis on Tweets and classifies a particular Tweet as either positive or negative. The program fetches Tweets with the hashtag that a user inputs. The Tweets are then tokenized and processed against an existing data set, or *vocabulary*, using the Naïve Bayes algorithm. Based on calculated probabilities, Tweets are then given a corresponding class.

* 1. **Significance**
     1. **Academe**

The project can be used as a baseline for future works and research in the field of Natural Language Processing especially in sentimental analysis and text classification. Future works and research could use the resources like tools, libraries, APIs, and data sets that the project used.

* + 1. **Student**

Students can use the project as reference to their projects in Artificial Intelligence. Students could also use the tools, libraries, APIs, and data sets that the project used.

* + 1. **Society**

Text and sentiment analysis, specifically on data from social media, benefits all aspects of society. It can aid the government gauge what citizens are happy or unhappy about with regard to their programs and way of governance. These concepts may also be applied to searching Twitter, and other social media services online, for alarming news to alert others, crisis preparation and prevention as well as shed light on issues that have roused strong emotions from people.

* 1. **Scope and limitation**

The application can only process a maximum amount of 1,500 Tweets from the past 7 days at most because of the limitation posed by the Twitter Search API. Only Tweets in the English language can be classified by the application.

In classification, emoticons are not used as attributes. Internet jargon and lingo, however, are included in the vocabulary and affect the outcome.

The analysis relies on the lexical qualities of the text as opposed to a contextual understanding of the Tweets as a whole. Because of this, the application may have some difficulties classifying Tweets that contain numerous instances of negation.

* 1. **Architecture/framework**

The web application has two major modules which are the tweet retrieval module and the text classification module. The tweet retrieval module is responsible for retrieving tweets from twitter using the Twitter API. It will retrieve tweets based on the search keyword that the user will input. On the other hand, the text classification module is responsible for labeling each tweet positive or negative. It will use the Naive- Bayes Algorithm in classifying the sentiment of each tweet. All data needed for the Naive- Bayes Algorithm are stored in a database.

* 1. **Student’s role**

The team consists of two members, Mike Dayupay and Kim Blanquera. The project workload is divided equally to the members of the team. Kim Blanquera is responsible for the tweet retrieval module while Mike Dayupay is responsible for the text classification module. Both members worked on producing the training set and test set for the text classification module.

1. **Methodology** 
   1. **Phases/Activities**
      1. **Planning**

First, the specifications and requirements of the project were discussed. The expected requirements upon completion were defined. Based on this, a list of tasks to be accomplished was produced. Tasks were prioritized and assigned to each group member. The estimated duration of each task or milestone was also decided.

* + 1. **Research**

The first two weeks were spent conducting research on the concept of Natural Language Processing and sentiment analysis as well as tools and libraries that may be used in implementation.

In researching Natural Language Processing and sentiment analysis, course material available online were used. Online journals and papers were also sought after to further aid with implementation using machine learning algorithms. Existing libraries and API were also investigated to gain knowledge of methods that can be used during development. The Twitter Search API was especially focused on as it is the main source of the data to be processed.

* + 1. **Implementation**

EmoTweets is a web application and was developed by using HTML, CSS, Jquery, Ajax, PHP, and MySQL.

The team used HTML and CSS to develop a simple user interface for the web application. Jquery and AJAX was used to make the search and display of tweets to be dynamic wherein there is no need to reload the page.

The main modules of the web application were developed using PHP and MySQL. PHP was used to retrieve tweets from Twitter with the help of Twitter API. PHP and MySQL was used to implement the Naive Bayes Algorithm.

* + 1. **Testing**

2000 annotated tweets were used as input for the initial testing of the sentimental analysis model. There were 1000 positive tweets and 1000 negative tweets. A PHP script was created to classify the test set and tallied the count of True Positive and True Negative.

* 1. **Inputs**

The only input of the application is the hashtag by which it searches Twitter for Tweets.

* 1. **Tools used**

The Twitter API was used to search Twitter for Tweets by hashtag and to retrieve them. TwitterOUath, a PHP library for Twitter license authentication, connected to the API. Tweets were saved on a database stored locally using XAMPP. To perform Natural Language Processing tasks through PHP, the Nlp-tools API was utilized. Nlp-tools facilitated the use of tokenizers in PHP implementation. The pattern-en API, written in Python and integrated with PHP, allowed the checking of noun singularity or plurality which is a function needed in both the processing of the data set as well as for new instances.

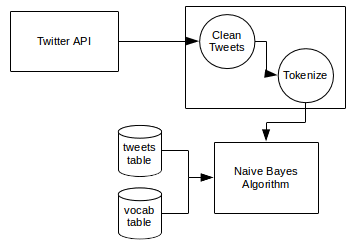
* 1. **Outputs**

The application outputs the resulting Tweets from the search, each labeled with its respective class. A chart is also displayed with statistical information on the results, including the amount of Tweets per class.

* 1. **Problems encountered and solutions undertaken**

Tweets can contain user mentions, hyperlinks, and pronouns which do not represent a sentiment. The problem with this is that these words made a huge difference when computing for the value of the sentiment using Naive Bayes algorithm. The solution was to remove the user mentions, hyperlinks, and pronouns from the training set before tokenizing. This will ensure that these words will not be part of the computation for the value of the sentiment.

1. **Technical Documentation (includes diagrams – where applicable, such as ERDs, DFDs, architecture, business model, manuals, etc.)**
   1. **Architecture**
      1. **Sentiment Classification of Tweets**



The Twitter API will retrieve tweets based on the searched keyword or hashtag input by the user. These tweets will be cleaned by removing user mentions, hyperlinks, and emoticons. After cleaning a tweet, it will be tokenized and stored in an array that will serve as an input to the computation of the sentiment using the Naive Bayes Algorithm.

The Naive Bayes algorithm computes the probability of a document/text belonging to a particular class. The formula for Naive Bayes is as follows:

P(sj) → probability of the sentiment j

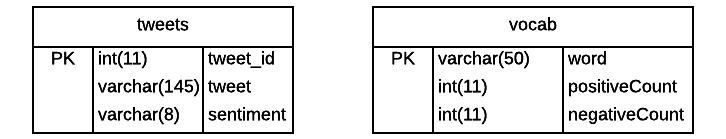
P(wi |sj) → probability of a word in the tweet given a sentiment j

In order to compute the probability of sentiment j, the program will query the database for tweets annotated as sentiment j and count how many it is then it will be divided to the total number of tweets in the database.

In order to compute the probability of a word given sentiment j, the program will query the database for the total number of times the word wj appeared in tweets annotated as sentiment j. It will then be divided to the total number of words annotated as sentiment j plus the total words in the vocab table.

After computing for the probability of the tweet in positive or negative, the two values will be compared and the sentiment with the highest value will be the sentiment annotated to the tweet.

* 1. **Database Design**



Emotweets' database contains two tables which are *tweets* and *vocab*. The *tweets* table has three columns; tweet\_id, tweet, and sentiment. The *tweets* table contains training set of tweets. These tweets are cleaned so user mentions, hyperlinks, and emoticons are removed.

On the other hand, *vocab* table has also three columns; word, positiveCount, and negativeCount. The *vocab* table contains the words in the training set tweets and the number of times it belong to a positive annotated tweet or a negative annotated tweet. This table is the major component of the training model because all the queries in this table will be used in the formula of the Naive Bayes algorithm.

* 1. **Software Requirements**

The following are required to successfully run the application: Python 2.5 or 2.6, pattern-en, XAMPP, web browser and internet connection.

* 1. **Manual**
     1. **Setup and Installation**

Place the main folder, *emotweets*, in the following location

/XAMPP/htdocs

Open XAMPP and start *Apache* and *SQL.* When both of their statuses are *Running*, open a browser. In the URL bar, type *localhost*

* + 1. **Usage**

Open a new tab or new browser window and enter the following in the URL bar:

localhost/emotweets/index.php

Enter a *hashtag* to search for and press enter or click the *Go* button. The tweets retrieved for the searched hashtag will appear on the right with their corresponding classification while a chart showing some statistics of the classification results will be displayed on the left.