SentViz 1.0

Twitter Emotion Visualization

The Final Report

Abstract

Twitter is a rapidly growing social media platform. Every day the size of growing Twitter audience is increasing. Twitter has 300 million current active users and each day 2 million tweets created and post to the twitter by the audience. Each user can compose a length limit micro tweet which is a maximum of 140 characters. Because of this character limitation tweet contents are written in very clear on the user's opinion. Also, tweets are very easy to analyze than any other social media. These tweets contain opinions of millions of people around the world. Especially analyzing tweets is more helpful in business marketing. But manually analyzing tweets is not practical and the solution is a twitter analyzing system based on keywords. This report introduces a web-based software application that can achieve those goals a reliable and efficient way. The front end of the application was implemented using JavaScript and backend implementation using python. The system will collect the tweets based on saved keyword and analyze the emotion of tweets, save results and visualized the results. In addition, it will provide easy and user-friendly interfaces for the user to understand how twitter community thinks about the given topics.

Table of Contents

Table of Figures

1. Introduction

1.1 Background of the Application Domain

SentViz 1.0 is a web-based Twitter emotion the analyzer and visualizer. The business marketing is the main domain this application is going to apply. Every day twitter collect 20 million of its user's tweets. Business can use Twitter to understand how the community thinks about their products and service. Business can use Twitter to monitor what people are talking about their brand online and think about your product that would help you to develop a better marketing strategy and provide inputs to improve your product or services. These tweets have a character limitation(140 characters). These tweets create by people all around the world and tweets contained people opinion. Manually analyzing the collected tweets is not practical. So the system fully automates the tweet collection and analyzing the process. Furthermore, the system visualized the analyzed data to users in an understandable way. Hence the most feasible solution would be a software system. The system hasn't separate users. The user can add keywords and the system will collect tweets and generate the visualization output.

1.2 Motivation for the Selected System Development

Twitter is a rich source with full of people opinions. When comparing Twitter with other social media platforms, Twitter community use the platform in a very formal way. So Twitter is very reliable and analyzing the tweet can use in many fields to understand how people think about any particular topic. The primary domain of this application is business marketing. The customers use Twitter to express their opinion on various products and services. Those tweets are like customer feedbacks, and they can use to understand how customers think about products and services. Manually operation of tweet analyze isn't practical because humans can't handle a large amount of data real-time. So there is a need for a system which collects the tweets and analyzes them automatically. Twitter provides similar services named as Twitter Trends, but it visualizes only details of the most popular keywords. Sentviz is a system that user can add any keyword and get the emotional analyze result of the given keyword.

1.3 Importance and Main Purpose of the System

The objective of this system is to automate the twitter emotion analyzing process and visualize the analyze data which users can easily understand. The importance of this system is users no longer want to examine the tweets manually. Also, system visualizes the analyzed data of tweets in various graphs. Also, the user doesn't worry about the tweet collecting process because web job collects tweet continuously and analyze them. The primary domain of this system is business marketing so companies can get an idea of how the customers think about their products and services. The Twitter trends system is not fully supported for analyzing any keyword. This system is supported to analyze any user given keywords. Also, the system can use to identify the change of the emotion with respect to the time.

1.4 Overview of the System and Used Approach and Outcome

Two types of users will be benefitted from SentViz1.0. First, a student can use the system to analyse individual performance, to compare with another friend, and to predict the likely grade for a future module. Second, a lecturer can use the system to enter/update results and modules and analyse the performance of students. Also, a lecturer can provide meaningful feedback for students after analysing their performance. As the world is more trending towards web based applications than desktop applications, SentVizwas implemented as a web-based system which provides user-friendly and easy web interfaces for the users. Existing libraries and frameworks were used when implementing the system, thereby increasing reusability and saving time. The ultimate outcome of the system is managing the grades of the students and illustrating the analysed data in meaningful and informative reports and graphs.

2. Literature Review

Twitter is a micro-blogging social media that has become more popular with the global. Twitter users can create a short message which also known as Tweets, which are limited to 140 characters(now Twitter change limit for 200). Twitter users update their personal opinions on many subjects and discuss topics through tweets. The Twitter platform is used by many users in world wide because it is easily available to large number of people and has no political or economic restrictions. Micro-blogging social media are becoming a place to find strong viewpoints and emotion when the number of users increase. People use twitter to analyze behavioral and emotional changes of community. Stock market prediction systems are now able to forecast the stock market success by using data from Twitter [7]. Researchers use twitter to analyze the popularity of products and services. From these case, studies proved that Twitter is a beneficial resource for predicting services, products or markets. Twitter is more important when researchers want to analyze how people think about the day to day products. Also, Twitter is a worthy platform for emotion analysis due to its large number of worldwide users. Twitter is a large social media platform with 126 million active users.400 million tweets are added every day. This tweets can be collected through the Twitter stream API.

Twitter tweets analyzing and visualization is not a new concept. But most of the twitter data analyze systems perform the sentiment analyze such as positive, negative or neutral. As an example movie review analyzing system can be considered as a sentiment analyzing system. Analyze the tweets and classify them into emotion classes is a new concept and still, in research level. There are existing systems that analyze tweets and show sentiment or emotion results."sentdex.com" and "Tweet Viz" are examples for such systems."Sentdex" is provided sentiment analyze of twitter data on 100 keywords. Also, the system provides a geographical representation of the sentiment analysis."Tweet Viz" is more complicated system than "sentdex". "Tweet Viz" provides four graphs to visualize analyzed data. Those are "Topics", "Tag cloud", "Emotion Visualizer" and "World Map". This system classified tweets into 16 main emotion categories. Under Topics, Tweets about a common topic are grouped into topic clusters. Keywords above a cluster indicate its topic. Under Tag cloud, Common words from the emotional regions Upset, Happy, Relaxed, and Unhappy are shown. Words that are more frequent are larger. Hover the mouse over a word to see how often it occurred.under the World Map, Tweets are drawn on a map of the world at the location where they were posted. Please note most Twitter users do not provide their location, so only a few tweets will be shown on the map. Hover your mouse over a tweet or click on it to see its text.

The central core of this system is the emotion classification algorithm. There are several approaches for classifying the text to emotions. Naive Bayes, Support Vector Machines are such approaches and Neural Networks. Naive Bayes approach has very low accuracy.By using SVM can acquire significant accuracy of emotion classification. This system is implemented using the SVM approach.There are various unsupervised algorithms to clarification with different efficiency. The research conducted using supervised algorithms have some drawbacks and impact on the efficiency of algorithms. In section IV, we proposed a system which will overcome the problem.Identifying and classifying opinions which are present in the source text are done in sentiment analysis. A huge amount of sentiment rich data in the form of tweets are generated by Twitter users every day. To know the opinion of the people , we can use analyzed twitter data.The maximum character limitation of Twitter is 140. The machine learning approach is the best method for analyzing emotional data from the text.By performing emotion analysis in a specific domain, it is possible to identify the effect of domain information in emotion classification. Tweet s are classified into four main emotion categories by using new feature vector.

Another research tried to pre-processed the dataset, after that extracted the adjective from the dataset that have substantial meaning which is called feature vector, then selected the feature vector list and thereafter applied machine learning algorithms such as Naïve-Bayes, Maximum Entropy and SVM along with the Semantic Orientation based Word-Net which extracts synonyms and relation for the content feature. At the end, they measured the performance of classifier in terms of recall, precision and accuracy.

3 . System Models

3.1 System Requirement

This system has not multiple types of users with different privileges.The main goal of the system is to visualize the emotional data of the tweets according to a keyword. The system provides a web interface to the user, and the user can give two inputs. The user can enter a keyword and the number of days to analyze. The system will retrieve all the tweets that contain provided the keyword and posted within the user's selected time period. The system will retrieve the corresponding tweets by using Twitter stream API.The functional requirements of the system are; retrieve the Tweets from the stream API, clean the content of new received tweets, analyze the Tweets, saved the analyzed data in the database, provide feature to add a new keyword, provide feature to delete an existing keyword, user can change the period of visualizing data, generate the graphs from analyzed data and visualize the all generated graphs.

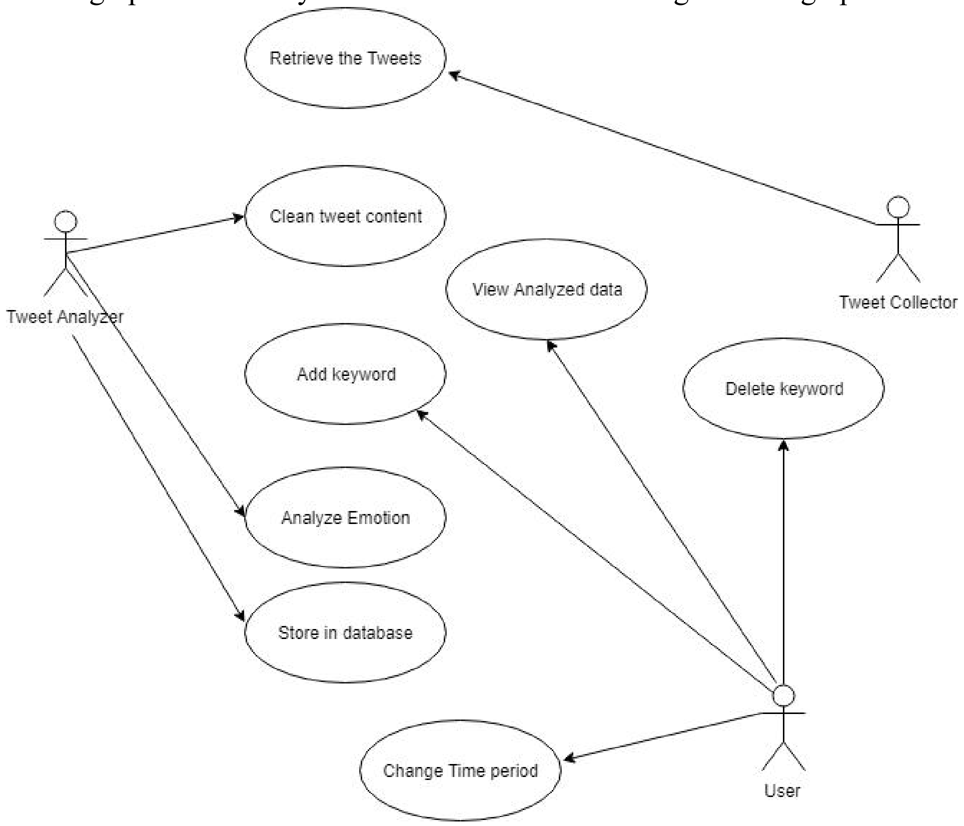


Figure 3.1 – The Main Use Case Diagram

The non-functional requirements can be classified into three categories. Of them, one usability requirement is; it should take a maximum of 1 hour for an average user to get complete training concerning how to handle the system. The SentViz system should provide user-friendly and simple interfaces for users. The system should keep track of all error activities and show a meaningful error message when an error occurs. The reliability requirements are; the system should be available 24/7 with a low MTTR and a high MTBF as well as the Mean Time between Failures (MTBF) of the system should be considerably high. Another essential requirement is, the system should provide higher accuracy when visualizing the heat map of emotions.

The performance requirements are; the response time for analyzing tweets and delivering it for the user should be less than 10 seconds (maximum). When visualizing each graph, the maximum time system should take for each task should be 5-10 seconds.

3.2 System Design

The system follows a three-layer Model,View, and Controller architecture(MVC). All the user interfaces and controllers of the system belongs to the presentation layer. Any data store and retrieve to the database done by the data access layer on request. The functionalities and the service implementation of the system belong to the business logic layer. The Logical view provides an overall representation of the system without going into detailed architectural levels. The logical view is the object model of the design which concerned with the functionality that the system provides to end-users. The logical view of the system is represented by using Class diagram, Communication diagram, and Sequence diagram.UML is used to draw these diagrams. The overview of the system can be shown using a Class diagram as follows.

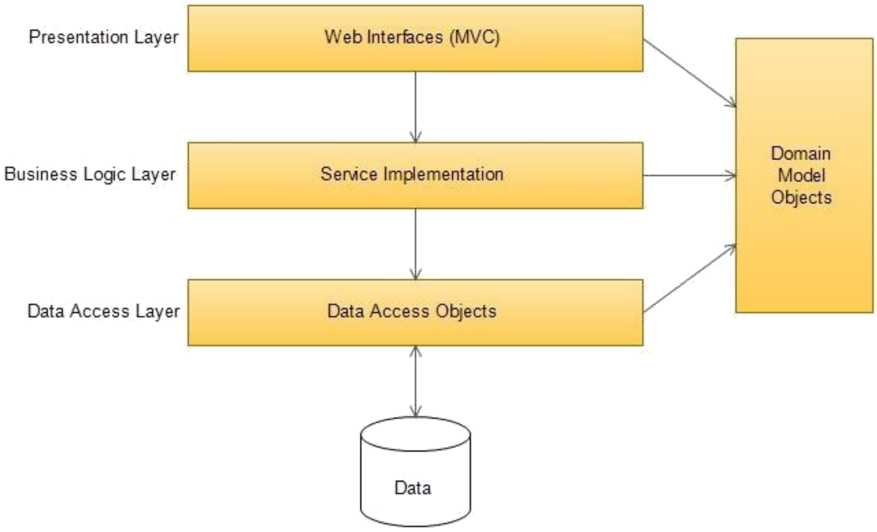


Figure 3.2 - The layered architecture of the application

The logical view of the system is represented by the UML class diagrams which gives the interaction between each design class.

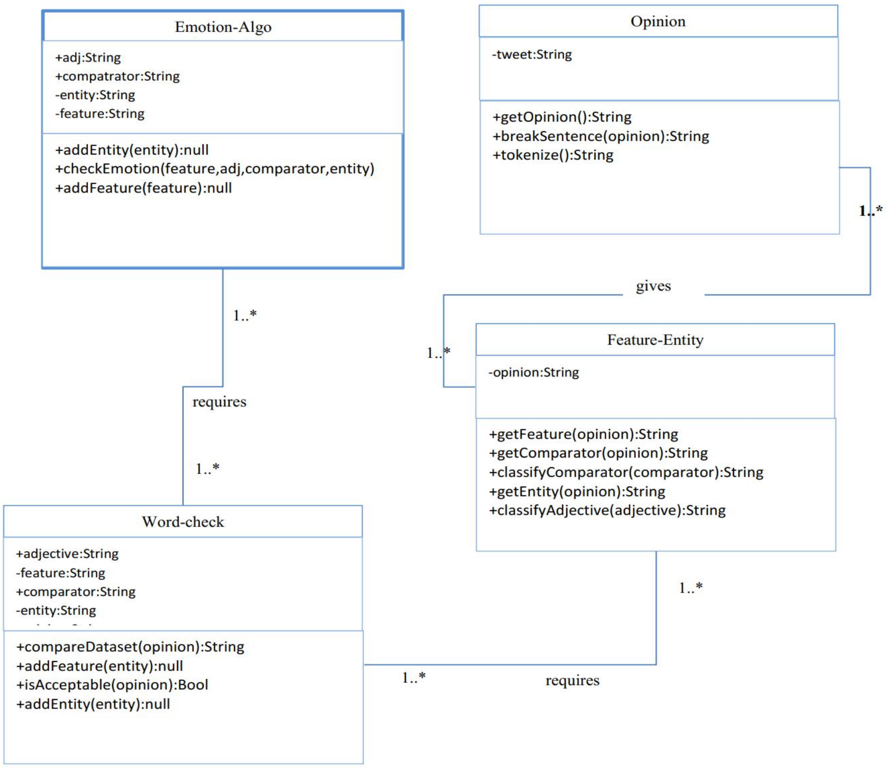


Figure 3.3 - Design class diagram for system

System is a web application so it uses client server architecture. Within the server MVC architecture-based web component is used. System architecture divide into the business logic control and Data access two separate layers. Further Domain model Objects represents the Logical Objects of the system that are used by underlying components.The process view of the system is described using the UML activity diagram. Figure 3.4 shows two main activity diagrams of the system.

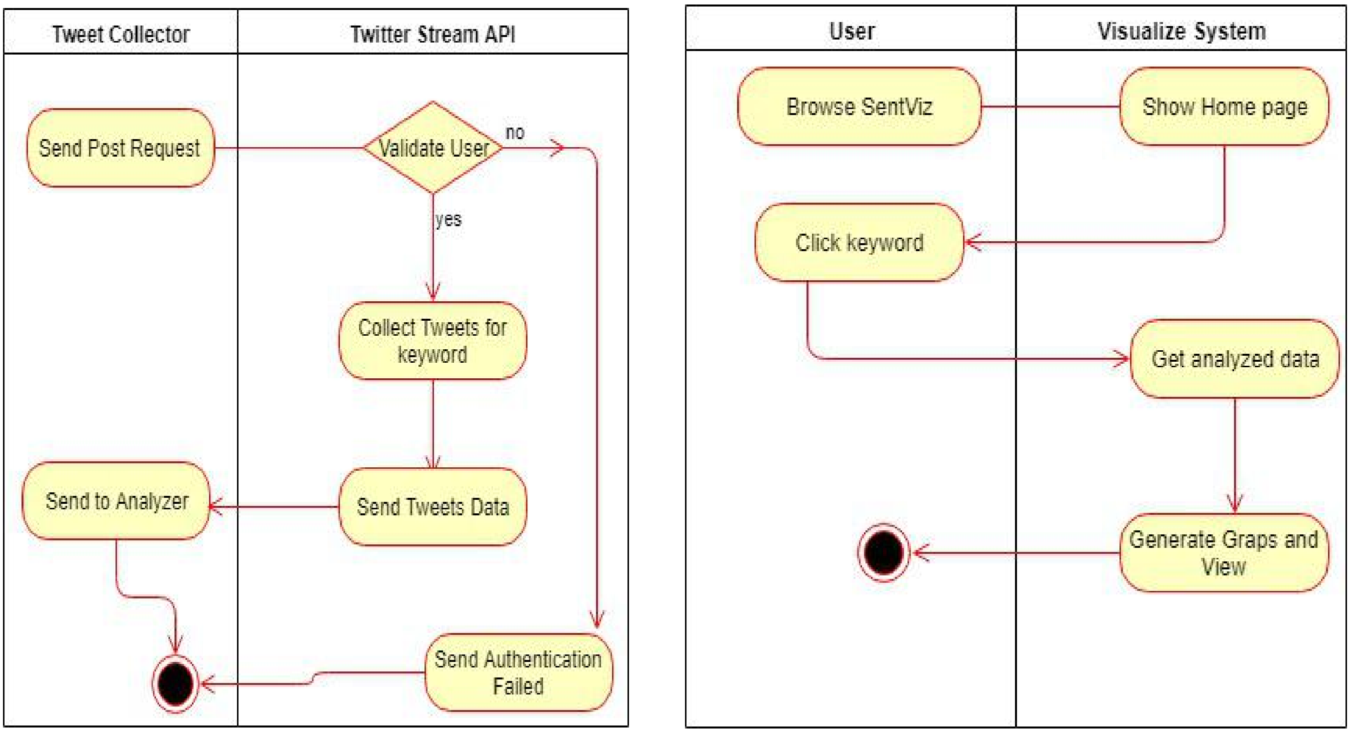


Figure 3.4 - Activity Diagrams for System Collect Tweets(left) and User request results for keyword(right)

To get the analyze result of a keyword, the user must add a keyword to the system. Every 6 minutes system will collect the tweets according to saved keywords then analyzed them and saved in the database. So the user has to wait 6 minutes to get the result of a newly added keyword.

3.3 Database Design

Entity Relationship Diagram shows how the database is designed for the system and the relationship among relations.

Assumptions:

When tweets are collected one tweet can belongs to multiple keywords.

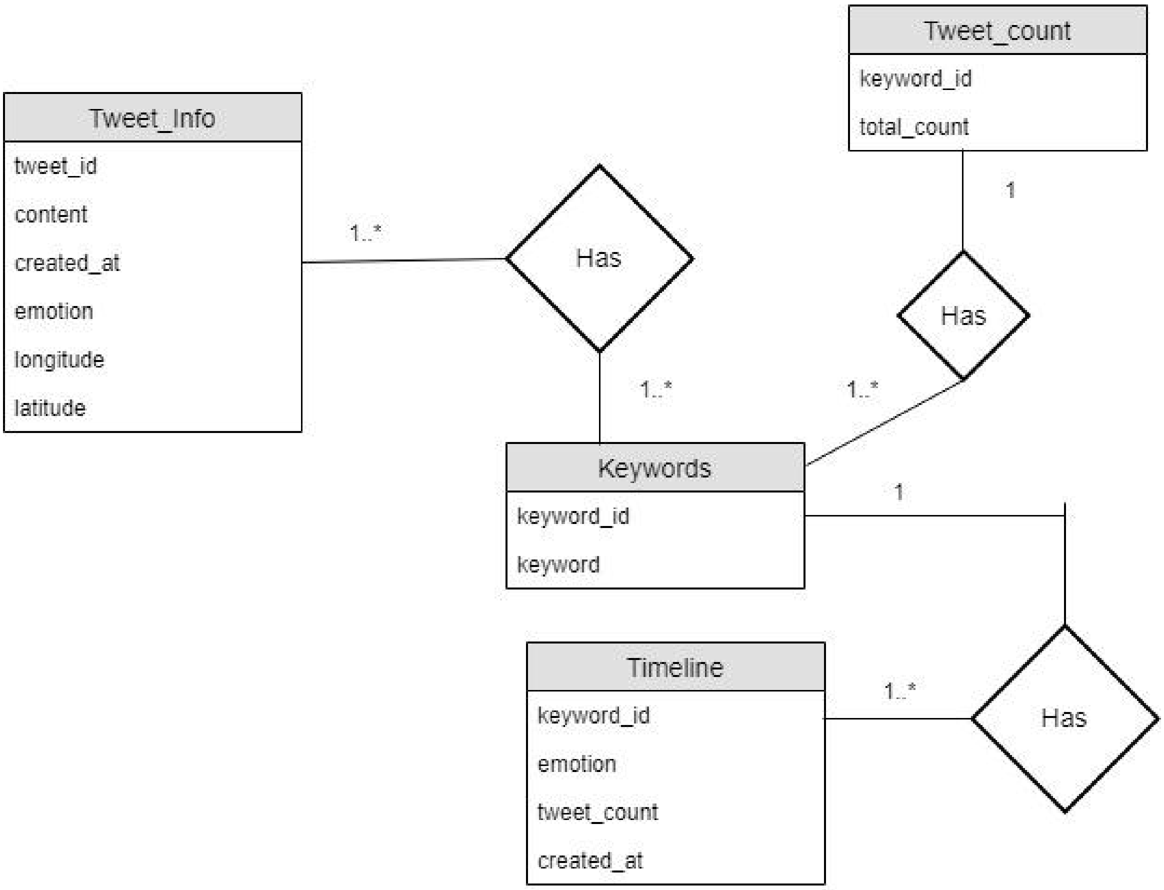


Figure 3.5 shows the ERD of SentViz 1.0. For simplicity, only the main fields were illustrated.

Relational Schema of the SentViz 1.0 system

Tweet_Info(tweet_id,content,created_at,emotion,latitude,longitude)

Timeline(keyword_id,emotion,tweet_count,created_at)

Keywords(keyword_id,keyword)

Tweet_count(keyword_id,total_count)

4 . System Implementation

4.1 Implementation Procedure

This section provides the tools, techniques and applied strategies and methodologies to develop the system.Initially, the first implementation of the system was decided to implement the system as a desktop application. But, the trend at present is more towards web-based applications. Therefore, SentViz1.0 was developed as a web-based application. The technologies used in the front end were javascript, HTML, and CSS. Python was used as backend development. The implementation was done according to MVC architecture. Bootstrap and CSS were used to develop user interfaces. Home page shows the all the existing keywords and number of collected tweets. The only form available in the system is add new keyword form.To validate this form system use bootstrap Form Validation.The system is implemented using Azure cloud services. Tweet collecting and analyzing process is running as background processes. These background processes are implemented as Azure web jobs. Azure SQL database used as the system database.Database was implemented using MySQL. Furthermore, for testing purposes, Pytest, Selenium and JUnitPerf were used.

First, research was done about the application domain and the problem in order to identify the correct scope and requirements of the system. To identify the system scope and requirements , SRS was done.After the SRS , requirements were identified and finalized. Then the project proposal was submitted to be accepted. Then a schedule was prepared covering all the phases of Rational Unified Process after project proposal is accepted. The core of the system and the database is implemented t the design phase. The system uses a layered architecture in which there are three main layers namely; service implementation, web interface and data access layers . According to this architecture.

The backend of the system is implemented using python. To implement the web jobs system is used python and implement the web server system used the python flask framework. The main component of the system is the emotion classification algorithm. Initially, the emotion classification model implements using Naive Bayes classifier. It had to be changed due to the accuracy of the model is between 30%-40%.To achieve the accuracy of more than 90% the emotion classification model is implemented using a Support Vector Machine(SVM).

Originally, the system was designed to use node.js web service and a flask API. It had to be changed due to the long delay of the response of web application. The use cases were implemented one by one after the database and the architecture was finalized. The whole implementation had 4 main iterations for the Main UIs , Emotion timeline generator, Emotion Heat map generator, Tweets analyze and collect web jobs ,and Tweet emotion classifier. Once these subsystems were implemented, the results prediction functionality was implemented.The main component of the system is emotion classification algorithm. This algorithm is designed using a support vector machine. The SVM was implemented and tested. Implementation period was 38 days long and at the end of each iteration, the components were tested to confirm they are working as intended. At the final stage, the whole system was subjected to a carefully designed test plan and the system was refined to improve the quality of the code.

Rational Unified Process (RUP) is the software development methodology used in this system. RUP is an iterative software development process created by the Rational Software Corporation. In RUP at the end of each iteration as each component was implemented iteratively to achieve an exhaustive testing phase. Also at the end of each iteration, there was a working product. The refined of the design due to some reasons all of which were recorded in the previous reports of the system.

4.2 Materials

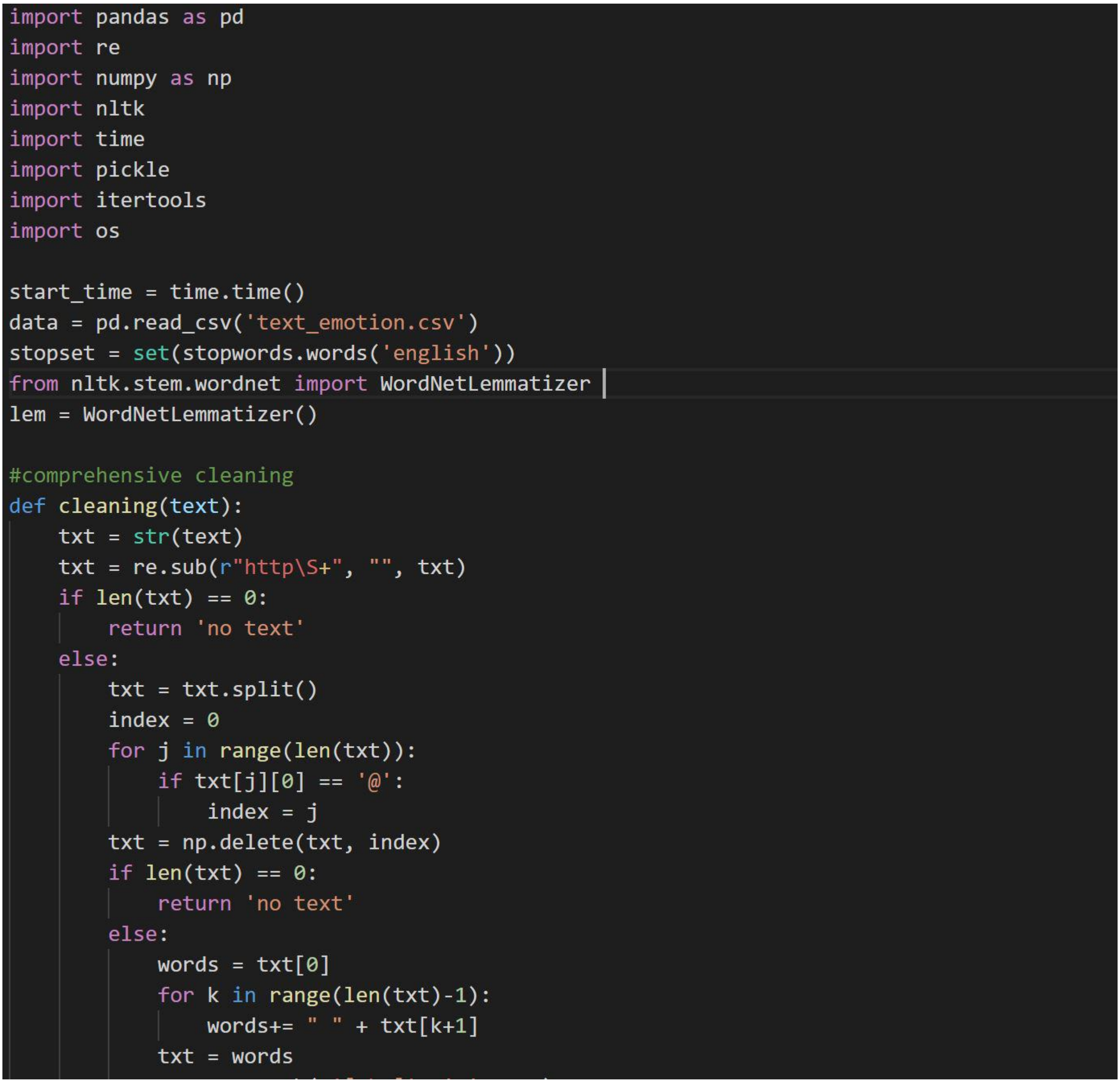
When implementing the system some existing libraries were used to increase reusability. One of the libraries was flask. The flask library is used to implement the web server. Also, Bootstrap CSS library was used to the front end user interface development. It was easy to improve the aesthetic aspects of the user interfaces because of the Bootstrap . The Graph Visualize subsystem heavily uses graphs to plot relationships. These graphs were plotted using an external library called “HighChart.js” and “Anychart.js” . “Highchart.js” is used for the emotion timeline and “Anychart.js” is used for the emotion heat map.When validating the forms, Bootstrap Form Validation [10] was used.

The user can add a particular keyword to the system and system store it in database. The system has three background processes. These background processes implement using Azure web jobs. Twitter collecting, Emotion analyzing and Tweet counting are those web jobs, and they implement using python. Every 6 minutes Tweet collector web job will run and collect tweets according to all keywords in the database. Tweet collecting web job will run every 6 minutes because these system use azure standard version and web job can only have a total run time of 60 minutes per day. When creating the emotion heat map, some keywords may not have enough geo data to generate the heat map. Because of most of the tweets don't have geo coordinates attached. If the keyword is a rare keyword in the Twitter community, then the graphs may have the problem of fewer data to analyze.

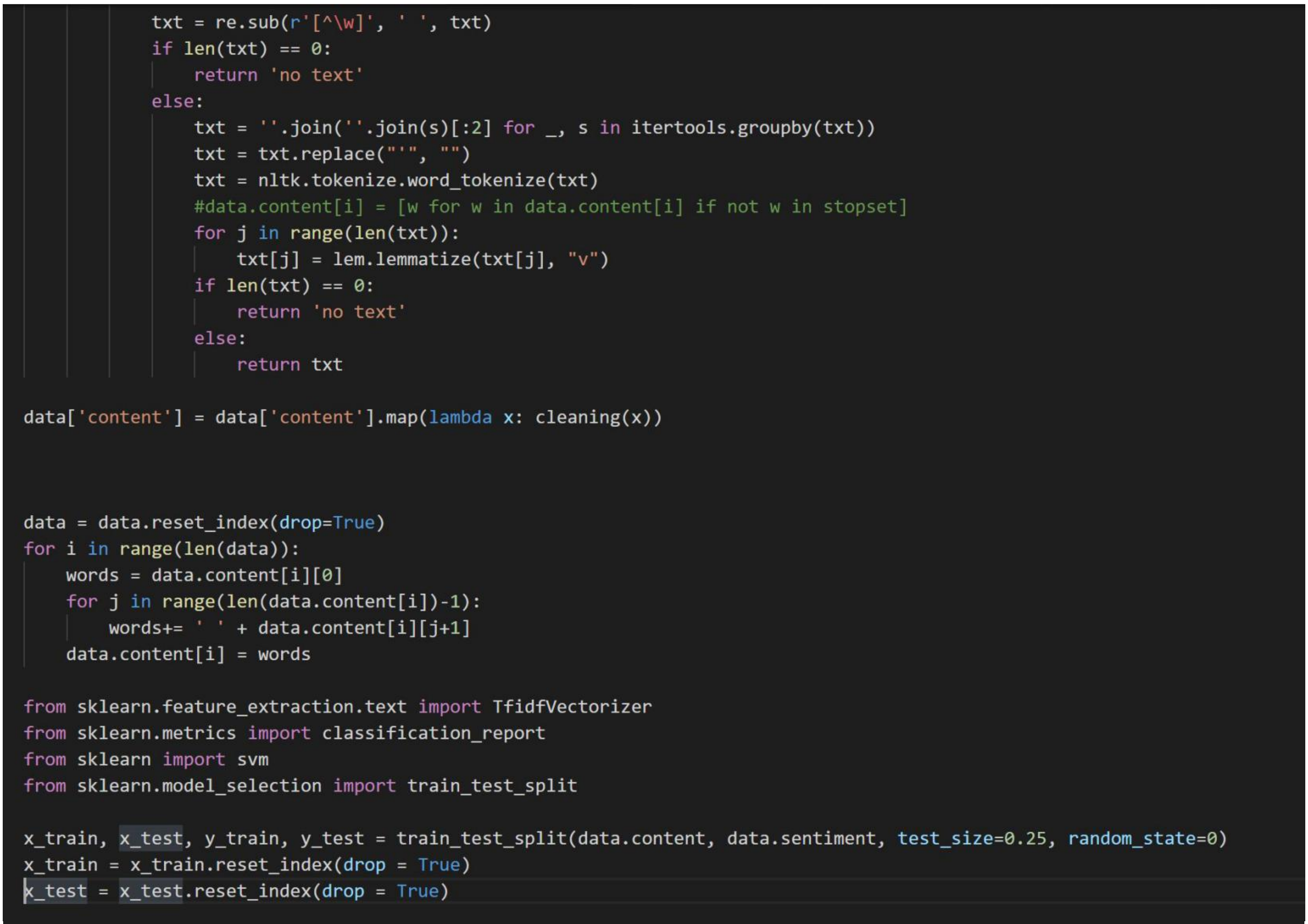
4.3 The Algorithm

The Most critical part of the system is the emotion classification algorithm. This algorithm is implemented using Python. Before analyzing the tweets, there is some pre-work done on tweet contents. Tweet contains limited to 140 characters. Twitter allows post web link inside the tweets. Some of the parts of the tweet text doesn't help for the analyzing process. Most of the tweets contain URLs which link to additional information. URLs aren't providing any help to the emotion analyzing process. As a process of tweet cleaning, First, all the unwanted URLs remove from the tweets. Also the stop words have to removed from the tweet content.In computing, stop words are words which are filtered out before or after processing of natural language data. Though "stop words" usually refers to the most common words in a language, there is no single universal list of stop words used by all natural language processing tools, and indeed not all tools even use such a list. Some tools specifically avoid removing these stop words to support phrase search.

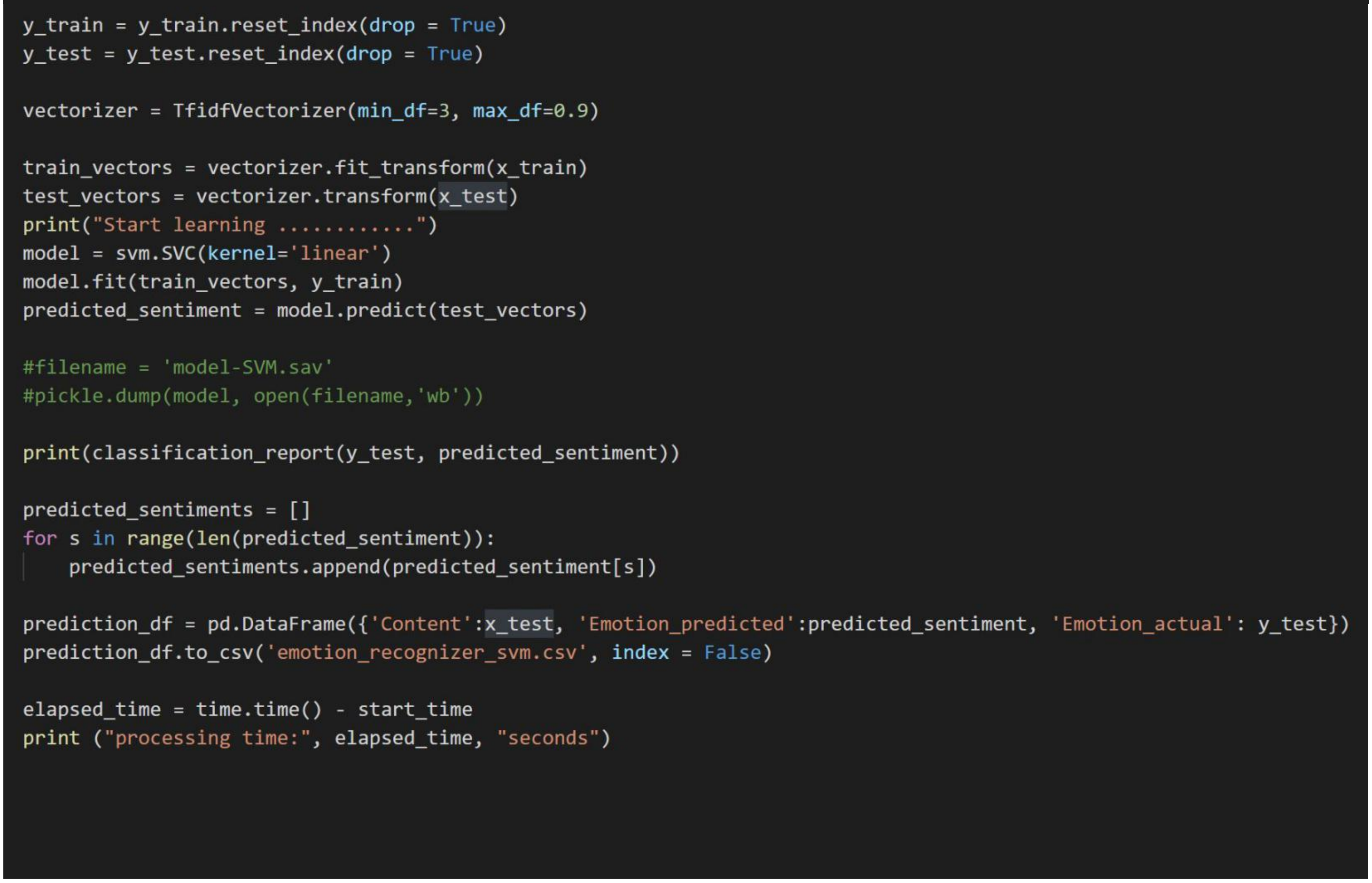
Tweet collector, Tweet analyzer and the Visualizer are the main subsystems of the SentViz. The functionality of the tweet collector is to collect the tweets from the Twitter stream API.Standard stream API key use for connecting to the twitter stream api.Tweet collector implemented as an Azure web job. Every 6 minutes the Tweet collector web job runs and collects the tweets. Then tweet data are cleaned to analyze.



```python
import pandas as pd
import re
import numpy as np
import nltk
import time
import pickle
import itertools
import os

start_time = time.time()
data = pd.read_csv('text_emotion.csv')
stopset = set(stopwords.words('english'))
from nltk.stem.wordnet import WordNetLemmatizer
lem = WordNetLemmatizer()

#comprehensive cleaning
def cleaning(text):
    txt = str(text)
    txt = re.sub(r"http\S+", "", txt)
    if len(txt) == 0:
        return 'no text'
    else:
        txt = txt.split()
        index = 0
        for j in range(len(txt)):
            if txt[j][0] == '@':
                index = j
        txt = np.delete(txt, index)
        if len(txt) == 0:
            return 'no text'
        else:
            words = txt[0]
            for k in range(len(txt)-1):
                words+= " " + txt[k+1]
            txt = words
```



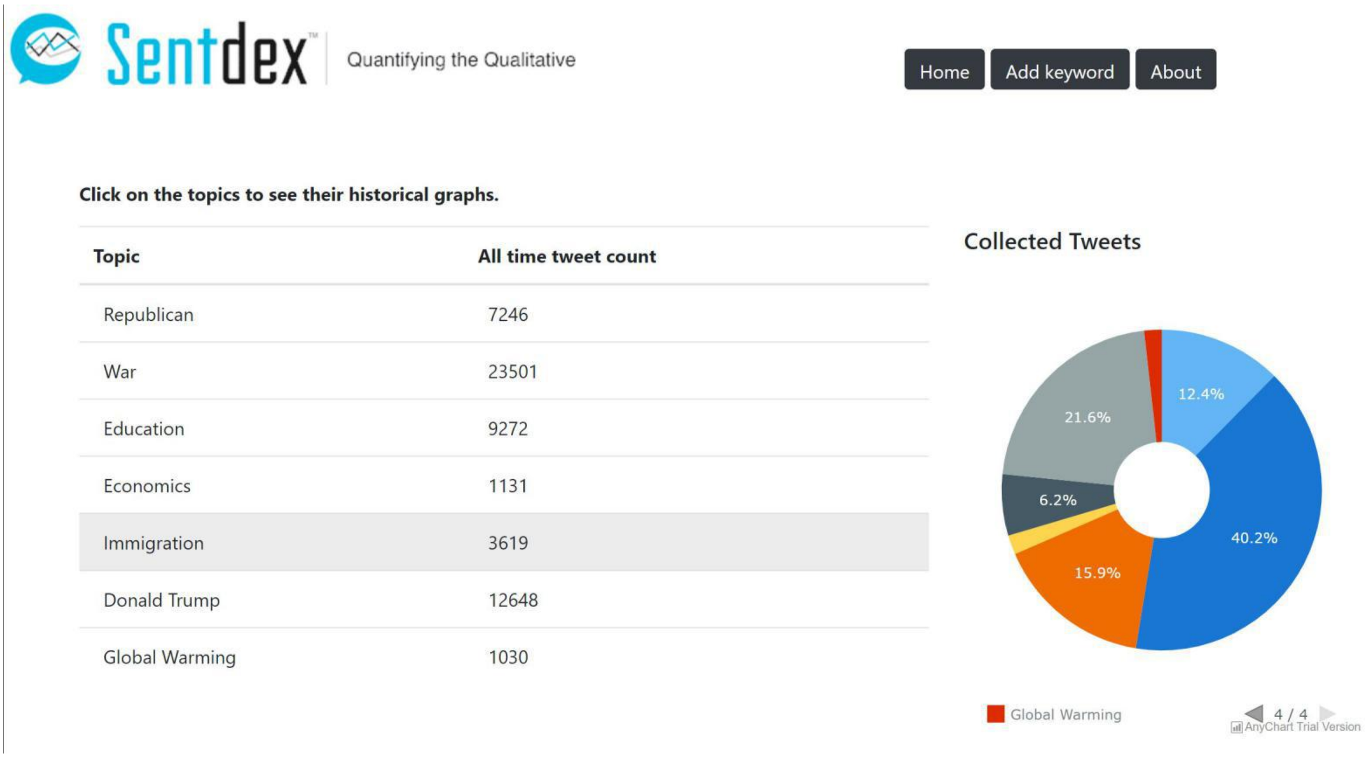
```
            txt = re.sub(r'[^\w]', ' ', txt)
            if len(txt) == 0:
                return 'no text'
            else:
                txt = ''.join(''.join(s)[:2] for _, s in itertools.groupby(txt))
                txt = txt.replace("'", "")
                txt = nltk.tokenize.word_tokenize(txt)
                #data.content[i] = [w for w in data.content[i] if not w in stopset]
                for j in range(len(txt)):
                    txt[j] = lem.lemmatize(txt[j], "v")
                if len(txt) == 0:
                    return 'no text'
                else:
                    return txt

data['content'] = data['content'].map(lambda x: cleaning(x))



data = data.reset_index(drop=True)
for i in range(len(data)):
    words = data.content[i][0]
    for j in range(len(data.content[i])-1):
        words+= ' ' + data.content[i][j+1]
    data.content[i] = words

from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import classification_report
from sklearn import svm
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(data.content, data.sentiment, test_size=0.25, random_state=0)
x_train = x_train.reset_index(drop = True)
x_test = x_test.reset_index(drop = True)
```



```
y_train = y_train.reset_index(drop = True)
y_test = y_test.reset_index(drop = True)

vectorizer = TfidfVectorizer(min_df=3, max_df=0.9)

train_vectors = vectorizer.fit_transform(x_train)
test_vectors = vectorizer.transform(x_test)
print("Start learning ...........")
model = svm.SVC(kernel='linear')
model.fit(train_vectors, y_train)
predicted_sentiment = model.predict(test_vectors)

#filename = 'model-SVM.sav'
#pickle.dump(model, open(filename,'wb'))

print(classification_report(y_test, predicted_sentiment))

predicted_sentiments = []
for s in range(len(predicted_sentiment)):
    predicted_sentiments.append(predicted_sentiment[s])

prediction_df = pd.DataFrame({'Content':x_test, 'Emotion_predicted':predicted_sentiment, 'Emotion_actual': y_test})
prediction_df.to_csv('emotion_recognizer_svm.csv', index = False)

elapsed_time = time.time() - start_time
print ("processing time:", elapsed_time, "seconds")
```

4.4 Main Interfaces

The system hasn't separate users. When the web application loads, it shows the main page with shortcuts for the essential features of the system. All the keywords and the total number of tweets are viewed on the home page. Also, only one graph on the home page is a pie chart of the total number of tweets. The home page contains the link to the add keyword page and the system information page.



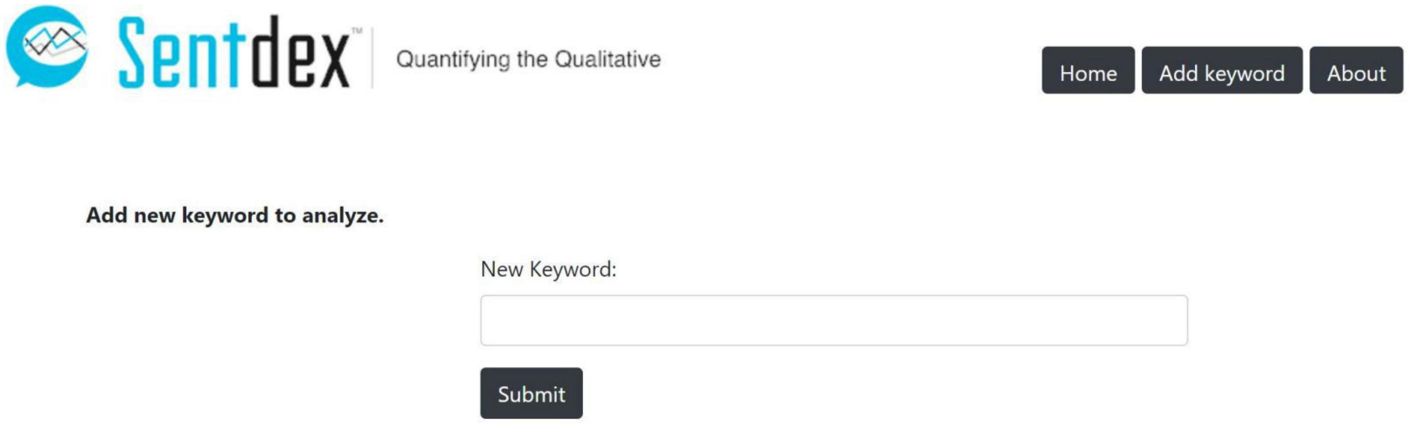
Click on the topics to see their historical graphs.

Topic	All time tweet count
Republican	7246
War	23501
Education	9272
Economics	1131
Immigration	3619
Donald Trump	12648
Global Warming	1030

Collected Tweets

Figure 4.1 - Homepage of a system

The home page shows the summary of collected data to the user. Table in the home page shows the each keyword save in the database and the their total number of tweets in the database. Also to get idea of the data of above table , the pie chart will shows the percentage of the tweets that store in the database.



Add new keyword to analyze.

New Keyword:

Figure 4.2 - Add keyword page

System provide a feature that can add new keywords to the system.Simple add keyword form will provide that feature.The only user data is saved to the system through this form.Also only user data validation is implement in here.

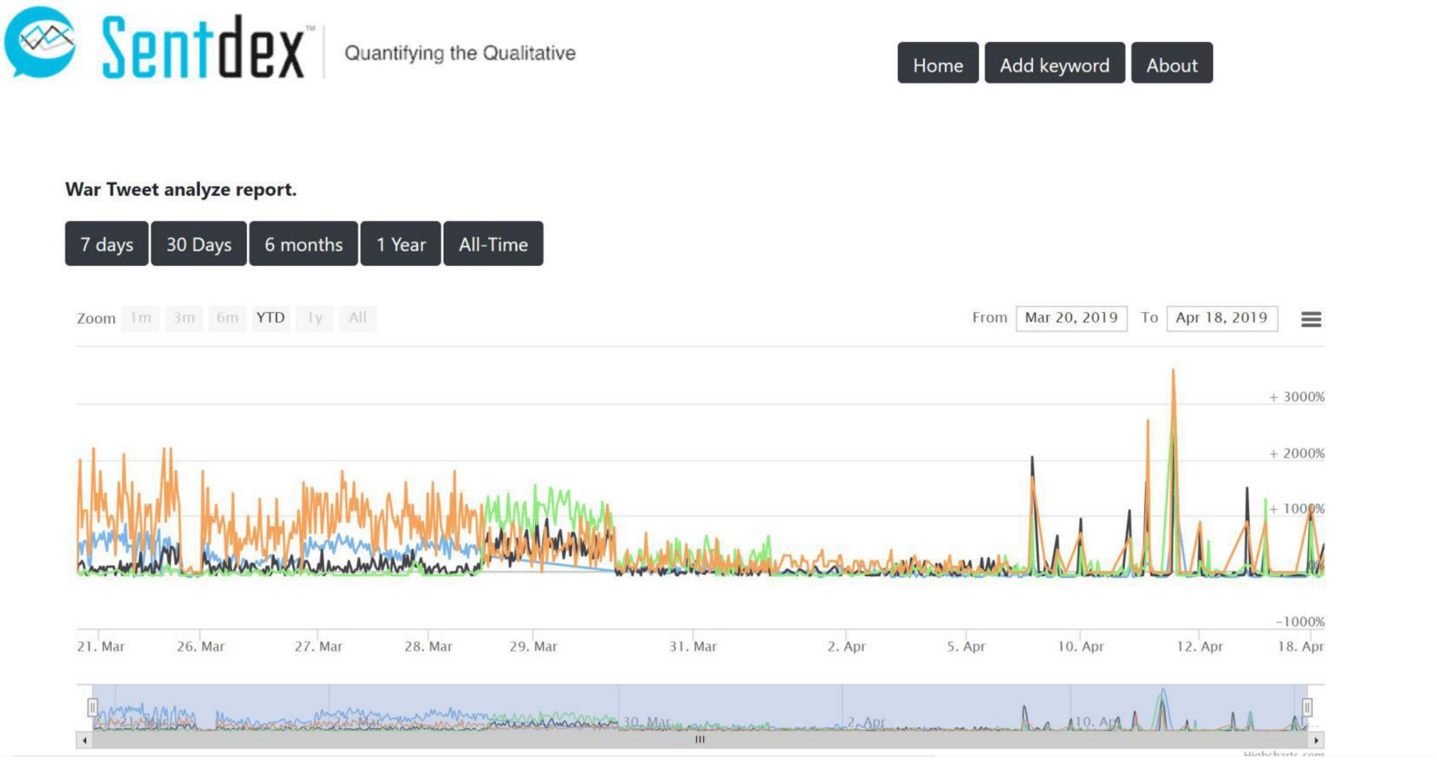


Figure 4.3 - Emotion timeline

The primary purpose of the system is to show how the Twitter community thinks about a particular keyword. This system shows three main graphs which are the emotion timeline, the emotion heat map, and the Emotion word cloud. Among these graphs, emotion timeline is the main part of the visualize subsystem. Emotion timeline shows the continuous graphs of the count of tweets of each emotion. The system classifies tweets into four emotion categories which are angry, happy, sad and fear. So the emotion timeline has four different graphs in one figure. Also, the bottom slide bar can use to view the particular period of the emotion timeline.

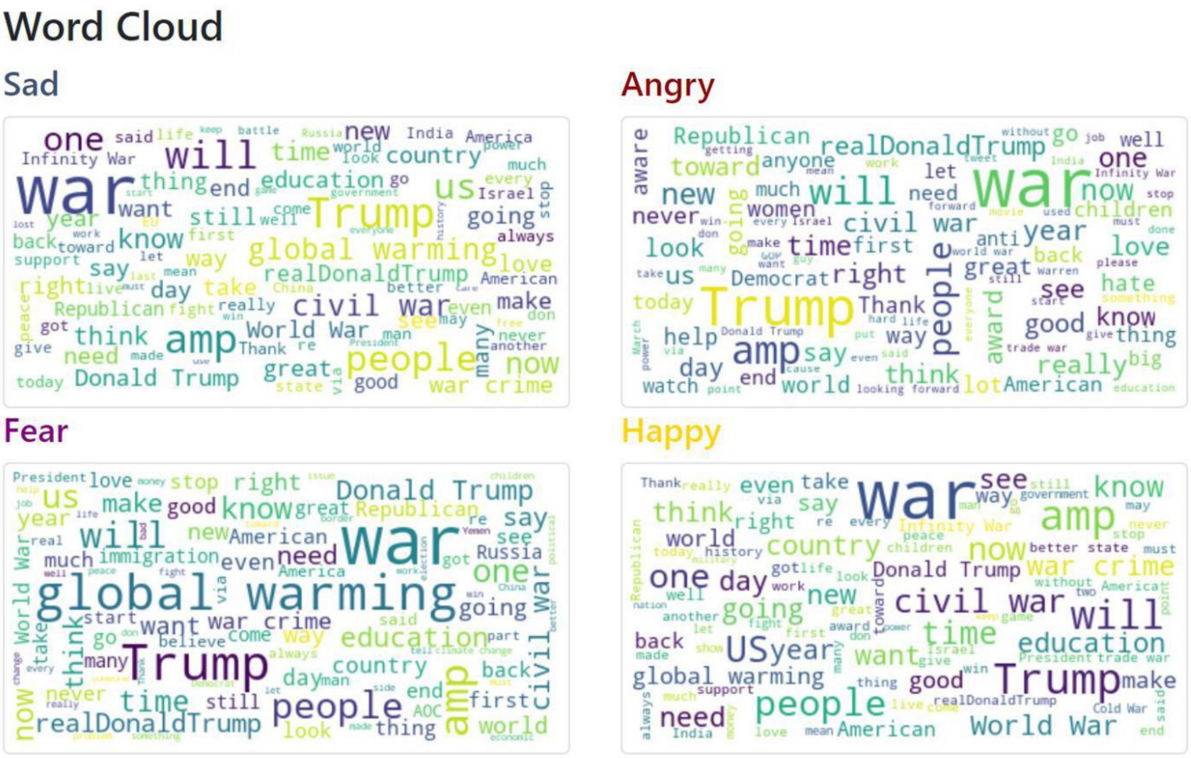
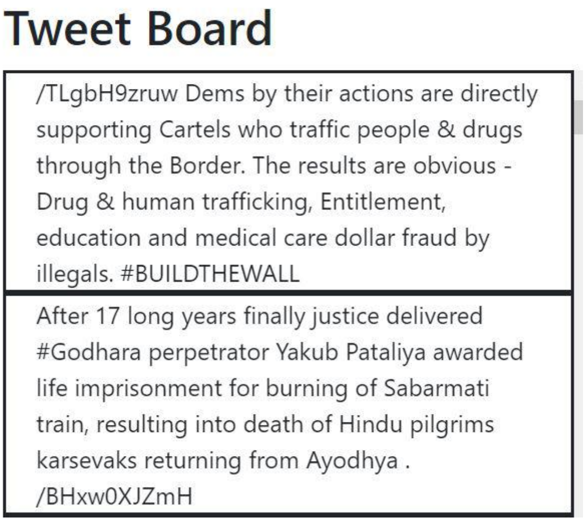


Figure 4.4 - Word Cloud

The Second graph of the system is the word cloud. A word cloud is a novelty visual representation of text data, typically used to depict keyword metadata on websites or to visualize free form text. The tags are usually single words, and the importance of each tag is shown with font size or color. In this system four word clouds are generated for a keyword. Word cloud shows the 3o most frequent words for each emotion. This format is useful for quickly perceiving the most prominent terms and for locating a term alphabetically to determine its relative prominence. When used as website navigation aids, the conditions are hyperlinked to items associated with the tag.



Tweet Board

/TLgbH9zruw Dems by their actions are directly supporting Cartels who traffic people & drugs through the Border. The results are obvious - Drug & human trafficking, Entitlement, education and medical care dollar fraud by illegals. #BUILDTHEWALL

After 17 long years finally justice delivered #Godhara perpetrator Yakub Pataliya awarded life imprisonment for burning of Sabarmati train, resulting into death of Hindu pilgrims karsevaks returning from Ayodhya . /BHxw0XJZmH

System collect tweets and save the content and metadata(geo coordinates) in the database. System contains an tweet board.This tweet board shows the 500 tweets for selected keyword and time period.

Figure 4.5 - Tweet Board

Emotion Heat Map

Tweet collector web job collects the tweet according to the saved keywords. Each tweet received through the Twitter stream API content metadata such as user-id, geo-coordinates, country, etc..To generates the emotion heat map analyze sub-system collect the geo-coordinates of each tweet. This graphs used a world map to show how emotions are spread over the globe. But there is a limitation for the emotion heat map. Most of Twitter users don't enable tag the geo-coordinates to the posted tweets. So some of the retrieved tweets didn't have the geo-coordinates where the user was posted that.

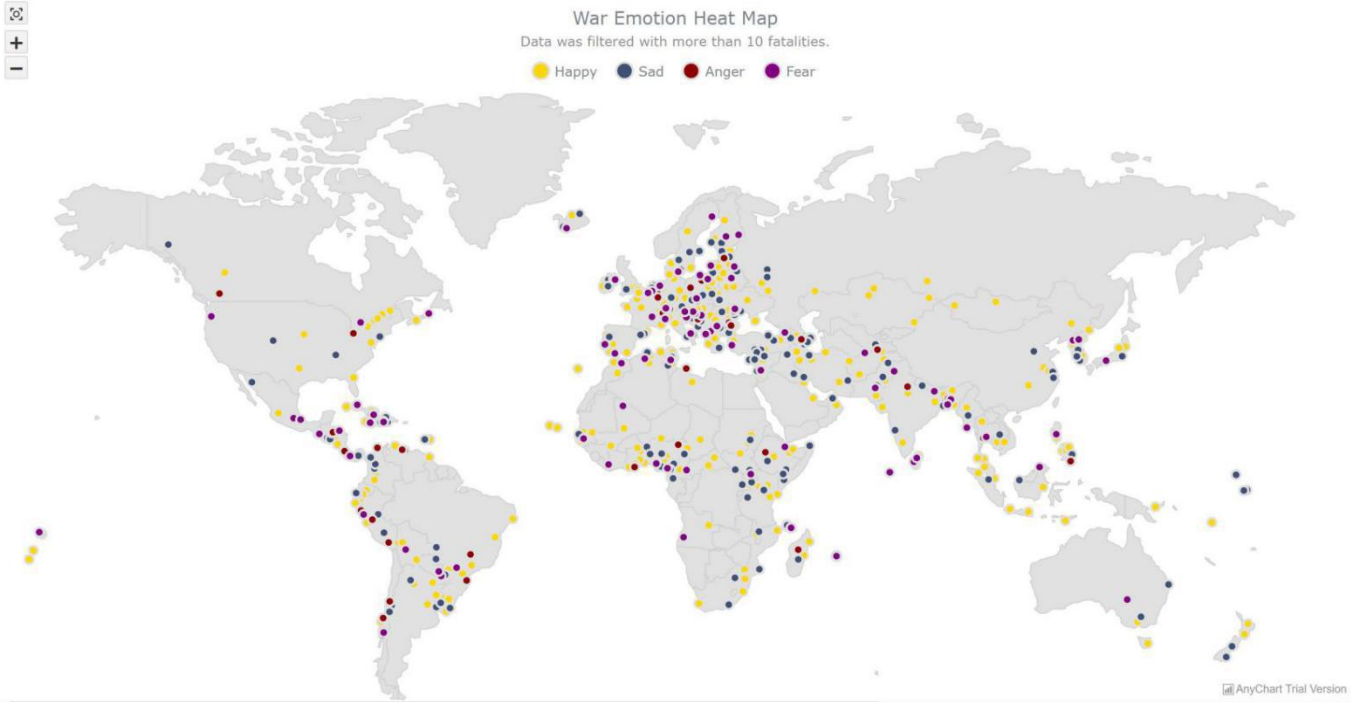


Figure 4.6 - Emotion Heat Map

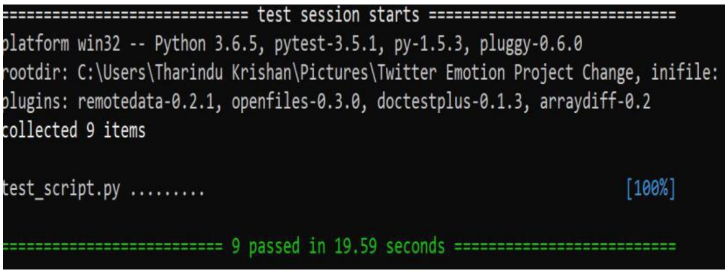
5. System Testing and Analysis

5.1 Testing Approach

To identify bugs and to see whether the system satisfies the client requirements as expected, every system should be fully tested. Also, SentViz was tested before to deployment. Once each component is finished. The iterative testing approach was done. It was tested before and after integrating with the whole system because there was a testing phase at the end of each iteration. Testing was covering four major aspects of the system; security testing, function testing,UI testing and performance testing . The outcome of each testing category is described below.

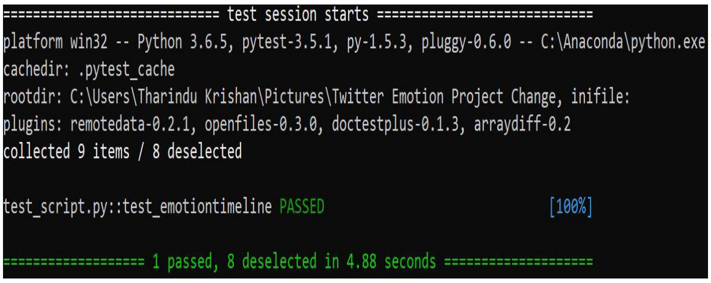
5.2 Unit Testing, Results and Analysis of Testing

Check individual component of the system and ensure they perform as intended is called unit testing. Unit testing is involved when doing function testing. All the data access objects and their methods were tested by unit testing. Also, the emotion classification algorithm which is the main core of the system was also tested. Unit testing was performed using Pytest library , Selenium and Eclipse IDE 8.0.



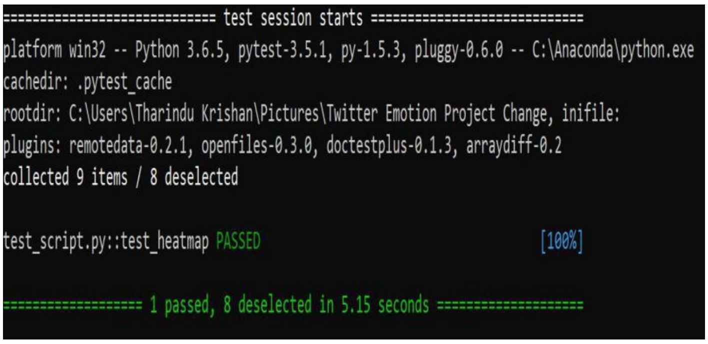
```
============================ test session starts =============================
platform win32 -- Python 3.6.5, pytest-3.5.1, py-1.5.3, pluggy-0.6.0
rootdir: C:\Users\Tharindu Krishan\Pictures\Twitter Emotion Project Change, inifile:
plugins: remotedata-0.2.1, openfiles-0.3.0, doctestplus-0.1.3, arraydiff-0.2
collected 9 items

test_script.py .........                                              [100%]

========================== 9 passed in 19.59 seconds ==========================
```

Figure 5.1 - Test Result



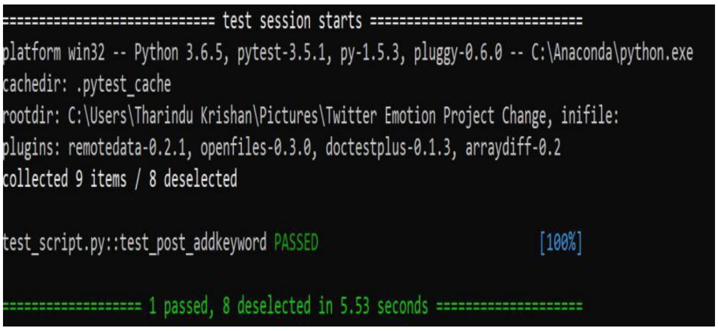
```
============================ test session starts ============================
platform win32 -- Python 3.6.5, pytest-3.5.1, py-1.5.3, pluggy-0.6.0 -- C:\Anaconda\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\Tharindu Krishan\Pictures\Twitter Emotion Project Change, inifile:
plugins: remotedata-0.2.1, openfiles-0.3.0, doctestplus-0.1.3, arraydiff-0.2
collected 9 items / 8 deselected

test_script.py::test_emotiontimeline PASSED                            [100%]

=================== 1 passed, 8 deselected in 4.88 seconds ===================
```

Figure 5.2 - Test Result for emotion_timeline



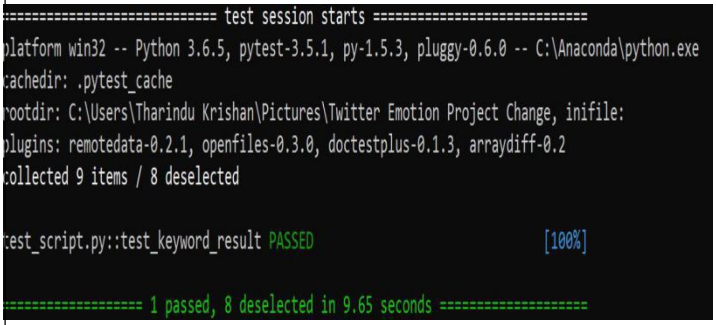
```
============================ test session starts ============================
platform win32 -- Python 3.6.5, pytest-3.5.1, py-1.5.3, pluggy-0.6.0 -- C:\Anaconda\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\Tharindu Krishan\Pictures\Twitter Emotion Project Change, inifile:
plugins: remotedata-0.2.1, openfiles-0.3.0, doctestplus-0.1.3, arraydiff-0.2
collected 9 items / 8 deselected

test_script.py::test_heatmap PASSED                                    [100%]

=================== 1 passed, 8 deselected in 5.15 seconds ===================
```

Figure 5.3 - Test result for heatmap

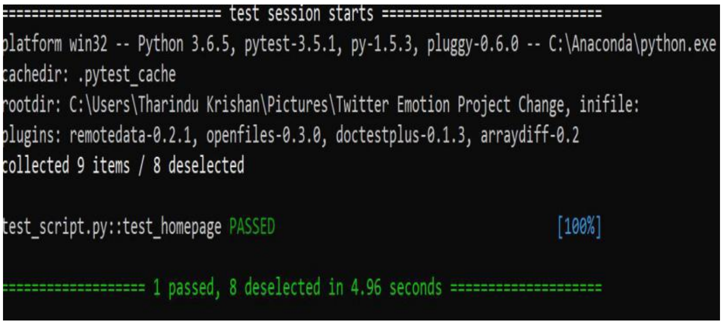


```
============================ test session starts ============================
platform win32 -- Python 3.6.5, pytest-3.5.1, py-1.5.3, pluggy-0.6.0 -- C:\Anaconda\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\Tharindu Krishan\Pictures\Twitter Emotion Project Change, inifile:
plugins: remotedata-0.2.1, openfiles-0.3.0, doctestplus-0.1.3, arraydiff-0.2
collected 9 items / 8 deselected

test_script.py::test_post_addkeyword PASSED                            [100%]

=================== 1 passed, 8 deselected in 5.53 seconds ===================
```

Figure 5.4 - Test result for add keyword



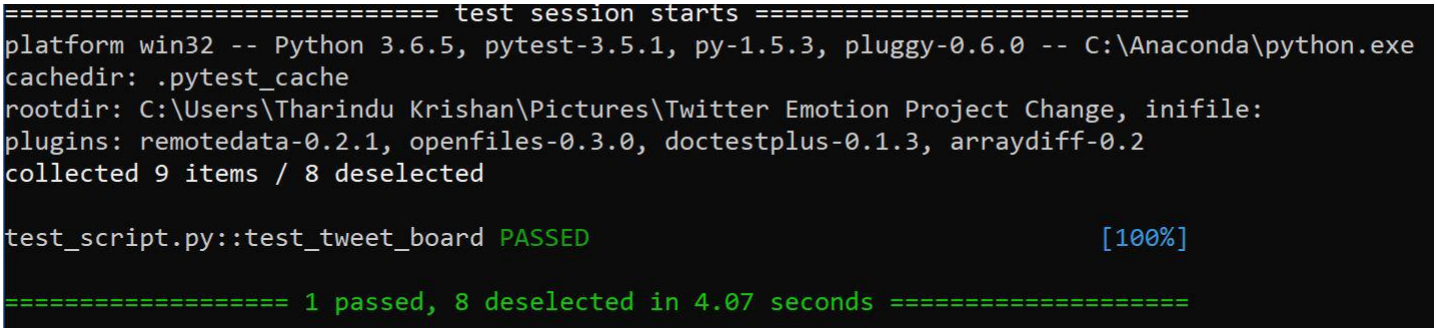
```
============================ test session starts ============================
platform win32 -- Python 3.6.5, pytest-3.5.1, py-1.5.3, pluggy-0.6.0 -- C:\Anaconda\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\Tharindu Krishan\Pictures\Twitter Emotion Project Change, inifile:
plugins: remotedata-0.2.1, openfiles-0.3.0, doctestplus-0.1.3, arraydiff-0.2
collected 9 items / 8 deselected

test_script.py::test_keyword_result PASSED                             [100%]

=================== 1 passed, 8 deselected in 9.65 seconds ===================
```



```
============================ test session starts ============================
platform win32 -- Python 3.6.5, pytest-3.5.1, py-1.5.3, pluggy-0.6.0 -- C:\Anaconda\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\Tharindu Krishan\Pictures\Twitter Emotion Project Change, inifile:
plugins: remotedata-0.2.1, openfiles-0.3.0, doctestplus-0.1.3, arraydiff-0.2
collected 9 items / 8 deselected

test_script.py::test_homepage PASSED                                   [100%]

=================== 1 passed, 8 deselected in 4.96 seconds ===================
```

Figure 5.5 - Test result for keyword result

Figure 5.6 - Test result for homepage response



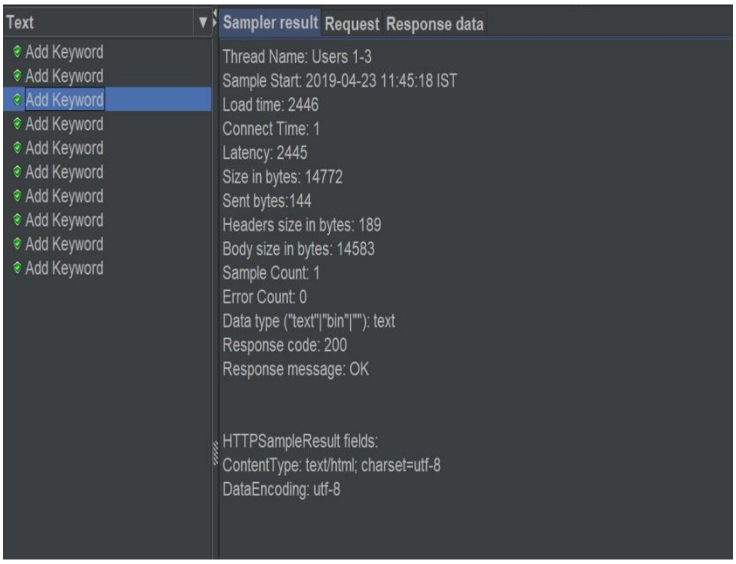
```
============================= test session starts =============================
platform win32 -- Python 3.6.5, pytest-3.5.1, py-1.5.3, pluggy-0.6.0 -- C:\Anaconda\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\Tharindu Krishan\Pictures\Twitter Emotion Project Change, inifile:
plugins: remotedata-0.2.1, openfiles-0.3.0, doctestplus-0.1.3, arraydiff-0.2
collected 9 items / 8 deselected

test_script.py::test_tweet_board PASSED                                 [100%]

=================== 1 passed, 8 deselected in 4.07 seconds ====================
```

Figure 5.7 - Test result for Tweet board

5.3 Aspects Related to Performance, User Interface, Security and Failures

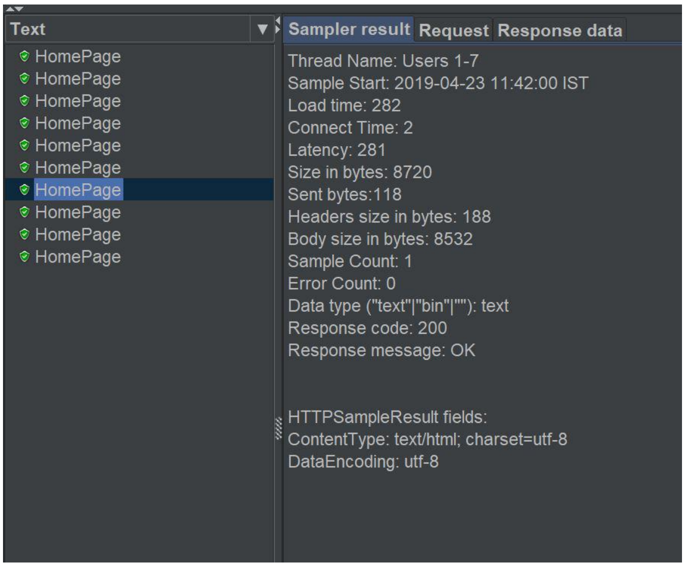
Verify whether the system has satisfied the performance requirements is the main objective of performance testing Performance testing was done by using JMeter library. Most critical functionalities such as retrive tweets from stream api, add new keywords and store tweets, emotion result response time in the database was tested to see the response time of them is acceptable or not. A maximum elapsed time (m) was set for each function to achieve that. Then the test-target and timer are started at the same time. If the execution time is less than or equal to maximum time then the test was considered to be succeeded. If time was exceeded then test consider as failed. The major difficulty of performance testing occured when testing the request time for the result of the keyword. The estimated maximum elapsed time as it varies with the time period. Initially the response time for result page were tested for different time periods(7 days,30 days, 6 months). The tests were performed using the same machine at different time slots.

UI testing was performed to ensure that system navigation happens as expected and requirements correctly and that the elements in the pages function as desired.Page navigation was tested for all the user interfaces.JUnit and Selenium libraries were used for UI testing. Furthermore, all the elements in the web pages such as text fields and buttons were tested to see whether they provide correct navigations.The system has to fix some bugs because Some test cases gave unexpected results.They were corrected and re-tested. User acceptance testing done by several students and few changes were made based on their feedback. Hence the usability of the system is at an acceptable level.Application level security and system level security are two areas were focused on security and access control testing. The access to data and functionalities considered in application-level security. Logging into the system is considered by system-level security. Testing was done by using JUnit and Selenium libraries. Some test cases gave unexpected results, which were corrected and re-tested.Security aspect of the system is in acceptable level because of all test are successfully passed.



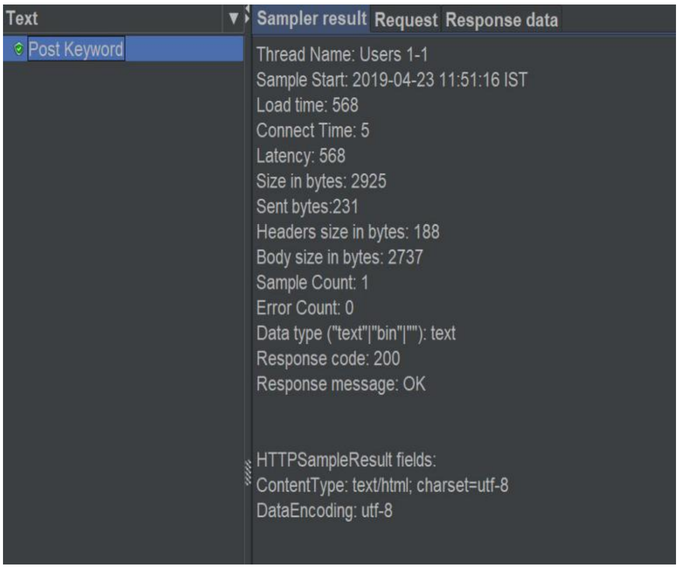
```
Thread Name: Users 1-3
Sample Start: 2019-04-23 11:45:18 IST
Load time: 2446
Connect Time: 1
Latency: 2445
Size in bytes: 14772
Sent bytes:144
Headers size in bytes: 189
Body size in bytes: 14583
Sample Count: 1
Error Count: 0
Data type ("text"|"bin"|""): text
Response code: 200
Response message: OK


HTTPSampleResult fields:
ContentType: text/html; charset=utf-8
DataEncoding: utf-8
```

Figure 5.8 - Performance test-Add keyword



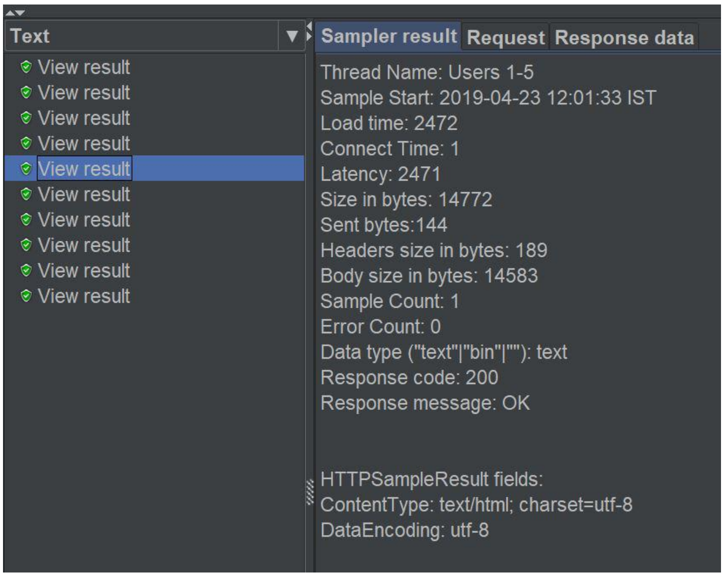
```
Thread Name: Users 1-7
Sample Start: 2019-04-23 11:42:00 IST
Load time: 282
Connect Time: 2
Latency: 281
Size in bytes: 8720
Sent bytes:118
Headers size in bytes: 188
Body size in bytes: 8532
Sample Count: 1
Error Count: 0
Data type ("text"|"bin"|""): text
Response code: 200
Response message: OK


HTTPSampleResult fields:
ContentType: text/html; charset=utf-8
DataEncoding: utf-8
```

Figure 5.9 - Performance test-Home Page



```
Thread Name: Users 1-1
Sample Start: 2019-04-23 11:51:16 IST
Load time: 568
Connect Time: 5
Latency: 568
Size in bytes: 2925
Sent bytes:231
Headers size in bytes: 188
Body size in bytes: 2737
Sample Count: 1
Error Count: 0
Data type ("text"|"bin"|""): text
Response code: 200
Response message: OK


HTTPSampleResult fields:
ContentType: text/html; charset=utf-8
DataEncoding: utf-8
```

Figure 5.10 - Performance test-Post keyword



```
Thread Name: Users 1-5
Sample Start: 2019-04-23 12:01:33 IST
Load time: 2472
Connect Time: 1
Latency: 2471
Size in bytes: 14772
Sent bytes:144
Headers size in bytes: 189
Body size in bytes: 14583
Sample Count: 1
Error Count: 0
Data type ("text"|"bin"|""): text
Response code: 200
Response message: OK


HTTPSampleResult fields:
ContentType: text/html; charset=utf-8
DataEncoding: utf-8
```

Figure 5.11 - Performance test-View Result

6. Conclusion and Future Work

Twitter is the most popular microblogging social media.In 2019 twitter community have 320 million users and 180 million active users. Each day 50 million tweets post by users. Most of the tweets contain information about people opinions on various products, services and topics. Twitter can use for identifying the new trending topics among the people.By analyzing the tweets could get more valuable information about people opinion. This information is important for the business. The SentViz system analyzes the tweets and visualized the emotional data in various graphs which can understand easily. Also, the system provides more information on a given topic(keyword) more than sentiment analysis.The system categories tweets into four main emotion categories.

Future Work:

This system uses Support vector machine model to classify the tweets into four emotion classes. When the system is designed various machine learning algorithms are implemented and tested the accuracy. The first algorithm was the Naive Bayes algorithm. This algorithm is very simple and easy to implement. But when the accuracy has measured the accuracy of the test set is between 20%-30%. Because of the less accuracy, the Naive Bayes approach is considered as a failure algorithm and moved to SVM(support vector machine). This system has an accuracy between 80%-90%.

But the more accuracy can be achieved using the neural network. Instead of using Support Vector Machine, Convolutional neural network can give accuracy higher than 90%.The system can be extended to use the convolutional neural network in the future. Still, there is no functionality to save the result as a pdf or word document. The system can be add new functinality to save result as a pdf.

References

[1]. “Csc2.ncsu.edu.” (2019). Tweet Sentiment Visualization App. [online] Available at: https://www.csc2.ncsu.edu/faculty/healey/tweet_viz/tweet_app/ [Accessed 23 Apr. 2019].

[2]. “AnyChart.js,” [Online]. Available: https://www.anychart.com/. [Accessed 23 October 2019].

[3]. “HighChart.js,” [Online]. Available: https://highcharts.com/stock/. [Accessed 23 October 2019].

[4]. “Sentdex,” [Online]. Available: https://sentdex.com. [Accessed 23 October 2019].

[5]. En.wikipedia.org. (2019). *Emotion classification*. [online] Available at: https://en.wikipedia.org/wiki/Emotion_classification [Accessed 23 Apr. 2019].

[6]. “Pytest,” [Online]. Available: https://docs.pytest.org/en/latest/. [Accessed 23 October 2019].

[7]. “JMeter,” [Online]. Available: https://jmeter.apache.org. [Accessed 23 October 2019].

[8]. “Selenium,” [Online]. Available: https://www.seleniumhq.org/. [Accessed 23 October 2019]