Design Manual

Student: Ignas Usinskas
Student ID: C00166783
Supervisor: Greg Doyle
Date: 15/04/2016

Table of Contents

Introduction

The purpose of this document is to demonstrate how the twitter sentiment analysis web application works. This document takes the functionality specified in functional specification document and provides detailed description on how it is implemented.
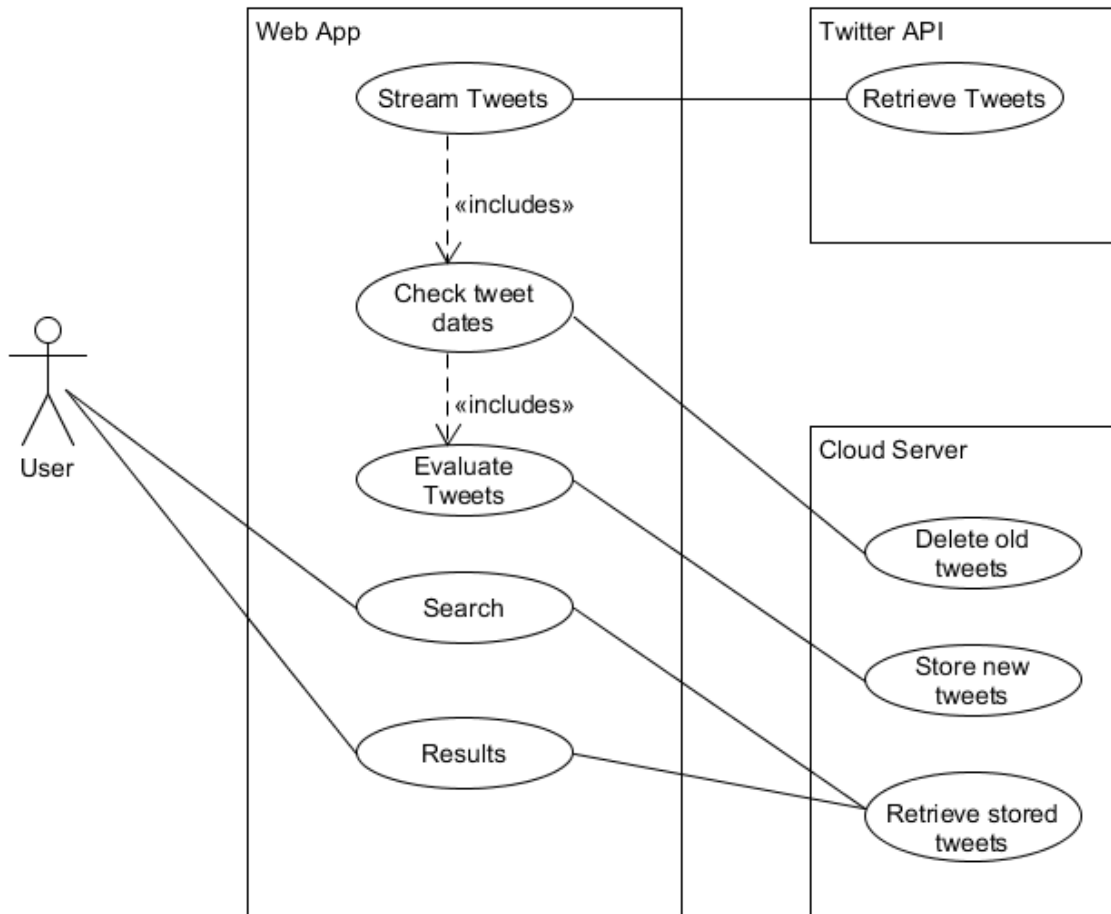
Scope

The scope of this document is the complete description of this web application. This document provides information on architecture design, class diagram, use cases, database design and user interface. The application is developed accordingly to this document.

Overview

The aim of this project is to develop a web application which would stream tweets from twitter, evaluate those tweets using sentiment analysis and store them on MySQL database. This application will provide a search page where users will be able to search something and get a complete evaluation of how people on twitter feel about the searched query. The information will be displayed in a separate window containing - graph, searched term, overall evaluation, tweet count and all relevant tweets with their individual evaluations.

Architecture Design

Use Case Diagram

Class Diagram

```
+---------------------------------------+
|   MyStreamListener(StreamListener)     |
+---------------------------------------+
|                                        |
+---------------------------------------+
| old_tweets()                           |
| on_data(self, data)                    |
| on_status(self, status)                |
+---------------------------------------+
                    |
                    |  *
         sent_mod.sentiment(tweet)
                    |
                    ▽ 1
+---------------------------------------+
|          Vote(ClassifierI)             |
+---------------------------------------+
|                                        |
+---------------------------------------+
| __init__(self, *classifiers)           |
| classify(self, features)               |
| confidence(self, features)             |
+---------------------------------------+
```
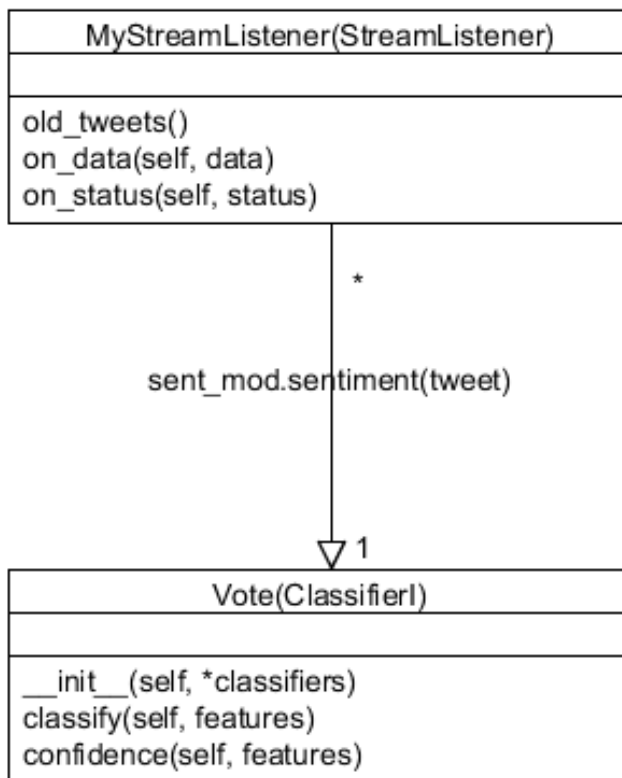
This web app is written in python, as such – it consists of only two classes, where the rest of the code is functions outside these classes. The two classes interact as shown in the class diagram.
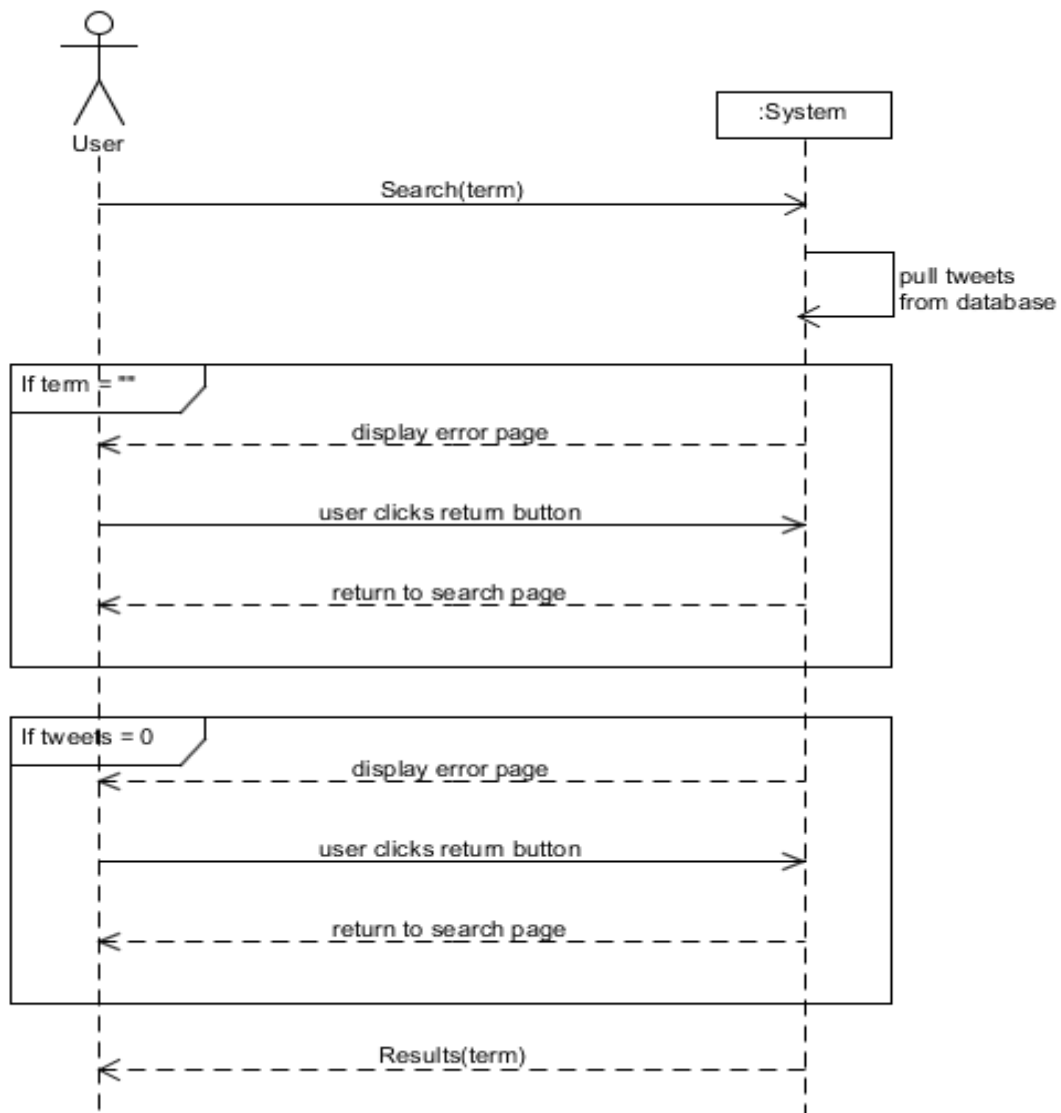
Use Cases

Actors: User

Use Case:

1. This use case begins when user wishes to search for a specific term.
2. After user inputs the search term and presses the search button, the web application pulls tweets from the database.
3. Tweets containing the search term are selected.
4. Results are displayed in the results page.

System Sequence Diagrams

Search

The streaming, sentiment analysing and storing tweets are done by the application itself, thus there is no user to system interaction.

Database Design

Twitter sentiment analysis web application will stream tweets using Tweepy and store them in MySQL database. MySQL database together with the web application are hosted on the same private server.
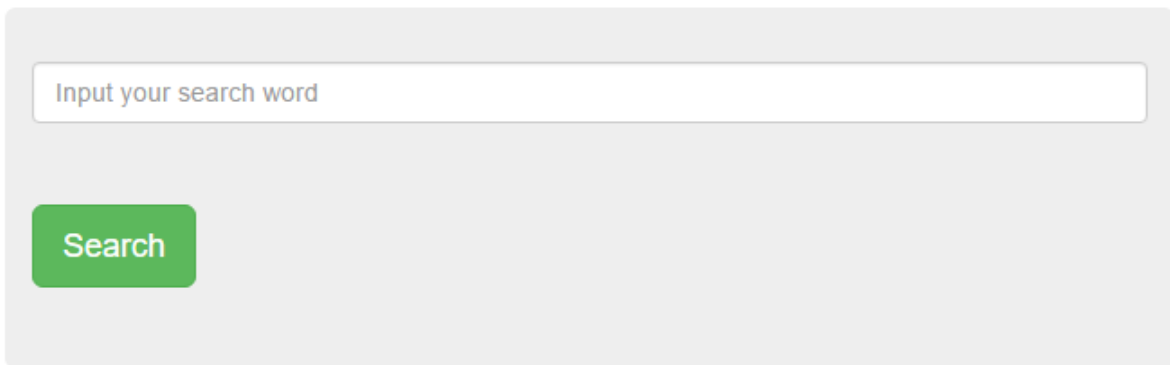
Tweets

Property	Data Type	Comment
my_date	varchar	Date when tweet was stored
tweet	varchar	Text of the tweet
sentiment	varchar	Sentiment evaluation (positive, negative or neutral)
confidence	varchar	Condifidence in the evaluation of each tweet – represented from 0 to 100

User Interface

Search Page

Search Page

Input your search word

Search

A simple page where users enter their search terms and press search button do get their results.

Error page

Empty Search

Return

Oops! You did not enter anything, please go back and try again!

This page appears when user forgets to enter anything and just presses enter.
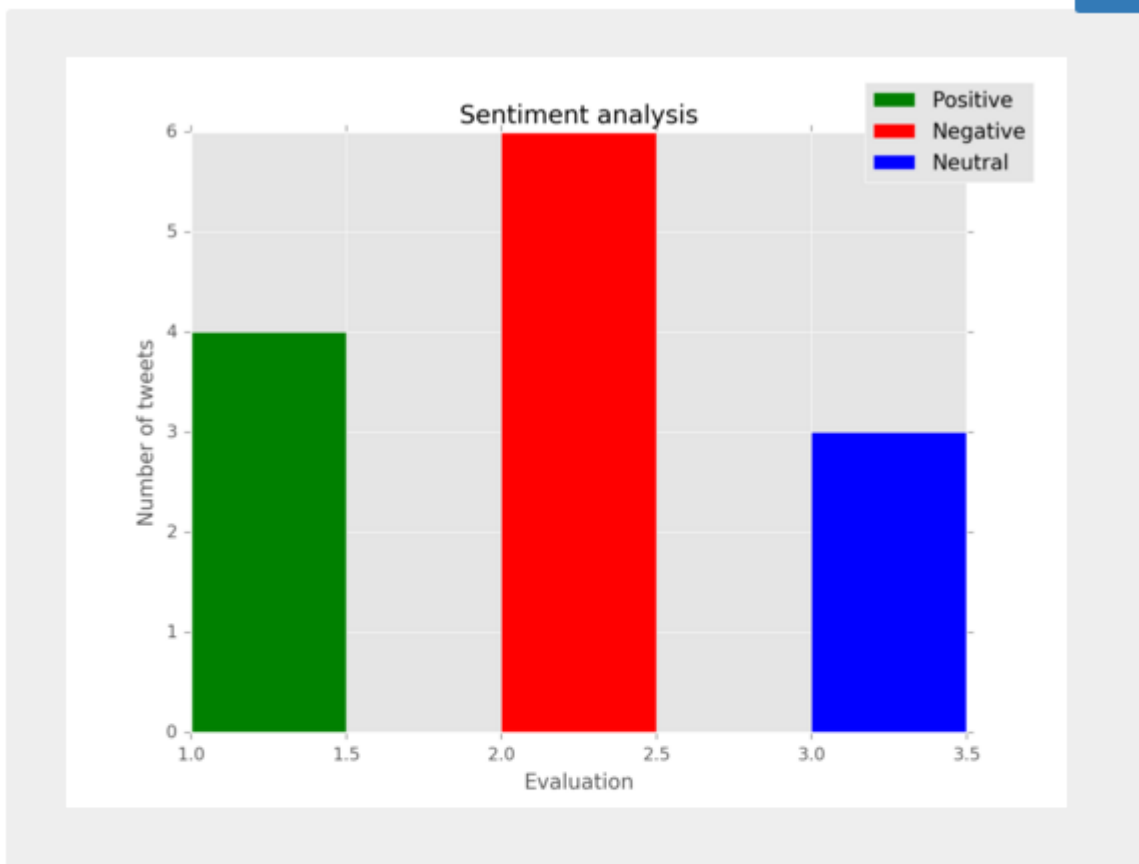
Error page

No Tweets

Oops! There were no tweets found according to your search, please go back and try to search for something else!

This page appears if the search term user has entered was not found in any tweets.

Results Page

Results Page

You searched for: car.

Overall evaluation: **Negative... :(**

Count of tweets: Evaluated the total of 13 tweets, out of which: 6 were negative, 4 were positive and 3 were neutral.

Tweets: 'DATE', 'TWEET', 'SENTIMENT', 'PERCENTAGE IN EVALUATION OF TWEET'

('2016-04-14', '@GardaTraffic not careless ?? He was putting every bit of care n skill into that. You try it', 'negative', '100.0')

('2016-04-14', 'Like ask me my fave artist, I have loads, I cannot pick just 1. What career do I want? Amm makeup artist, DJ, photography, producer... why ?', 'negative', '100.0')

('2016-04-14', 'My cousin got this mad creepy text from a stranger trying to buy his car. https://t.co/9zxDF5GN1a', 'negative', '57.14285714285714')

('2016-04-14', '#LFC need a to carry a health warning. #EuropeanAnfieldNights.', 'neutral', '100.0')

('2016-04-14', 'Sakho scares me.', 'negative', '85.71428571428571')

('2016-04-14', 'Men gave me cars, houses,expensive fresh fruit & veg., new knickers, ETC...afternoon & nights out. I gave them my pussy & sad/happy humor.', 'neutral', '85.71428571428571')

('2016-04-14', '@theamandabatty But it is the UCI, so you can make it as safe as you want, as long as you are happy for cars and moto's to run you over', 'positive', '85.71428571428571')

('2016-04-14', '@ghoner all my male relatives are in cardiac arrest right now?', 'neutral', '71.42857142857143')

('2016-04-14', '#€1million but your first child will be the reincarnation of #Hitler. Yes or no?', 'positive', '57.14285714285714')

('2016-04-14', '@garyjamesmagic Thanks for the majestic following, pure magic. @infochriscarrol https://t.co/uTjbWaNZPa', 'positive', '85.71428571428571')

('2016-04-14', 'Benteke sorry to say it, your #lfc career is over. #avfc', 'negative', '85.71428571428571')

('2016-04-14', '@stevecarey33 thanks Steve', 'positive', '85.71428571428571')

Results page shows up after user enters a search term and relevant tweets were found. Here we can see that the user entered 'car' as a search term, all tweets that contained a word 'car' were displayed here. Graph divides tweets into three categories: positive, negative and neutral with a separate bar and a distinct colour for each category. This graph displays the number of tweets belonging to each category.

Under the graph, user can see what they search for, total evaluation of their search term, how many tweets were used in this evaluation in total and how many tweets belong to each category.

Lastly, the user can see the tweets that were used in this evaluation with their individual evaluations and confidence scores.

In terms for GUI (Graphical User Interface) – there is not much in this web application since most of the work is done in the background by the application. User just has to enter a search term, the application provides user with graphical results and does the rest without the user knowing about it.

Non user part

Streaming analysing and storing tweets

The application connects to twitter and starts streaming tweets according to Ireland's geo location. As tweets are being streamed, they are put through sentiment analysis module which analyses every tweet and determines if it is positive, negative or neutral. Once analysis is done – tweets are stored in database with their sentiment evaluation and confidence for that evaluation. All this is done by the application as soon as 'sent.py' is run. User is not aware of this functionality as this is a separate part application run by the owners of the entire application.

This process can be run for at least a few hours each day. Once the streaming limit is reached – Twitter disconnects the stream. Once that happens, it is best to not try and reconnect, because Twitter might black list you.

Usage instructions

The application consists of two parts:
1. Sent.py – is the streaming part of the application. Once sent.py is run, the application will connect to twitter using access keys and tokens. The streaming filter is set up to stream Ireland's GEO location. Other than having MySQL database set up and ready, nothing else needs to be done or adjusted when this file is run, provided all required libraries are preinstalled. This part of the application is online on pythonanywhere.com server, but it proves to be too heavy on CPU and cannot be run as it burns through the little CPU allowance. Thus it needs to be run locally.
2. Flask part of the application – provides the user interface to search for a keyword. Any tweets containing the search keyword will be graphed and displayed in the results page. If no tweets were found to contain the search keyword or user did not enter anything in the search field, an error page will be displayed to the user. This part of the application is run separately from the previous one and it also is on pythonanywhere.com server.

Acknowledgements

I would like to thank my supervisor Dr Greg Doyle for providing me with the opportunity to do this project. Thanks to that I was able to expand my knowledge and programing skills by working on something I have never done before. Thanks to my supervisor's constant request for progress feedback I was able to tackle and successfully achieve what I have achieved in the give period of time.
I would also like to thank my Web and Cloud Development lecturer Paul Barry for teaching me about python and databases which ended up using in the development of this project. I would also like to thank him for providing help in tackling some of the problems I have encountered along the way.
I would like to thank our group's head – Joseph Kehoe for all the support and schedules for this project he has provided.
I would like to thank all of my class mates who would give me a hit or a tip in solving particular problems.
Finally I would like to thank Stack Overflow, pythonprogramming.net and all other online resources that provide online information to tackle programming problems.