

An Analysis of the Relationship between Twitter feeds and Stock Market Movement

Philip Walsh

**Higher Diploma in Science in Data Analytics
National College of Ireland
2014**

Table of Contents

Abbreviations.....	4
Table of Figures.....	5
1 Executive Summary	6
2 Introduction	6
2.1 Existing APIs and Technologies.....	7
2.1.1 Twitter API	7
2.1.2 Yahoo API	8
2.2 Technologies Overview	8
2.3 Background and related work	8
3 System	11
3.1 Structure.....	11
3.2 Requirements Specification	11
3.2.1 Purpose	11
3.2.2 Project Scope	12
3.2.2.1 Scope	12
3.2.2.2 Project Objectives	12
3.2.2.3 Project Criteria	12
3.2.2.4 Project Supposition	12
3.2.2.5 Project Restrictions	12
3.2.2.6 Project Risk	13
3.2.2.7 Contingency Plans.....	13
3.2.2.8 Human Resource.....	13
3.2.3 Requirements.....	13
3.2.3.1 Functional Requirements	13
3.3 Design and Architecture.....	15
3.3.1 Use Case Models	15
3.3.1.1 Access Twitter API.....	15
3.3.1.1.1 Scope.....	15
3.3.1.1.2 Description	16
3.3.1.1.3 Use Case Diagram	16
3.3.1.2 Access the Yahoo finance API.....	16

3.3.1.2.1 Scope.....	16
3.3.1.2.2 Description	16
3.3.1.2.3 Use Case Diagram.....	17
3.3.1.3 Build a “Positive” and “Negative” word list	17
3.3.1.3.1 Scope.....	17
3.3.1.3.2 Description	17
3.3.1.3.3 Use Case Diagram.....	18
3.4 Implementation – Technologies and Methodologies.....	19
3.4.1 Technologies Overview.....	19
3.4.1.1 Python 2.7	19
3.4.1.2 SQLite	20
3.4.2 Methodologies	21
3.4.2.1 Accessing data through the Twitter API.....	21
3.4.2.2 Building a “Positive” and “Negative” word list for the model	24
3.4.2.3 Algorithm to calculate the sentiment rating of the data.....	29
3.4.2.4 Accessing stock trading data from Yahoo finance API.....	30
3.5 Testing.....	32
4 Results	33
5 Conclusions	36
5.1 Further Development of research	36
Appendix 1: Project Proposal.....	38
Appendix 2: Requirements Specification	44
Appendix 3: Code Segments	49
Bibliography	53

Abbreviations

API	Application Programming Interface
SQL	Structured Query Language
UML	Unified Modelling Language
AAPL	Apple.Inc
CSV	Comma Separated Values
REST	Representational State Transfer
URL	Uniform Resource Locator
RDMS	Relational Database Management System
HTML	Hyper Text Mark-up Language
IST	Irish Standard Time
EDT	Eastern Daylight Time
NYSE	New York Stock Exchange

Table of Figures

Figure 1: Sentdex sentiment index for American Tower Corporation (AMT)	9
Figure 2: Uncorrelated stock prices and twitter sentiment	10
Figure 3: System Architecture	15
Figure 4: UML Twitter API	16
Figure 5: UML Yahoo API	17
Figure 6: UML Word List	18
Figure 7: Python 2.7 source	19
Figure 8: Python Shell	20
Figure 9: SQLite source	20
Figure 10: SQLite browser	21
Figure 11: Twitter Developer page	21
Figure 12: Developer Account	22
Figure 13: Auth Codes	22
Figure 14: Twitter Libraries	23
Figure 15: Python libraries	23
Figure 16: Python code to access API	24
Figure 17: Print Tweets to console	24
Figure 18: Synonyms for "good"	25
Figure 19: Synonyms for "bad"	25
Figure 20: Create database code	26
Figure 21: Scraping and populating database	27
Figure 22: Word list in SQLite browser	28
Figure 23: Sentiment Rating Algorithm	29
Figure 24: Yahoo finance URL	30
Figure 25: AAPL Trading data output	30
Figure 26: Script to retrieve trading data	31
Figure 27: Subset of trading data	31
Figure 28: AAPL stock price movement	33
Figure 29: AALP stock price movement	34
Figure 30: Change in sentiment rating	34

1 Executive Summary

With the proliferation of Twitter as a source of data through its API and with the use of natural language processing, sentiment analysis has become an insightful way of understanding trends of public opinion and various subject matter. Twitter is used by millions daily thus providing an appropriate supply of relevant data for opinion mining. Techniques that have been developed with natural language processing provide the tools to analyse this data in a meaningful way.

In this paper, I investigate the relationship between Twitter feed content and stock market movement with respect to a specified stock. The specific aim is to identify if the sentiment information deduced from the Twitter feeds has a correlation with the stock price changes over a specified period of time and whether this could be used to predict future shifts in prices by analyzing and study different trends. To achieve this, a quantitative model is constructed to index the polarity of the sentiment of the data retrieved from the Twitter feed using Twitter's API. A numeric figure rating is assigned to the data so that an assessment can be made on any potential correlation. It is a time-series analysis and the data is sourced daily over a period of five days with a sentiment rating computed each day. The Yahoo API will be used to source the trade data for the stock over this period of time. The share price of the specified stock over this period of five days is graphed against the time-series sentiment rating of the stock, and inferences will then be made on the results of this.

2 Introduction

Stock market prediction is the act of trying to determine the future value of a company stock or other financial instrument traded on a financial exchange. Some suggest that stock prices are too unpredictable to forecast while others disagree, pointing to multiple methods which supposedly allow them to gain future price information.

There is no established or clear equation that helps anticipate how a stock price will behave. However, there are a number of factors that are known that influence why a stock's price moves up or down. These factors fall into three different categories: fundamental factors, technical factors and market sentiment. Market sentiment is considered to be a vague and difficult factor to grasp. Its influence is quite significant but is often subjective, biased and obstinate. (Harper, 2010) Sentiment analysis can contribute to creating a better understanding of this as factors such as politics, quant analysis, fundamental analysis and market conditions can pour into sentiment. Therefore, potentially, valuable information may be extracted from such analysis.

At the most basic level, sentiment analysis is an attempt to identify the overall mood, feeling and speculation of a text. The rise of social media such as blogs and social networks

has significantly increased interest in this type of analysis. Sentiment analysis involves building a system to collect and examine opinions about whatever topic has been discussed in such blog posts, comments, reviews or tweets.

There are different ways of analysing a body of text for sentiment or opinion. A method that I will use is a “Bag of words” model. This model focuses completely on the words, or sometimes a string of words, but usually pays no attention to the “context” so-to-speak.

This method is similar to a naïve Bayes approach as the model considers each of the words of the text as being independent of one another, each having as much meaning as the other. The “Bag of words” approach allows the model to have the necessary quantitative features required for the analysis.

A scoring system is established for each of the tweets using a pre-defined word list that represent individual positive and negative words. There will be list of “positive” words which will each have a rating of “+1” and a list of “negative” words which will have a rating of “-1”. Each tweet is tokenized and the items from the tweet are passed through the pre-defined word list. Any item that is in a given tweet that does not appear in the word list is discarded. A score is then accumulated of “positive” and “negative” words that correspond in the tweet to the word list. An average rating can be calculated from the set of tweets and provides a sentiment value for each day.

2.1 Existing APIs and Technologies

An API (application programming interface) is a software to software interface that allows for interaction between software components. It is a collection of different programming commands and standards for accessing a Web-based software application or Web tool. A software company releases its API to the public so that other software developers can design products that are powered by its service. (Roos, 2007)

2.1.1 Twitter API

According to (Strickland & Chandler, 2007), ‘Twitter bases its application programming interface (API) off the Representational State Transfer (REST) architecture.’ Data can be accessed and addressed through a network of designed applications which is principally REST architecture.

To access the Twitter API I will use a Python library called “Tweepy”. With this method, tweets on a particular subject or word and can be streamed and saved using the API’s search parameter.

2.1.2 Yahoo API

To obtain the relevant stock price trade data the Yahoo API is used. The Yahoo API allows access to as much as 1 year worth of trading data for a wide range of different stocks. To access this data a Python module was used that can fetch URLs. When the parameters for the desired stock and the range of trading data are identified, the Python module can fetch the URL from the Yahoo API with the required data within the HTML returned. This data can then be parsed from the here.

2.2 Technologies Overview

The approach taken for the development of this analysis required the use of different types of third party software tools. It required the acquisition, installation and configuration of these tools.

I chose Python as the programming language I would use for the analysis due to its practical functionality. It is a widely used, general purpose language that also supports multiple programming paradigms. There is also a large community base for open source software for Python with useful libraries available for download.

A database system was required for the storage and easy access to the word list that would be used for the sentiment rating of the tweets. I decided to use SQLite for this function. It is a relational database management system but in contrast to other RDMS such as MySQL, it is accessible locally and is not a separate process that can only be accessed from the client application. SQLite has an application browser locally that is easy to use and its basic functionality was deemed appropriate for the tasks required of this analysis.

Microsoft Office software tools Excel and Word were also used. The tweets that were streamed through the Twitter API were saved in CSV format and Excel was used to open the saved files.

Notepad++ was used to access the stock price charting data.

2.3 Background and related work

Without any prior education or deep knowledge of the financial markets industry, I had a curiosity about stock price markets and the factors that influence trading.

I came across a channel on Youtube called Sentdex. They are a business that provides big data analytics solutions, primarily in the field of natural language processing and sentiment analysis. Sentdex specializes in financial market sentiment, political topic and politician sentiment and business and sentiment.

‘We pull from over 20 major news sites (stocks, politics, general news), from even smaller news sites, and also process several million tweets a day from Twitter.’ (Sentdex, 2014)

Further research of the Sentdex website gave me the idea of experimenting with sentiment that could be extracted from tweets and varying stock market price changes. There are various tutorials regarding different techniques for sentiment analysis, specifically pulling data from Twitter, stock market data from Yahoo and algorithmic strategies. This information is accessible on Sentdex Youtube channel.

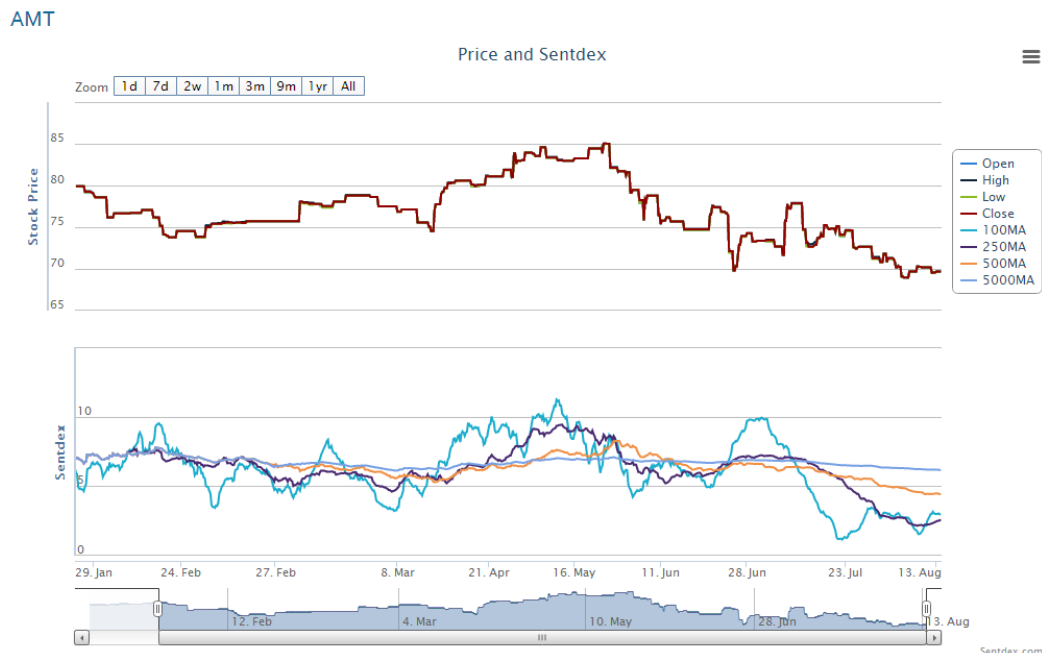


Figure 1: Sentdex sentiment index for American Tower Corporation (AMT)

Above is an example of how Sentdex chart the sentiment rating that they extract from sources such as Twitter and webpages. 'If you are a long investor, then I would look for stocks that continue increasing in sentiment over time. A good stock to show is American Tower Corporation (AMT). This stock has been in levels of a sentiment index of almost 11. Since then, however, it has been in a decline. The trend was clear near the end of May.' (Sentdex, 2014)

Sentdex appear to have a very refined model and framework for their sentiment analysis procedures. They have numerous charts like the example of AMT which strongly suggest that financial market sentiment can be used as a factor when trying to predict stock price movement.

There are also a number of papers that have been written on this particular topic. A paper written by (Chen & Lazer, 2011) investigate the relationship between Twitter feed content and stock market movement. They proposed a similar approach to the model I have opted with, using a "pre-generated word list of roughly five thousand common words along with log probabilities of 'happy' or 'sad' associated with the respective words." The process they follow is alike to the rating system I have adopted but instead they have accumulated the

log probabilities of each tokenized word. However, this method resulted in “highly uncorrelated data”. They then tried representing their sentiment rating value in an alternative way by counting the frequency of ‘happy’ tweets and representing this as a percentage of all tweets for that day. The conclusion of their paper reflects positively on the theory behind this topic as they state that, “even with much simpler sentiment analysis methods, a correlation between Twitter sentiment data and stock market movement can be seen.”

Another paper written by (Zhang, 2013) investigates how effective different machine learning techniques are on providing ranges of sentiment polarity on a tweet corpus. An extension to this investigation comes in the form of two additional tasks, looking for a “correlation between twitter sentiment and stock prices” and determining “which words in tweets correlate to changes in stock prices by doing a post analysis of price change and tweets.” To examine the correlation between twitter sentiment and stock prices they decided to “look for intra-day correlation with a lag period of k minutes.” Therefore, they gathered the data they were using (from the Twitter API and Yahoo API) on a minute by minute basis as they were only running their analysis over a one day period during operating hours of the stock exchange. Again, they implemented a rating system to obtain a sentiment score. To do this they took the ratio between positive and total sentiments. If there were no negative sentiments, the ratio would be 1. Conversely, if there was less positive sentiment, the ratio would be closer to 0.

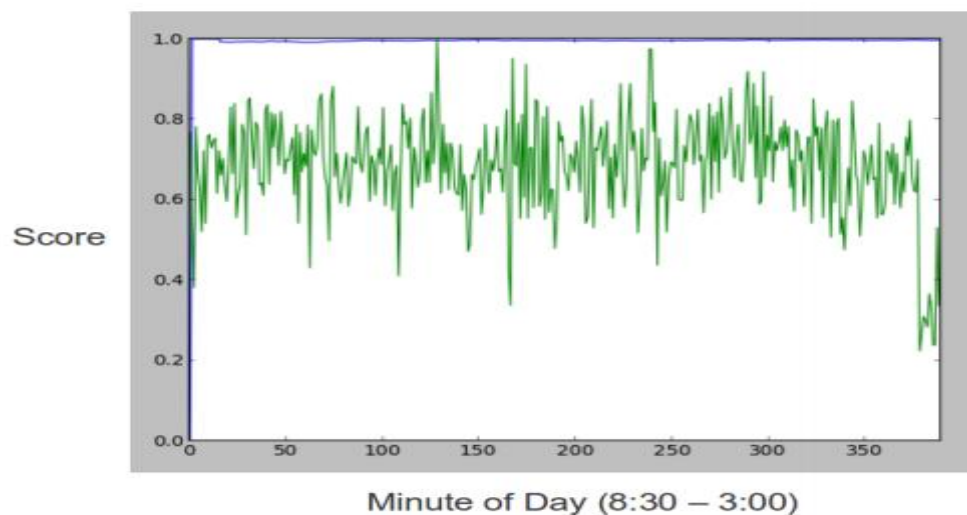


Figure 2: Uncorrelated stock prices and twitter sentiment

The results of this part of their analysis proved to be disappointing and they concluded that “there was almost no correlation between intraday tweets of any lag value”. They also suggested that twitter sentiment analysis “most likely requires a larger time granularity and more data to be truly effective”.

3 System

In this chapter I will begin by outlining the overall structure of the technical report, before presenting a detailed description of the requirements of the project. I will then introduce the design and architecture of the methods and how the different workings operate together to form the overall working environment. Lastly, I will look in detail at the technologies and methodologies employed in the implementation of the analysis and how they were tested during the project.

3.1 Structure

The paper is structured to best describe the process of the implementation of the analysis and explain how each step of the procedure was executed. The introduction and background explained briefly the purpose of the analysis, the reasons for the methods being chosen and an overview of previous analyses that have taken place. Some of the different software and APIs that are used, and why they were chosen, were discussed.

The remainder of this paper is structured as follows:

- The next section will contain the requirements of the project, the design and architecture of the methods, the technologies and methodologies employed, and the testing that took place throughout.
- The following section will contain the results of the analysis and will discuss the inferences that were made.
- The final section will be the conclusion of the analysis and will discuss the findings of the paper, the difficulties that occurred throughout, as well as further developments of research for the project.

The appendix of the paper contains the various deliverables that were previously submitted; the project proposal, requirements specification and project planning documentation. It also contains all the coding segments that were used.

3.2 Requirements Specification

3.2.1 Purpose

The aim of this analysis is to identify if a correlation can be made between changing stock prices and sentiment associated with that stock. The sentiment will be extracted from relevant texts and quantitatively modelled.

Trading stock prices can be affected by market conditions, politics, quantitative factors and other fundamental factors. This analysis aims to investigate whether these factors have an impact on the sentiment related to the stock, and how value may be potentially obtained from it.

The objective is to investigate if sentiment analysis can be considered as a viable tool in assisting with stock price prediction.

A structured collection of information which represents the requirements of this analysis will be documented with a description of its behaviours and a set of use cases that describe interactions.

3.2.2 Project Scope

3.2.2.1 Scope

There are a number of stages regarding the implementation of the analysis. Firstly the data must be acquired. This will be done by accessing the Twitter API and Yahoo API. The data will then be parsed to the required format. Processing the data will be the next step where an algorithm will be used to classify the nature of the text, defining its polarity (positive or negative), and the degree of this polarity. A rating for the sentiment will be calculated on a daily basis and graphed against the changing price of a chosen stock. Trends and comparisons can then be assessed from this output.

3.2.2.2 Project Objectives

Predicting stock market prices is the act of determining the future value of a company's stock traded on a financial exchange. There are many factors which influence the value of stock prices and with this analysis I am investigating whether sentiment analysis has an affect or can be used as a tool to aid the prediction of stock prices or trends.

3.2.2.3 Project Criteria

The most relevant data must be obtained and processed efficiently for the analysis to have a chance of yielding a credible output. A high standard of accuracy will be required from the algorithm used on the data when estimating its sentiment.

3.2.2.4 Project Supposition

Data that will be obtained through access of the Twitter API will be assumed credible in relation to the trading stock that is being analysed. As a strategy, sentiment analysis will be viewed as a plausible technique for this project. The accuracy of the sentiment analysis will be tested and refined to achieve the best results.

3.2.2.5 Project Restrictions

From the research conducted on this topic, the volume of data was highlighted as an issue regarding the quantity required for an effective output. Twitter sentiment analysis "most likely requires a larger time granularity and more data to be truly effective". (Zhang, 2013)

As with any analysis, a larger population of data will usually yield more accurate results. Data was retrieved on a daily basis over a seven day period. Ideally, having data over a three to six month period would be better for the analysis so that trends could be assessed further. Time and human resources are a restriction in this instance.

There are limits associated with the retrieval of data required for the analysis. Twitter also has certain restrictions with the use of their API but the volume of data, taken from twitter, which will be used for the analysis, will not exceed that limit and is negligible.

Tests that I have carried out already on quantities of data similar in size to those I plan on processing on a daily basis suggest that hardware and software limitations will be not be a significant issue.

3.2.2.6 Project Risk

The main risk I have identified with the analysis is the structuring of the quantitative model that is required for generating the sentiment rating. Putting a value on the degree of sentiment may prove to be complex and refined testing will be required.

Another risk that I have recognized is the accuracy of the algorithms that will be used for deciding on the polarity of the texts. Unless relatively high accuracies can be achieved, the analysis may be deemed redundant.

3.2.2.7 Contingency Plans

I have an initial approach defined that I will follow to generate a test output. I aim to achieve this output with enough time left so that I can go back and further refine and develop technical details of the analysis such as the algorithm choice for machine learning and the modelling of the sentiment rating.

3.2.2.8 Human Resource

I chose a seven day period where I accumulated a certain quantity of data through the twitter API daily.

3.2.3 Requirements

A requirements specification is a comprehensive description of the intended purpose and environment of a project under development. It fully describes what the project is intended to achieve and how it will perform.

Eliciting requirements is a key task as the requirements serve as the foundation for the solution to the projects needs. To examine and define the requirements a combination of complementary elicitation techniques is used. For this project brainstorming and document analysis were the two techniques used.

3.2.3.1 Functional Requirements

In this section I will outline the functional requirements of the project. A functional requirement is simply a statement of what the system must do and describes what the system can achieve.

- Access the Twitter API to obtain data regarding the stock
 - Create a Twitter account
 - Create a Twitter Development account
 - Receive Authentication codes for access of the API
 - Identify a relevant Twitter library to use for accessing the API (Eg: “tweepy”)
 - Stream tweets using a Python script with the Authentication codes and the Twitter library
 - Save tweets as CSV files while running the script during the opening hours of the stock exchange

- Access the Yahoo finance API to obtain stock trading data
 - Download and install the Python library “urllib2” to access URLs
 - A Python script using this library to access the Yahoo finance URL containing the appropriate parameters (stock, range, etc)
 - Parse the relevant columns and save the data as a CSV file

- Build a “Positive” and “Negative” word list for the model
 - Download, install and configure SQLite (Database)
 - A source of “Positive” and “Negative” words is required (www.Thesaurus.com)
 - A Python script to scrape the words and store them in the SQLite database with “+1” and “-1” attributed to the “Positive” and “Negative” words respectively

- An algorithm to calculate the sentiment rating of the data
 - Python script is required to access the SQLite database
 - For every tweet processed through the algorithm, each word is cross referenced against every word from the database and a sentiment rating is accumulated

- Plot graphs for the stock price changes and for the sentiment rating over the specified seven day period
 - Python scripts is required to plot these graphs, accessing the stored data for both
 - Graphs are required to be formatted clearly and appropriate for inference

3.3 Design and Architecture

In this section I will introduce the overview of the design and architecture of the project which includes the system architecture and the use case models for the project.

The architecture of the project provides a conceptual model of the analysis and clearly defines the relationships between the individual elements which make up the overall application.

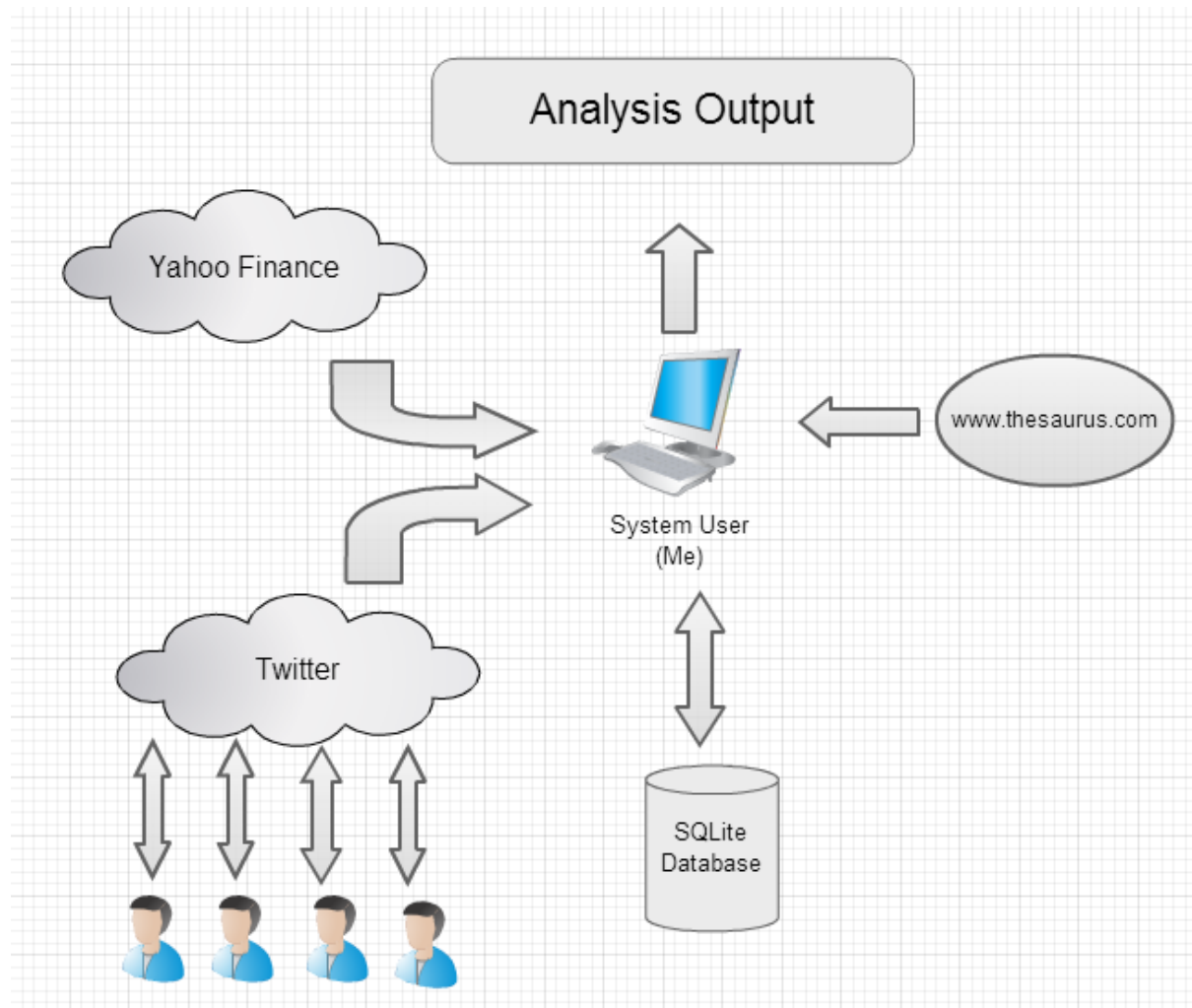


Figure 3: System Architecture

3.3.1 Use Case Models

3.3.1.1 Access Twitter API

3.3.1.1.1 Scope

Authentication codes must be acquired from Twitter by registering an application with them so that access can be made to their API.

3.3.1.1.2 Description

The user must register an account with twitter, and then register a twitter application before being granted the authentication codes.

3.3.1.1.3 Use Case Diagram

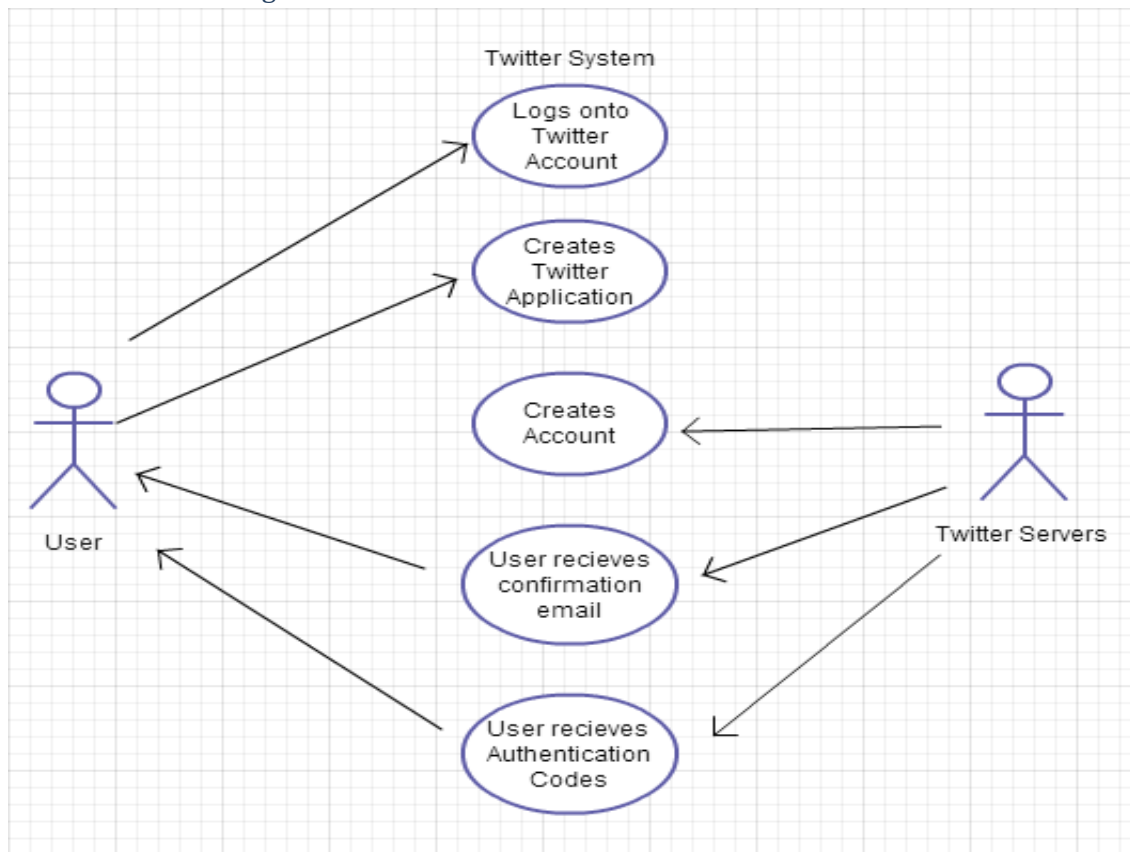


Figure 4: UML Twitter API

3.3.1.2 Access the Yahoo finance API

3.3.1.2.1 Scope

Trading stock data is to be sourced from the Yahoo finance API.

3.3.1.2.2 Description

The user must write a Python script to access the Yahoo finance URL with the data range and stock name parameters incorporated. The parsed data is returned to the user's system.

3.3.1.2.3 Use Case Diagram

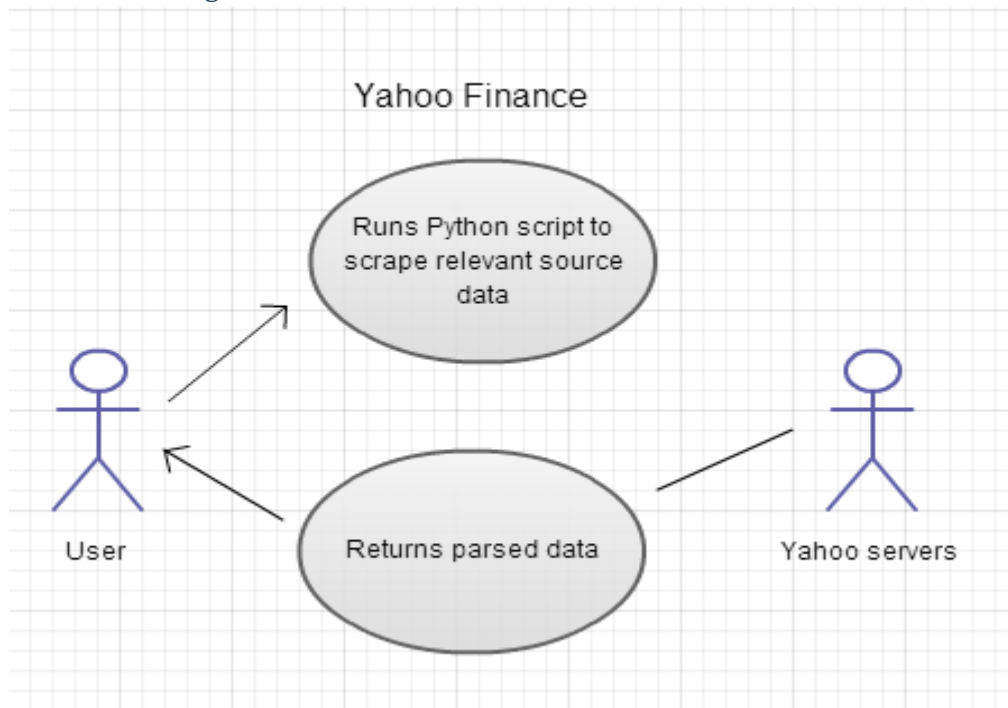


Figure 5: UML Yahoo API

3.3.1.3 Build a “Positive” and “Negative” word list

3.3.1.3.1 Scope

A source of “positive” and “negative” words is needed so that a sentiment rating can be accumulated for each tweet. The words will be stored in a relational database management system.

3.3.1.3.2 Description

Thesaurus.com will be used as the source of these words. A Python script will be run accessing the website and scraping every “positive” and “negative” word, attributing the words with a “+1” and “-1” respectively, and storing them in a SQLite database.

3.3.1.3.3 Use Case Diagram

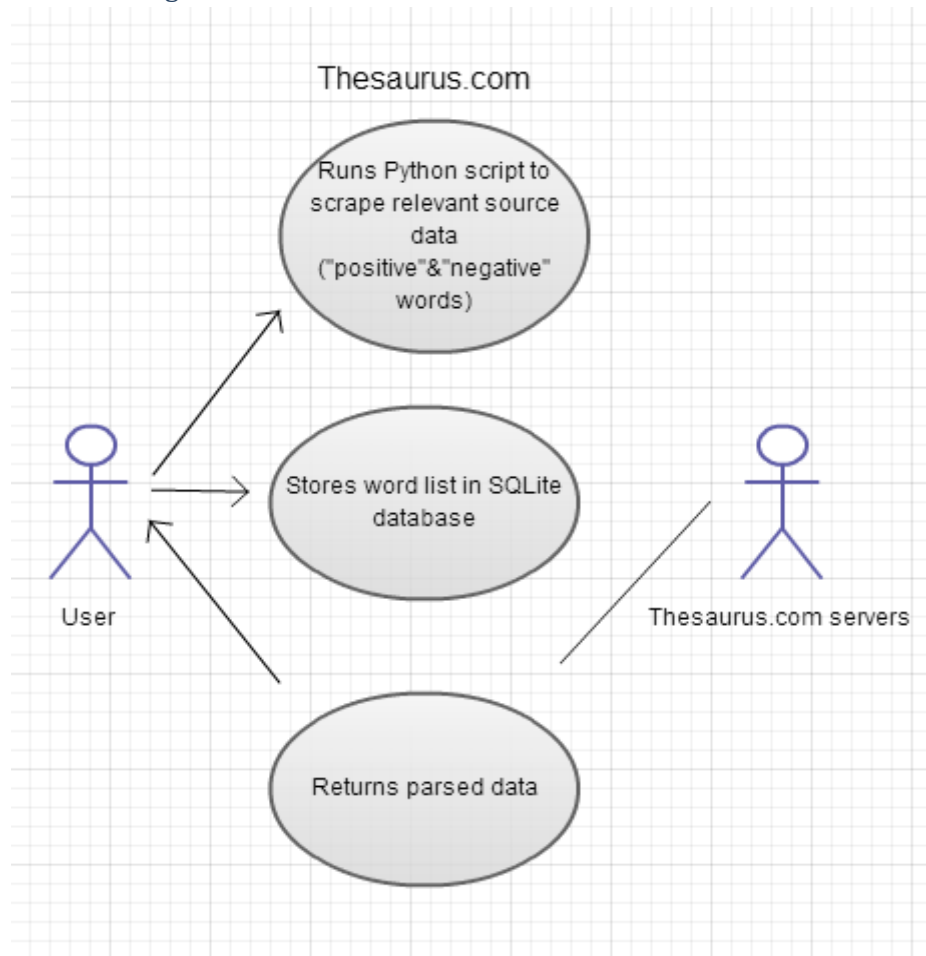


Figure 6: UML Word List

3.4 Implementation – Technologies and Methodologies

In this section I will discuss the implementation of the technologies and methodologies of the analysis. This will include the installation and configuring of the technologies used as well as the development of the methodologies employed.

The motivation for the various choices that were made will be explained in detail with an evaluation for the options that were available.

3.4.1 Technologies Overview

3.4.1.1 Python 2.7

A computer programming language was a fundamental necessity for the analysis. Python 2.7 was the language I used for the project. Python is the language I have most experience with and due to its practical functionality it was an obvious choice. It is a widely used, general purpose language that also supports multiple programming paradigms. There is also a large community base for open source software for Python with useful libraries available to download and install.

Python application is free to download from <https://www.python.org/downloads/>



Figure 7: Python 2.7 source

There is an option of two different versions of Python available to download, Python version 2 or Python version 3. The documentation on the website informed that version 2 was more compatible with various libraries and other open-source extensions than version 3. I was

aware that I would require various libraries for the analysis so Python version 2.7 was downloaded.

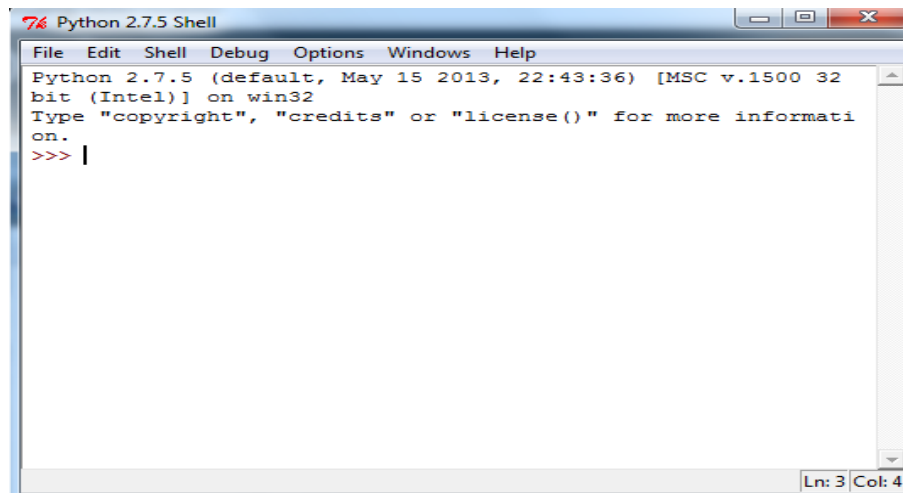


Figure 8: Python Shell

After downloading the version of Python I unzipped the contents of the folder and installed it. The application was then ready to run.

3.4.1.2 SQLite

A database system was required for the storage and easy access to the word list that would be used for the sentiment rating of the tweets. I decided to use SQLite for this function. It is a relational database management system but in contrast to other RDMS such as MySQL, it is accessible locally and is not a separate process that can only be accessed from the client application. SQLite has an application browser locally that is easy to use and its basic functionality was deemed appropriate for the tasks required of this analysis.

SQLite was free to download from <http://www.sqlite.org/download.html>

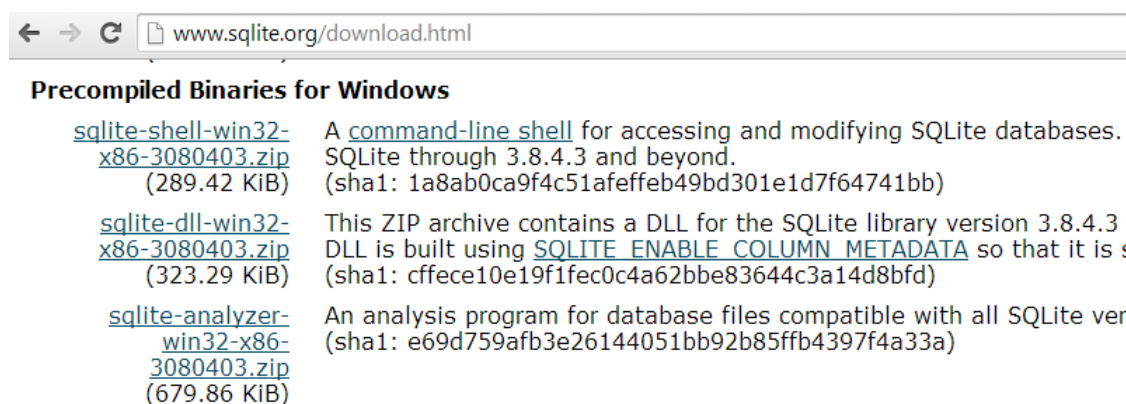


Figure 9: SQLite source

The first link in Figure 9, sqlite-shell-win32-x86-3080403.zip, was the appropriate version of SQLite to download. After downloading the file and extracting the contents, there was a SQLite browser application that could be run that accessed the browser directly.

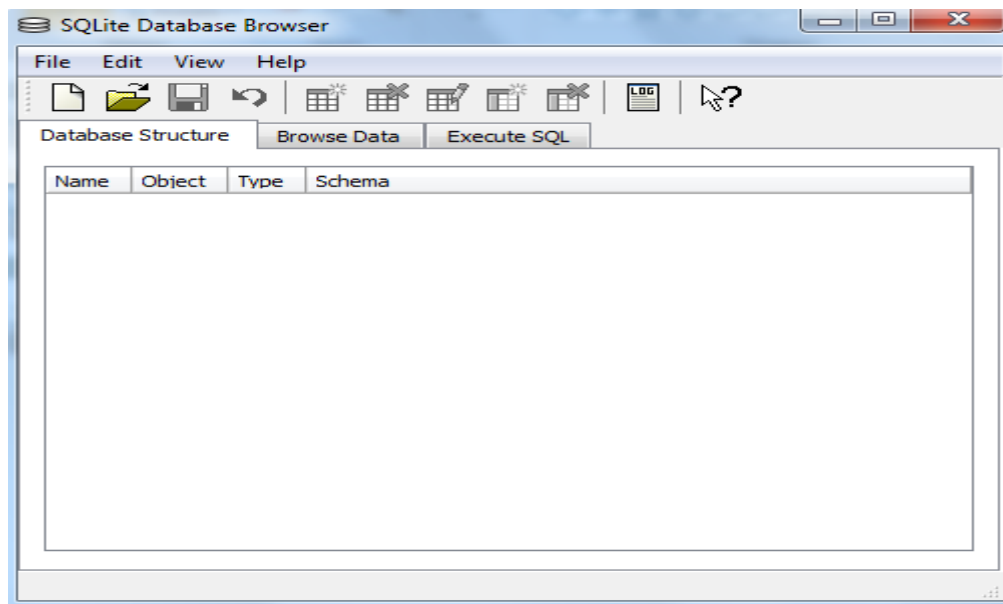


Figure 10: SQLite browser

3.4.2 Methodologies

3.4.2.1 Accessing data through the Twitter API

The main source of data for the analysis was from Twitter feeds. There were a number of steps required in the procedure to access this data.

Firstly a Twitter account must be set up – this was a standard procedure of filling out details and then logging onto your account using the log in details you chose in setting up the account.

The Twitter developer page would then be visited to set up a developer account which is a separate account for individuals wishing to use the Twitter API. <https://dev.twitter.com/>

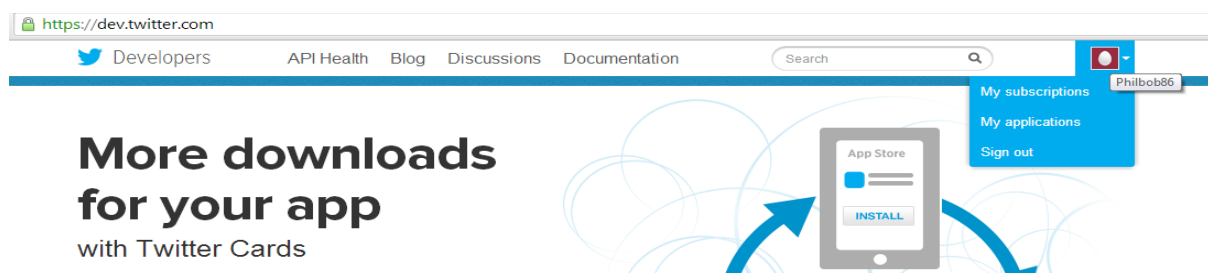


Figure 11: Twitter Developer page

A developer application is then required to be made where you provide details of the project you are carrying out with the use of the API.

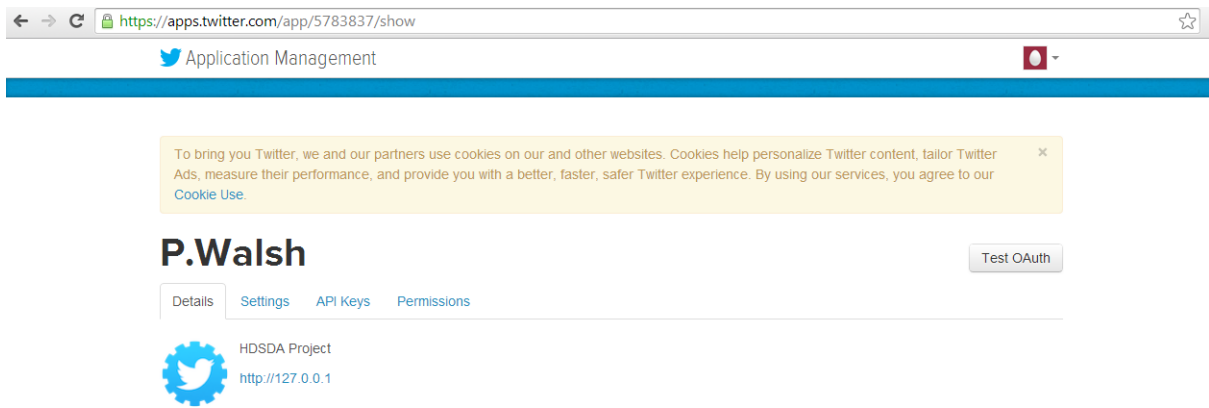


Figure 12: Developer Account

At this stage of the application you are then granted the four access keys. Twitter controls the access to their API through the use of a set of keys and tokens. The keys are used to identify who is consuming the API and to limit the utilisation of the API.

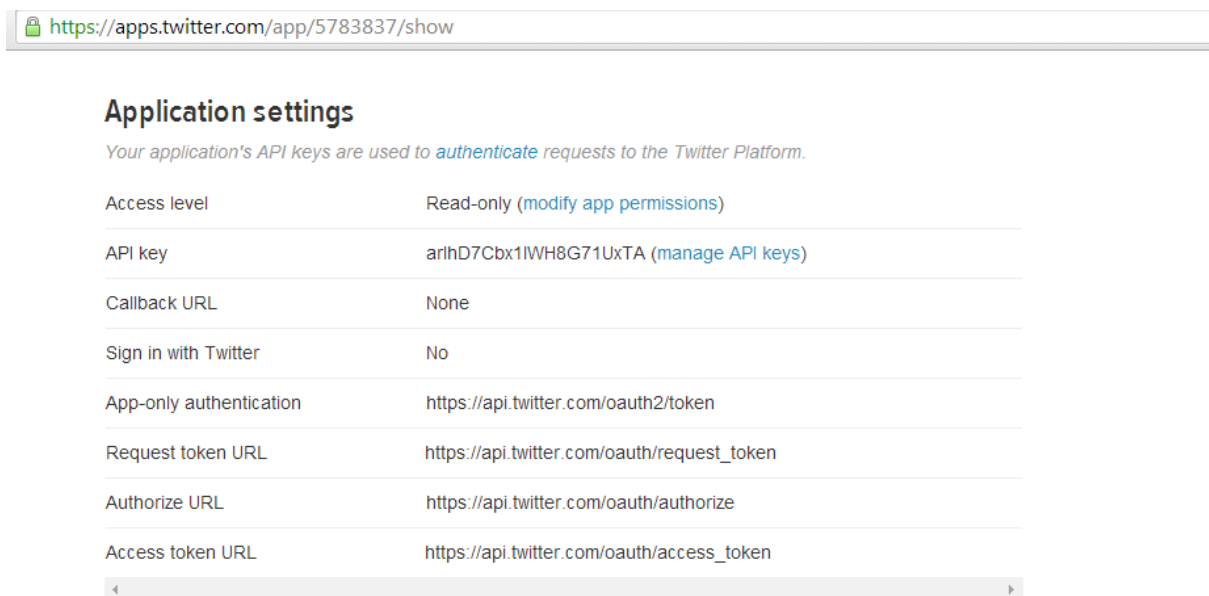


Figure 13: Auth Codes

The key information is either used to establish a session or is submitted with each request made to the API. These four codes will then be used in a Python script to access tweets using the search parameter.

Python has a very credible open source community with many different libraries available for various functions using the Python framework. Twitter compiled a list of libraries created by members of this community that could be used for access to the API.

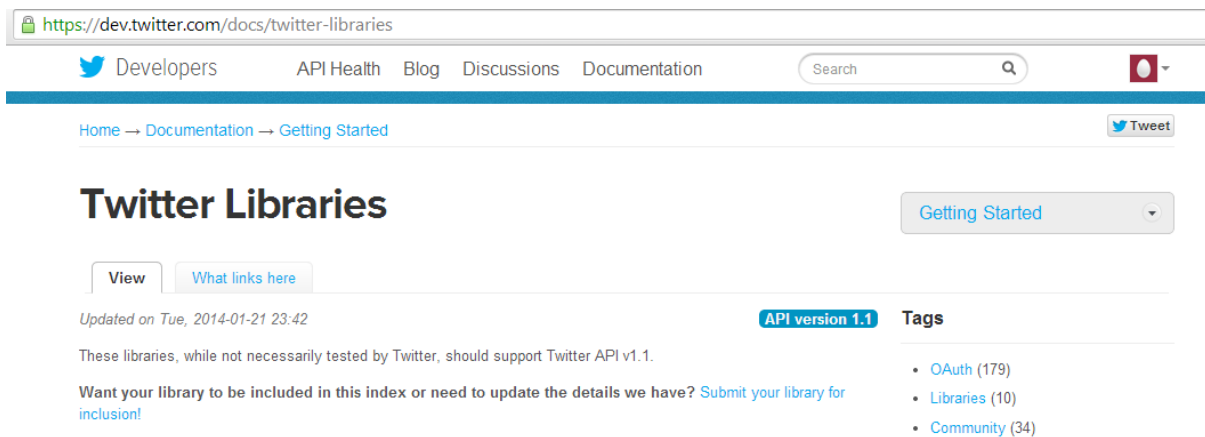


Figure 14: Twitter Libraries

There are a number of different libraries that are available in various computer programming languages. Twitter compiles a list of the libraries that are most used on host sites such as www.github.com.

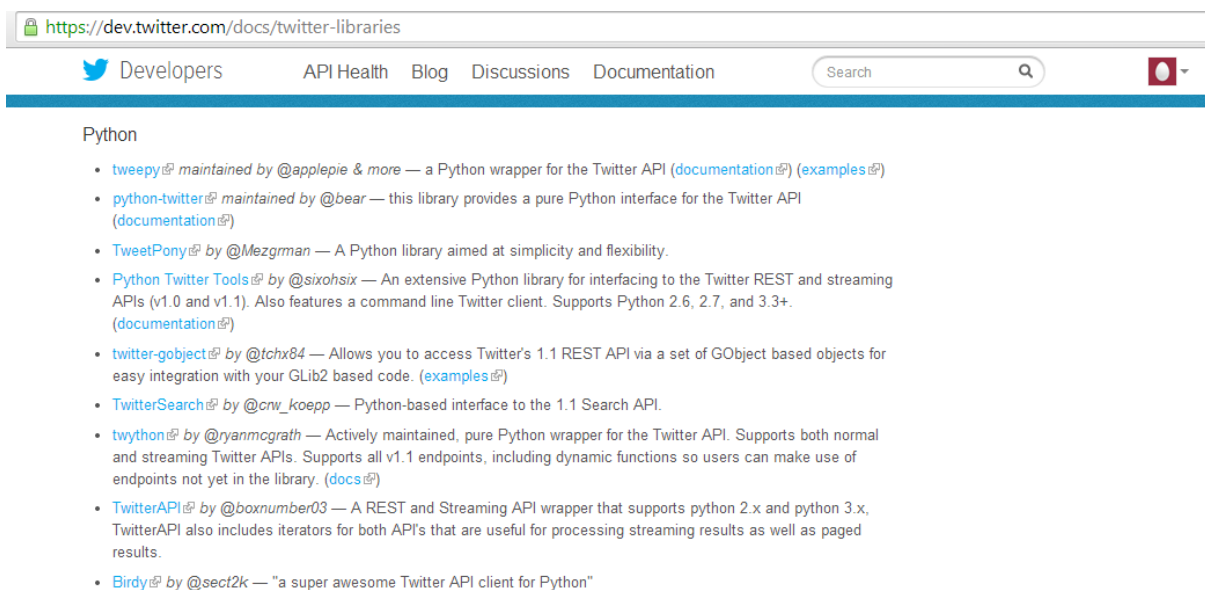


Figure 15: Python libraries

I decided to use “tweepy” as it was the most downloaded library and had a very explanatory documentation.

I followed a Sentdex tutorial and used the help of the “tweepy” documentation to code this script to access tweets and save the data to a CSV file. The search parameter at the bottom of the code is “AAPL” which is the stock name abbreviation for Apple.inc.

```

from tweepy import OAuthHandler
from tweepy.streaming import StreamListener
import time

consumer_key = 'arlhD7Cbx1lWH8G71UxTA'
consumer_secret = 'DvppaWwlvVz59V1Qt9y4ichg0S0c681ctuw2fzY1is'
token = '2332481052-Zp6FafDQDsqsWyIvSqWZ1nW5vS7VANDv7tWQq3N'
token_secret = 't0iIx9Amk15DrLFrHHOrJPrxpkcXPMYuzRGJNWB14ZX4'

class listener(StreamListener):

    def on_data(self, data):
        try:
            tweet = data.split(',"text":')[1].split(',"source')[0]
            print tweet

            saveThis = str(time.time())+'::'+tweet
            saveFile = open('tweetDB.csv','a')
            saveFile.write(tweet)
            saveFile.write('\n')
            saveFile.close()
            return True
        except BaseException, e:
            print 'failed on data,',str(e)
            time.sleep(5)

    def on_error(self, status):
        print status

auth = OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(token, token_secret)
twitterStream = Stream(auth, listener())
twitterStream.filter(track=["AAPL"])

```

Figure 16: Python code to access API (Sentdex, 2013)

```

Python 2.7.5 (default, May 15 2013, 22:43:36) [MSC v.1500 32 bit (Intel)] on win
32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
rumors of Rader resignation had to do with $AAPL $VHC case,someone also stated a
s a joke,he took a BIG gift ;)..
Could be breaktout time on $SNDK $DSW $SD $AAPL Must see http://t.co/1NiXOpK2
nH
How To Make Sure You Can Still Get Texts If You Ditch Your iPhone (AAPL) - Hope
fully, she disabled iMessage.Apple... http://t.co/PAQyRRrBGw
@jimcramer Agree. But you're smart enough to know $aapl rising due to upcoming I
phone6, iwatch, buybacks, nice Div @AngelaAhrendts, etc. No?

```

Figure 17: Print Tweets to console

3.4.2.2 Building a “Positive” and “Negative” word list for the model

A quantitative representation, or rating, is necessary so that a value could be attributed to the sentiment of the data. To do this a pre-defined words list that represent individual “positive” and “negative” words was compiled.

Rather than manually compile the list of words, www.thesaurus.com was identified as a suitable source of “positive” and “negative” words. The website offers a function of searching a word, say “good”, and it then displays all the synonyms for that word.



Figure 18: Synonyms for "good"

The method is to attribute each of these synonyms with a “positive” rating of “+1” as I am making the assumption that these words are positive words.

The same method is applied to the word “bad” which I assume is a “negative” word, as well as all the synonyms of the word bad, which are given a “-1” rating.

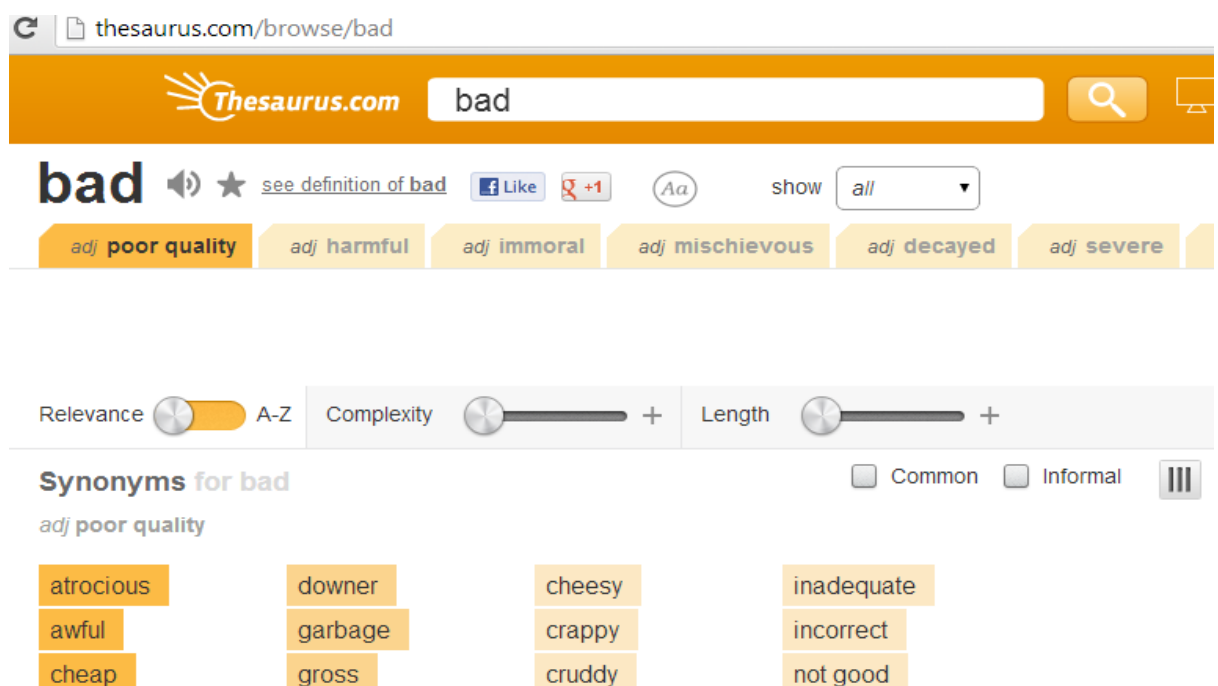


Figure 19: Synonyms for "bad"

To create the word list, a Python script was created that scraped the relevant words, and then populated the SQLite database with each word.

This procedure began with creating a table in the database. This was done by running the following Python script.

```
import sqlite3

conn = sqlite3.connect('knowledgeBase.db')
c = conn.cursor()

def createDB():
    c.execute("CREATE TABLE wordVals (word TEXT, value REAL)")

    c.execute("CREATE TABLE doneSyns (word TEXT, value REAL)")
```

Figure 20: Create database code (Sentdex, 2013)

The idea behind scraping the words from www.thesaurus.com was to retrieve the HTML page data and then parse, and extract the data. I did this with the help of a Sentdex tutorial.

```
import time
import urllib2
from urllib2 import urlopen
import re
import cookielib
from cookielib import CookieJar
import datetime
import sqlite3

cj = CookieJar()
opener = urllib2.build_opener(urllib2.HTTPCookieProcessor(cj))
opener.addheaders = [('User-agent', 'Mozilla/5.0')]
conn = sqlite3.connect('knowledgeBase.db')
c = conn.cursor()
startingWord = 'good'
startingWordVal = 1
synArray = []

def main():
    try:
        page = 'http://thesaurus.com/browse/'+startingWord+'?s=t'
        sourceCode = opener.open(page).read()
        try:
            synoNym = sourceCode.split('<a href="http://thesaurus.com/browse/great" class=common-word data-id="1"')
            x = 1
            while x < len(synoNym):
                try:
                    synoNymSplit = synoNym[x].split('<div class="synonyms-horizontal-divider"></div>')[0]
                    synoNyms = re.findall(r'"text">(\w*)</span>', synoNymSplit)
                    print synoNyms
                    for eachSyn in synoNyms:
                        query = "SELECT * FROM wordVals WHERE word =?"
                        c.execute(query, [(eachSyn)])
                        data = c.fetchone()

                        if data is None:
                            print 'not here yet, let us add it'
                            c.execute("INSERT INTO wordVals (word, value) VALUES (?,?)", (eachSyn, startingWordVal))
                            conn.commit()

                        else:
                            print 'word already here'

                except Exception, e:
                    print str(e)
                    print 'failed in 3rd try'

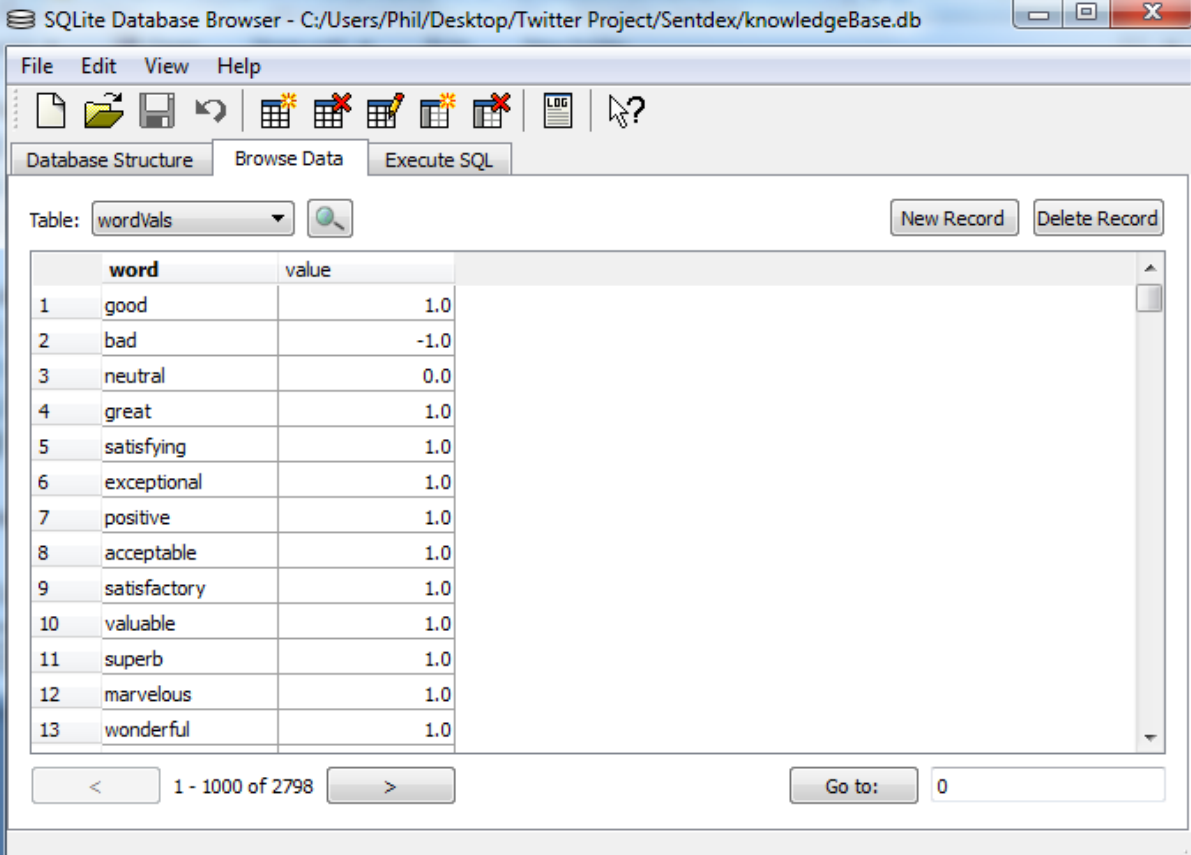
                x+=1

            except Exception, e:
                print str(e)
                print 'failed 2nd try'
        |
    except Exception, e:
        print str(e)
        print 'failed in the main loop'

main()
c.execute("INSERT INTO doneSyns (word, value) VALUES (?)", (startingWord))
conn.commit()
```

Figure 21: Scraping and populating database (Sentdex, 2013)

This script was run using the word 'good' and it populated the SQLite database with every synonym for "good" with a rating of "+1". Conversely, this was done for the word "bad" with a rating of "-1" for each word. For every synonym that appeared for "good" and "bad", every synonym of those words were also parsed and inserted into the database. This was done by manually changing the string value in the script for each synonym of "good" and "bad". This process resulted in roughly the same number of "positive" and "negative" words compiled in the database, with aim avoiding any bias for one set of words over the other.



SQLite Database Browser - C:/Users/Phil/Desktop/Twitter Project/Sentdex/knowledgeBase.db

File Edit View Help

Database Structure Browse Data Execute SQL

Table: wordVals

New Record Delete Record

	word	value
1	good	1.0
2	bad	-1.0
3	neutral	0.0
4	great	1.0
5	satisfying	1.0
6	exceptional	1.0
7	positive	1.0
8	acceptable	1.0
9	satisfactory	1.0
10	valuable	1.0
11	superb	1.0
12	marvelous	1.0
13	wonderful	1.0

< 1 - 1000 of 2798 >

Go to: 0

Figure 22: Word list in SQLite browser

When this process was completed the database had a list of "positive" and "negative" words, each with a rating of "+1" and "-1" respectively.

3.4.2.3 Algorithm to calculate the sentiment rating of the data

A scoring system is established for each of the tweets using the pre-defined word list. For each tweet that is processed through the algorithm, each word is cross referenced against every word that is in the database. Any word that is in a given tweet that does not appear in the word list is discarded. A rating is then accumulated of “positive” and “negative” words that appear in the tweet. An average rating can be calculated from the set of tweets and provides a sentiment value for each day.

```
import sqlite3
import time

conn=sqlite3.connect('knowledgeBase.db')
c=conn.cursor()

negativeWords = []
positiveWords = []

sql = "SELECT * FROM wordVals WHERE value =?"

def loadWordArrays():
    for negRow in c.execute(sql, [(-1)]):
        negativeWords.append(negRow[0])
    print 'neg words loaded'

    for posRow in c.execute(sql, [(1)]):
        positiveWords.append(posRow[0])
    print 'pos words loaded'

def testSentiment():
    readFile = open('AALP May 7.csv','r').read()

    sentCounter = 0

    for eachPosWord in positiveWords:
        if eachPosWord in readFile:
            sentCounter += 1

    for eachNegWord in negativeWords:
        if eachNegWord in readFile:
            sentCounter -=1

    if sentCounter > 0:
        print 'this text is pos'

    if sentCounter == 0:
        print 'this text is neutral'

    if sentCounter < 0:
        print 'this text is neg'

    print sentCounter

loadWordArrays()
testSentiment()
```

Figure 23: Sentiment Rating Algorithm (Sentdex, 2013)

The algorithm has two built in functions; one for loading the “positive” and “negative” words from the SQLite database and another for loading the file that is to be processed and accumulating a sentiment rating for that file.

The algorithm has two simple ‘for’ loops where each item in the input file is cross-referenced against every word in the word list. The first loop runs through all the words from the positive list and the second loop runs through the words from the negative list. In each loop a counter is set up. If an item in the input file matches a word in the word list, a “+1” or “-1” is added to the sentiment rating. The resultant value will be the overall sentiment rating for the input file.

3.4.2.4 Accessing stock trading data from Yahoo finance API

Trading data was sourced from the Yahoo finance API. The following URL outputs a HTML page with the relevant data that can be parsed.



Figure 24: Yahoo finance URL

The highlighted string value ‘stock’ is the parameter of the URL where the desired stock data you require can be inputted. Apple.Inc has been chosen as the stock being analysed. The abbreviation for Apple.Inc is “AAPL”. This is inputted into the URL as follow with “1m” (previous month’s data) being requested as the volume of data returned. The output is also in CSV format.

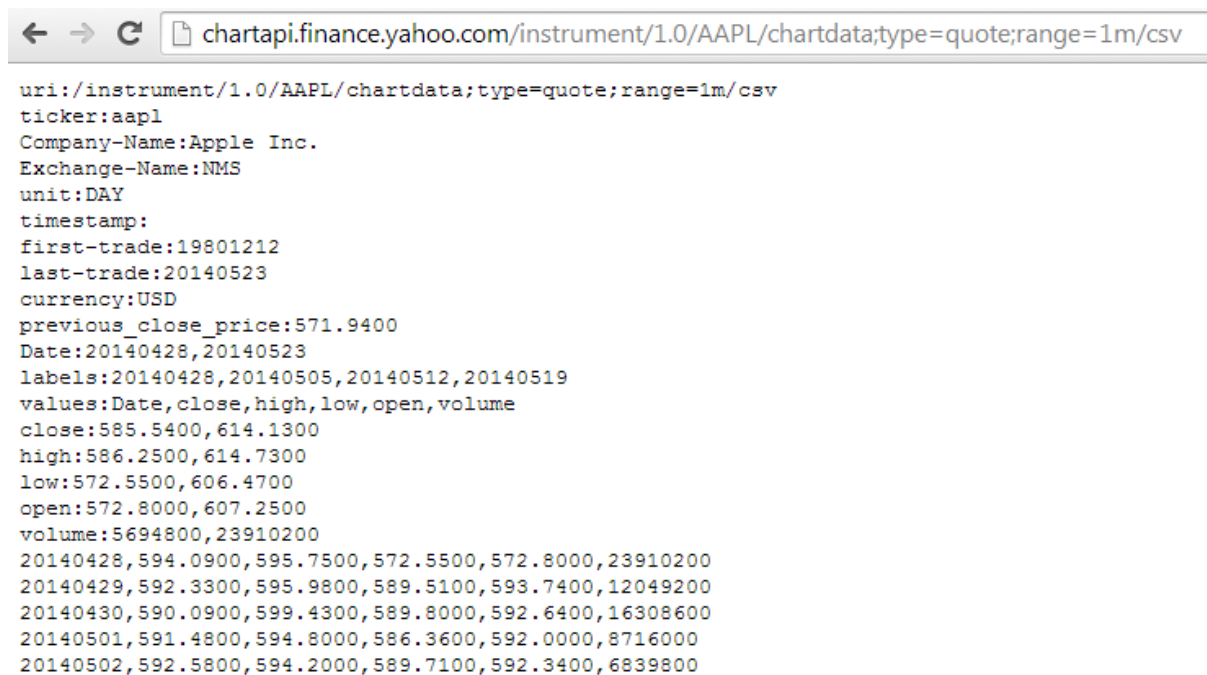


Figure 25: AAPL Trading data output

The HTML output from this URL is obtained using the following code. The relevant data from this output is parsed, retrieving the data, close, high, low, open and volume columns. These columns refer to the stock's price.

```
import urllib2
import time

stockToPull = 'AAPL'

def pullData(stock):
    try:
        fileLine = stock+'.txt'
        urlToVisit = 'http://chartapi.finance.yahoo.com/instrument/1.0/'
                        |+stock+'/chartdata?type=quote;range=1m/csv'
        sourceCode = urllib2.urlopen(urlToVisit).read()
        splitSource = sourceCode.split('\n')

        for eachLine in splitSource:
            splitLine = eachLine.split(',')
            if len(splitLine)==6:
                if 'values' not in eachLine:
                    saveFile = open(fileLine,'a')
                    lineToWrite = eachLine+'\n'
                    saveFile.write(lineToWrite)

        print 'Pulled',stock
        print 'sleeping'
        time.sleep(5)

    except Exception,e:
        print 'main loop',str(e)

pullData(stockToPull)
```

Figure 26: Script to retrieve trading data (Sentdex, 2013)

When this script was run it outputted a CSV file with the parsed trading data. Figure 27 is a subset of the data opened in Microsoft Excel. The column on the far left represents the date for each of the corresponding prices in each row.

8	20140505	600.96	601	590	590.14	10252400
9	20140506	594.41	604.41	594.41	601.8	13377300
10	20140507	592.33	597.29	587.73	595.25	10102300
11	20140508	587.99	594.41	586.4	588.25	8224900
12	20140509	585.54	586.25	580.33	584.54	10396500

Figure 27: Subset of trading data

3.5 Testing

In this section I will discuss the various testing procedures that were carried out throughout the project. Sourcing the data from the Twitter API was a sort of ‘trial and error’ situation with regard to the particular stock I would use for the analysis. Initially when I pulled the tweets down and printed them to the Python console, there was a lot of unwanted data as well as the main text of the tweet being streamed. This required parsing the tweets so that the excess data would be excluded.

I began running test by filling in the search parameter of the script with different well known stock names and examining the data returned. I had begun doing this in the early morning, about 9:30am IST (Irish Standard time). The stocks I was interested in were on the NYSE (New York Stock Exchange) as I expected there to be more interest and activity surrounding these stocks. However, there was very little response and there weren’t many tweets being streamed to the console as New York is five hours behind and the NYSE was closed. I ran the script again at 3p.m IST and there was a steady flow of tweets. This prompted me to decide that I should run stream the tweets between the hours the NYSE was open. I ran the script between the hours of 9.30am and 4pm EDT (Eastern Daylight Time) and I was returning about 2000 tweets which I deemed as suitable sample size of data for the analysis.

I chose Apple.Inc for the analysis. I found that there was a lot of activity on Twitter with regard to the company’s share price in comparison to others. I used the abbreviated “AAPL”, which is Apple’s stock representation, as the search parameter in the script. If I was to search “Apple” in the script I would get a huge volume of tweets that would mostly be of no real use for the analysis.

I accessed the Twitter API for data between the hours the NYSE was open for trading for five days starting on the May 5th. The NYSE is closed at the weekend so I ran the script from Monday through to Friday. Since I would be streaming data for a number of hours over the five day period I felt that this was the optimum volume of data I should retrieve. Ideally data over monthly period would have been made for a better analysis.

The “positive” and “negative” word list that was compiled required manual changes to some of the words that were inserted into the SQLite database. For example, in Figure 18 it can be seen that a synonym for the word “good” is “bad”. I can’t explain why that is the case but in this instance the word “bad” would have been inserted into the database with a value of “+1” since all the synonyms for “good” were inserted into the database with that value. This prompted me to scan through the browser manually changing the values of some of the words where I used my own judgement as what words constituted being positive or negative. The advantage of using SQLite and having a browser where I could manually make these changes justified the choice of database.

4 Results

The graph below shows the change in stock price for Apple, Inc over the course of thirty days from April 24th to May 23rd. The opening and closing price of the stock for each day are represented by the blue and red line respectively. The highest and lowest prices that the stock is recorded at for each day are represented by the purple and green lines respectively.

AAPL

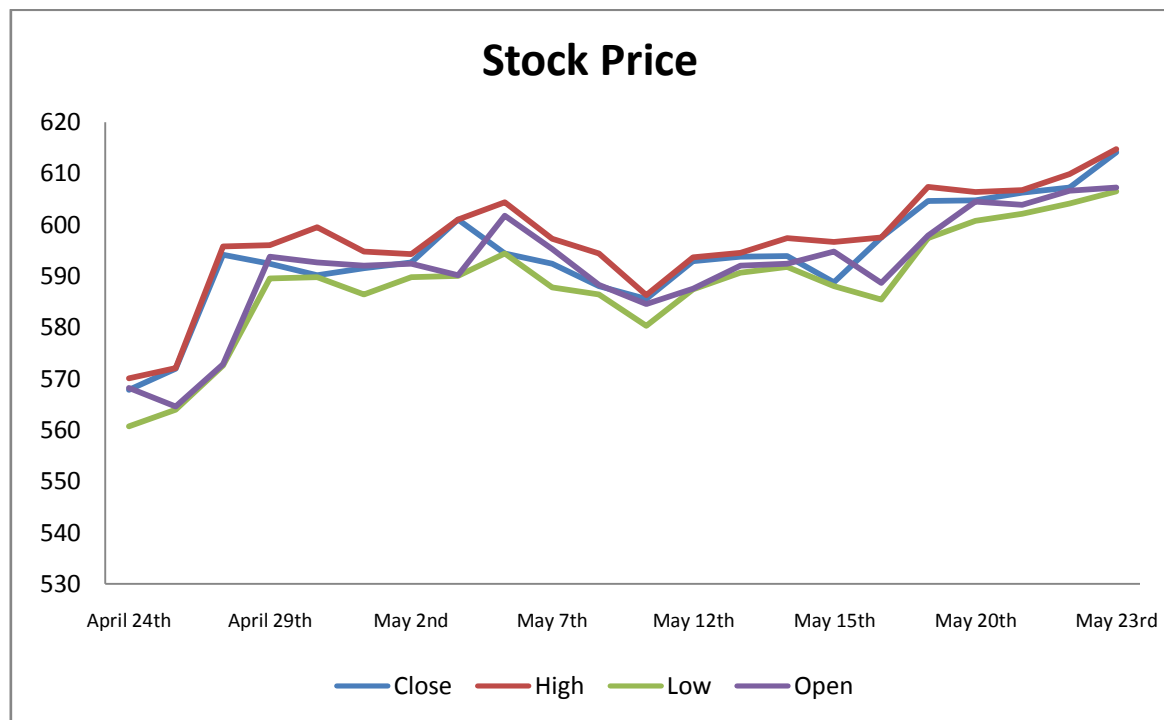


Figure 28: AAPL stock price movement

The trends that are of most interest for the analysis are the opening and closing prices with more focus on the stock price when the market closes at the end of each day. The assumption being made is that the sentiment rating that is calculated for each day corresponds to the closing price of the stock on that day. This assumption is being made with respect to the trend lines of both graphs.

The graph below in Figure 29 is a subset of the graph in Figure 28, displaying the stock price changes over a five day period beginning at May 5th through to May 9th. This is the range of days under examination for the analysis. The closing day price line is in bold to highlight its trend.

AAPL

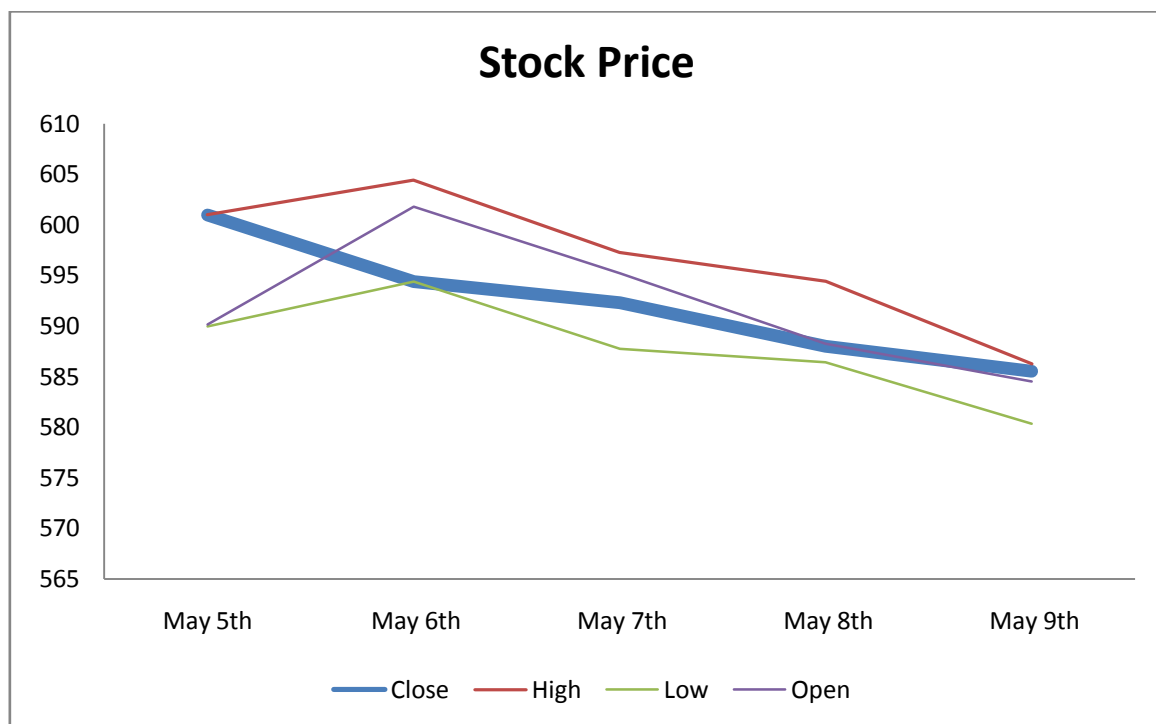


Figure 29: AAPL stock price movement

The graph in Figure 30 shows the trend line of the sentiment rating over the same five day period from May 5th through to May 9th.

AAPL

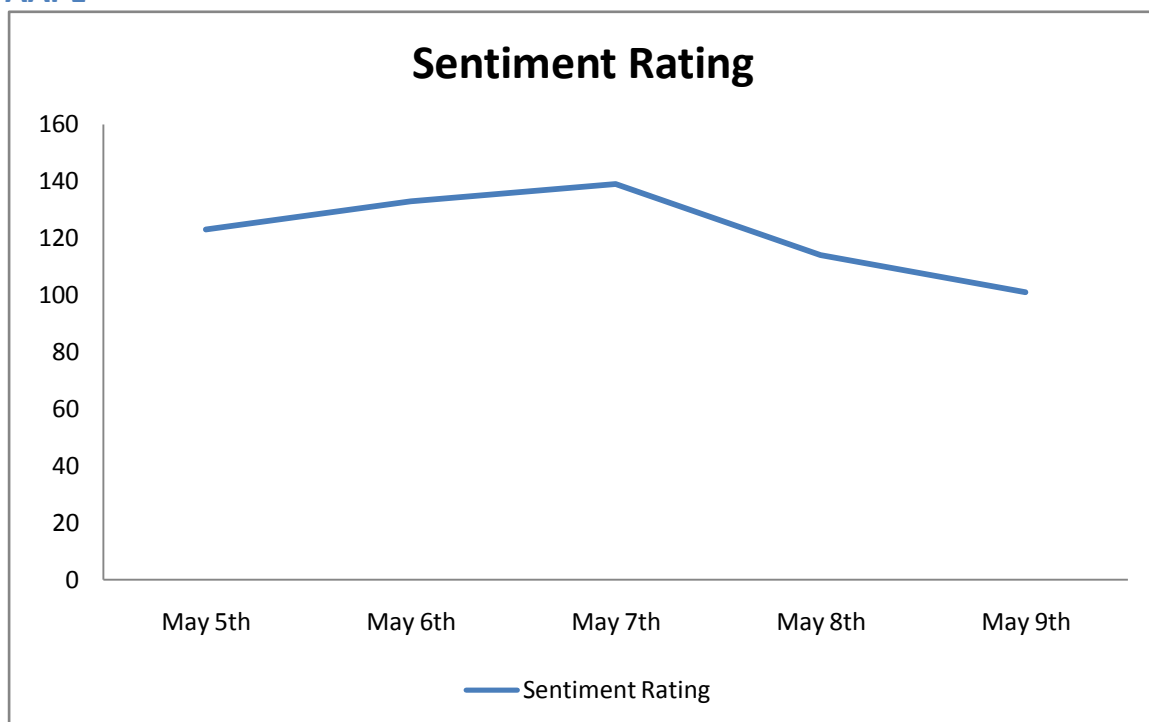


Figure 30: Change in sentiment rating

The trend of the closing day line can be seen to slowly move at a downward angle with a slightly decreasing stock price in Figure 29. The trend of the sentiment rating in Figure 30 increases slightly initially before changing direction and decreasing with a similar slope as the trend line in Figure 29.

There is a positive correlation between the two sets of data and the following equation was used to calculate the Pearson's correlation coefficient.

$$\rho_{X,Y} = \text{corr}(X, Y) = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y},$$

A value of 0.58 was calculated as the correlation coefficient for the two samples. This value infers that there is a positive correlation and that there is a reasonably strong relationship between the change in stock price and the sentiment rating for the examined time frame.

5 Conclusions

In this report I have examined the relationship between Twitter feed content and stock market movement with respect to AAPL (Apple.Inc). The specific aim of the paper was to identify if the quantitative sentiment information extracted from the Twitter feeds had a correlation with the stock price changes of Apple over a specified period of time. An extension of this analysis would be to test whether this method could be used to predict future shifts in prices by analysing the trends in the data.

I researched various methods from other work that has been done on this topic and investigated the technological options that were available to me for the project. I then looked in detail at the project as a whole, defining the requirements and outlining the design and architecture. I presented the methodologies of the analysis and the technologies used in its implementation. The testing that was carried out throughout the development of the project was also discussed in detail.

The results of the analysis are encouraging with regard to the inference that was made that a relationship existed. In Figure 28 it can be seen that after the five day period that was used as a subset for the analysis, the stock price begins to rise again. This raises the question of how this method would perform when attempting to predict future trends. It suggests that a much larger sample size would be required for predicting future market movement so that trends and patterns could be examined more thoroughly.

5.1 Further Development of research

The method that I adopted to extract the sentiment from the data is by no means robust or dynamic and it most definitely requires more testing and analysis. There are many more areas in reference to this topic where further ideas or methods could be used to contribute to a better output of results. With more time and resources there is a lot of potential for improvements in this area. One of the drawbacks of the analysis was the short range of time being examined, and a larger volume of data is necessary for a more comprehensive return.

Another option that the project could expand on could be the use of machine learning for text classification. With this approach the rating system would not be used with the “positive” and “negative” word list, but instead the percentage of positive tweets retrieved from the day as a whole would be used as the sentiment rating for each day. A classification algorithm would classify each tweet as being either positive or negative using a model it created from a set of training data. Once again, with a larger volume of data this could yield good results. However, this depends on how many tweets are made about the stock you wish to analyse.

As far as using sentiment analysis goes, an extension on the sources of data you could use is also something that can be considered. Other than just using Twitter, data could be sourced

by scraping different websites that are relevant to the stock being analysed. This would require a more refined model.

In conclusion I believe this project proved that a relationship exists between movements in financial markets and public sentiment, and that this proof can be potentially used a tool to make stock market predictions.

Appendix 1: Project Proposal

Objectives and Contribution to the Knowledge

Stock market prediction is the act of trying to determine the future value of a company stock or other financial instrument traded on a financial exchange. Some suggest that stock prices are unpredictable to predict while others disagree, pointing to multiple methods which supposedly allow them to gain future price information.

There are many factors that affect stock prices such as:

- Overall market conditions
- Politics (examples: new legislation, politician's election)
- Quantitative factors
- Other fundamental factors (examples: company changes, business models, earning reports)

A factor that could also be considered, and investigated, is sentiment analysis. Many times, politics, quant analysis, fundamental analysis, and market conditions can also pour into sentiment. Therefore, value can potentially be obtained from such analysis.

The initial objective of this project is based around financial market sentiment but it may also turn to or involve some of the following topics:

- Political Sentiment (examples: sentiment on elections)
- Business/Product Sentiment (general public sentiment of a particular business or product)
- Geographical Sentiment (this can be used in conjunction with the previous two topics)

The objective here is to identify if sentiment analysis can be used as a viable tool to assist stock price prediction. The sentiment generated from the analysis will be compared with prices to see if there is a significant correlation.

Background

At the most basic level, sentiment analysis is an attempt to identify the overall mood, feeling and speculation of a text. The rise of social media such as blogs and social networks has fuelled interest in this type of analysis.

Sentiment analysis involves building a system to collect and examine opinions about whatever topic has been discussed in such blog posts, comments, reviews or tweets. There are different ways of analyzing a body of text for sentiment or opinion. Two of the main methods used are:

- “Bag of words” model (This model focuses completely on the words, or sometimes a string of words, but usually pays no attention to the “context” so-to-speak)
- Natural Language Processing (This model attempts to have the machine actually understand the sentences structures, context, and is more focused on the succession of a string of words)

There are several challenges associated with this type of analysis. The first is that a word that is considered to be positive in one situation may be considered negative in another situation. Another challenge is that people don't always express opinions the same way. Also, people can be contradictory in their statements. Sarcasm can also be an issue.

A sophisticated, highly machine-trained system that has been rigorously tested would aim to achieve accurate results and be capable of minimizing the affects of these challenges.

Technical Approach

I plan on dividing up the tasks involved and following a systematic approach with regard to the implementation and execution of the project.

The initial aim will be to identify the requirements, and structure how the project will be completed. The main objectives of the analysis, which can be identified to begin with, are to source relevant data, analyse that data and output significant meaning from the analysis.

With these initial objectives I can begin to research possible methods or tools I can use to complete the tasks. For example:

- Research methods available to retrieve tweets or to scrape text from web pages
- Research suitable algorithms that could be used on the data
- Research how I can model the analysis to yield a relevant output

From there, a deeper understanding of the requirements will be acquired with the structure and implementation tasks of the project refined.

Special resources required

At this stage of the project I have identified that I will exclusively use Python. This programming language has a wide range of built in tools and open source material available, and from my initial investigation I have decided to choose it. As the project commences, other software tools may be acquired and used.

I may use MySQL to store the data in a database. As these are the preliminary stages of the project I am uncertain as to how I will deal with the data.

Obviously I will be using twitter and various web sites for the analysis, where I will have to adhere to the rules they have regarding the use of their data.

Project Plan

Below is a Gantt chart for the project with the general tasks and deliverables detailed on a timeline. As the project progresses and more detailed tasks become clearer, the gantt chart will be updated accordingly. The research task will be broken down into a group of tasks.

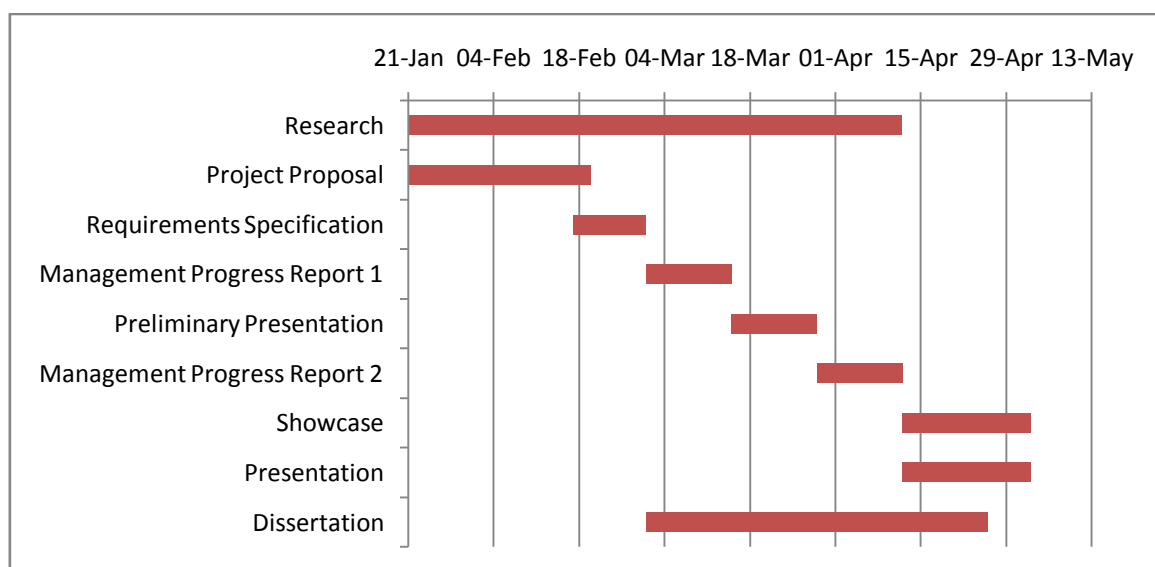


Figure 31: Project Gantt chart

Technical Details

As mentioned above, Python and its built in libraries will be used for the project. Any open source libraries or packages that I identify for use through Python will be referenced. I will implement my own programs from first principles where applicable or where time allows.

From my preliminary research, there are many libraries available for Python that access the Twitter API and Python wrappers for the Twitter API. They are very detailed and from first

impressions I will not require a lot of the features they offer which indicates to me that I will design and code my own scripts to access the tweets I wish to analyse.

To use access twitter in this way you must register with them so you can receive the four codes needed for authentication.

An example of one of these is “tweepy”. It is a Python wrapper for the Twitter API. Here is some code for a simple example where I can access tweets relating to a certain keyword.

```
from TwitterAPI import TwitterAPI

consumer_key = ' '
consumer_secret = ' '
access_token_key = ' '
access_token_secret = ' '

api = TwitterAPI(consumer_key, consumer_secret, access_token_key,
access_token_secret)

r = api.request('search/tweets', {'q':'Ryanair'})

for item in r:
    print(item['text'] if 'text' in item else item)
```

“Tweepy” has been downloaded and installed prior to running this code. Ryanair has been used for this example, when the code is run any tweets with the string word “Ryanair” are displayed in the Python GUI (graphical user interface). The tweets have been parsed already with this function and you can also limit the number of tweets you wish to return. Geographical coordinates can also be used as a parameter to restrict where the tweets are coming from.

Scrapping text from web pages is also a source of data. The code used for this is relatively straight forward, as far as I have researched at the moment, and an example of this is shown here:

```

import urllib2
from urllib2 import urlopen

website = 'http://www.ibtimes.com/apple-ibwatch-release-date-coming-200-employees-reportedly-working-project-device-could-have-simpler'

topSplit = "<div class='article-node-title'>"
bottomSplit = "</div></div></div> </div>"

def main():
    x=1
    sourceCode = urllib2.urlopen(website).read().lower()
    sourceSplit = sourceCode.split(topSplit)[1].split(bottomSplit)[0]
    content = sourceSplit.split('\n')

    for line in content:

```

Python has a built in library, natural language toolkit (NLTK), which can be used for sentiment analysis. As discussed previously, there are two main methods used for analyzing a body of text, which are the “bag of words” model and NLP.

The bag of words model usually has a large list, like a sort of “dictionary,” which is considered to be words that carry sentiment. These words each have their own “value” when found in text. The values are typically all added up and the result is a sentiment valuation. The equation to add and derive a number can vary, but this model mainly focuses on the words, and makes no attempt to actually understand language fundamentals.

The NLP model attempts to have the machine actually understand the sentences structures, context, and is more focused on the succession of a string of words. Usually, this structure requires the machine to have understanding of grammar principles. To do this, Natural Language Processing (NLP) techniques are used to tag parts of speech, named entities, and more, in order to actually understand the “language” of the text, and not just look for target words.

A combination of these is likely to be used for the analysis.

Systems/Datasets

I have outlined the systematic approach I plan to take with this project previously, the basic outline of the tasks being, sourcing the data, analyzing the data and outputting relevant results from the analysis.

The source data will be from tweets accessed from the Twitter API which will be returned within a set of parameters suitable to the particular analysis in question.

Web pages and blogs relevant to the analysis will also be identified and used as a source of data. The applicable text from these sources will be scraped and parsed accordingly.

Evaluation, Tests and Analysis

As I have mentioned previously I have only the outline tasks and objectives of the project identified at this moment. However, I anticipate testing the code, for accessing the tweets, for errors initially. Tests and evaluation will also be carried out at this stage on parsing the data returned and compiling the data in the desired format.

Scraping data from web pages will require detailed testing and evaluation as different websites have differing layouts which may require the coding to be altered. This will be carried out by initially calling the URL in question and printing the contents of it to the Python GUI. From here I can identify where the text is in the output and parse the data accordingly. This approach is what I have researched thus far and I assume I will find a more efficient way of processing for this task.

Running algorithms on the data will require thorough testing which I plan on conducting on a smaller sample of tweets and texts to begin with. I will also familiarize myself with the financial markets and run test analyses with mock results to see how I will correlate the results I will get from the model I will create and the actual stock price changes.

Since the project is at the preliminary stage, I anticipate that more tests and evaluation procedures will become more evident as it progresses.

Consultation with Specialisation Person(s)

Dr. Ioana Ghergulescu: Acknowledged my proposal and signalled for me to continue with my research.

Michael Bradford: Acknowledged my proposal and signalled for me to continue with my research. He also suggested researching NumPy, which is an extension to Python, to help with my project.

Appendix 2: Requirements Specification

Purpose

The aim of this analysis is to identify if a quantitative correlation can be made between changing stock prices and sentiment associated with that stock. The sentiment will be deduced from relevant texts and quantitatively modelled.

Trading stock prices can be affected by market conditions, politics, quantitative factors and other fundamental factors. This analysis aims to investigate whether these factors have an impact on the sentiment related to the stock, and how value may be potentially obtained from it.

The objective is to investigate if sentiment analysis can be considered as a viable tool in assisting with stock price prediction.

A structured collection of information which represents the requirements of this analysis will be documented with a description of its behaviours and a set of use cases that describe interactions.

Project Scope

Scope

There are a number of stages regarding the implementation of the analysis. Firstly the data must be acquired. This will be done by accessing twitters API (application programming interface) and scraping relevant texts from web pages, forums and journals. The data will then be parsed to the required format. Processing the data will be the next step where an algorithm will be used to classify the nature of the text, defining its polarity (positive or negative), and the degree of this polarity. A qualitative index will be calculated on a daily basis and graphed against the changing price of a chosen stock. Trends and comparisons can then be assessed from this output.

Project Objectives

Predicting stock market prices is the act of determining the future value of a company's stock traded on a financial exchange. There are many factors which influence the value of stock prices and with this analysis I am investigating whether sentiment analysis has an affect or can be used as a tool to aid the prediction of stock prices or trends.

Project Criteria

The most relevant data must be obtained and processed efficiently for the analysis to have a chance of yielding a credible output. A high standard of accuracy will be required from the algorithm used on the data when estimating its sentiment.

Project Supposition

Data that will be scraped from the web and obtained through access of the twitter API will be assumed credible in relation to the trading stock that is being analysed. As a strategy, sentiment analysis will be viewed as a plausible technique for this project. The accuracy of the sentiment analysis will be tested and refined to achieve the best results.

Project Restrictions

There are limits associated with the retrieval of data required for the analysis. Websites have differing stances regarding data protection and it won't always be possible to scrape websites and use the data legally. Twitter also has certain restrictions with the use of their API but the volume of data, taken from twitter, which will be used for the analysis, will not exceed that limit and is negligible.

As with any analysis, a larger population of data will usually yield more accurate results. At this moment I plan on retrieving data on a consecutive daily basis over a thirty period. Ideally, having data over a three to six month period would be better for the analysis so that trends could be assessed further. Time and human resources are a restriction in this instance.

Tests that I have carried out already on quantities of data similar in size to those I plan on processing on a daily basis suggest that hardware and software limitations will be not be a significant issue.

Project Risk

The main risk I have identified with the analysis is the structuring of the quantitative model that is required for the sentiment. Putting a value on the degree of sentiment may prove to be complex and refined testing will be required.

Another risk that I have recognized is the accuracy of the algorithms that will be used for deciding on the polarity of the texts. Unless relatively high accuracies can be achieved, the analysis may be deemed redundant.

Contingency Plans

I have an initial approach defined that I will follow to generate a test output. I aim to achieve this output with enough time left so that I can go back and further refine and

develop technical details of the analysis such as the algorithm choice for machine learning and the modelling of the sentiment index.

Human Resource

I will choose a thirty day period where I will accumulate a certain quantity of data through the twitter API each day. As well as this, I will be scraping data from different websites that have been identified as having data credible for the analysis.

Requirements

A requirements specification is a comprehensive description of the intended purpose and environment of a project (an analysis in this instance) under development. It fully describes what the project is intended to achieve and how it will perform.

Eliciting requirements is a key task as the requirements serve as the foundation for the solution to the projects needs. To examine and define the requirements a combination of complementary elicitation techniques is used. For this project brainstorming and document analysis were the two techniques used.

Functional Requirements

Access Twitter API

Scope

Authentication codes must be acquired from Twitter by registering an application with them so that access can be made to their API.

Description

The user must register an account with twitter, and then register a twitter application before being granted the authentication codes.

Use Case Diagram

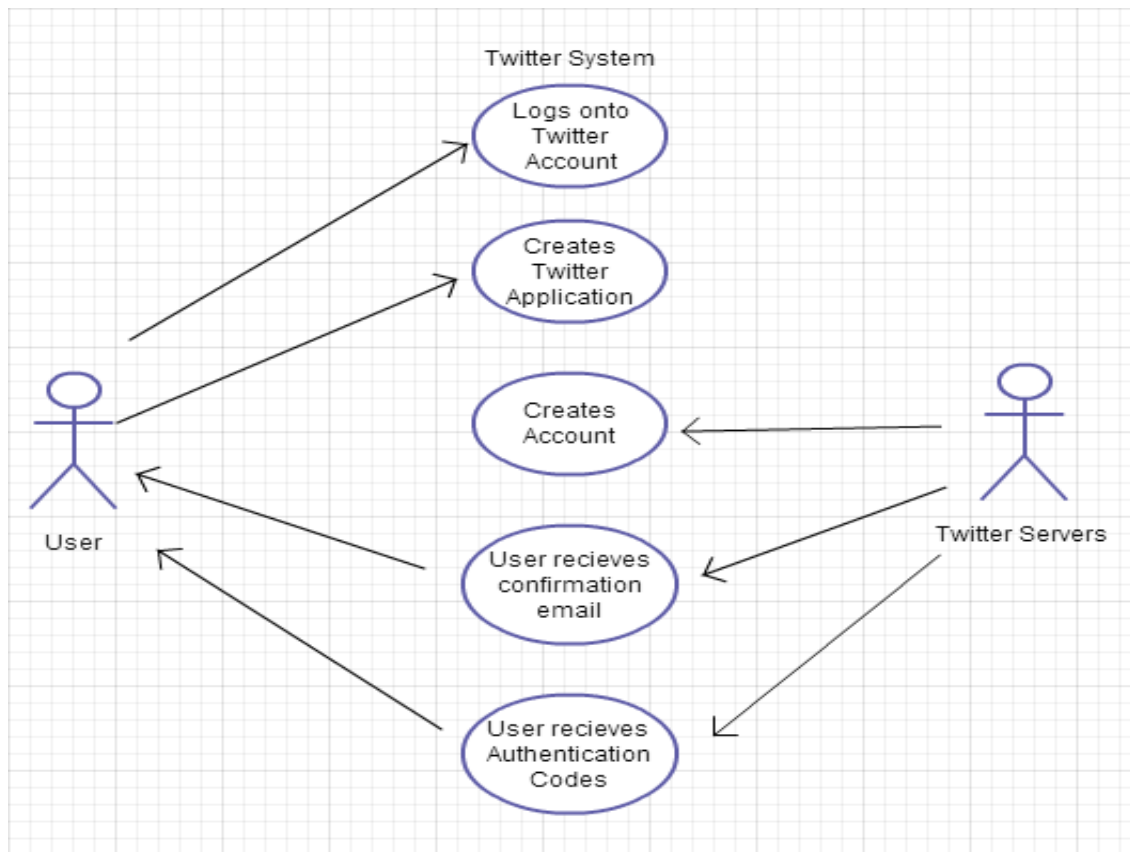


Figure 32: Use Case Diagram 1

Flow Description

Precondition

The user must log onto twitter and create a twitter account and then create a twitter application account.

Activation

The use case is initiated when the user inputs the required details and submits them.

Main Flow

Twitter receives the applications and processes the details. The user then receives a confirmation email and gains access to their new account, receiving their authentication codes.

Post-conditions

The user has been granted the authentication codes and has access to the twitter API.

Non-Functional Requirements

Non-functional requirements describe the required qualities of a project. These supplement the documentation of functional requirements, which describe the behaviour of the project.

Reliability

Potential errors will be managed by rigorous testing of the programming code used for the retrieval of the data for the analysis.

Performance Efficiency

The execution of the project and its deliverables has been highlighted in a time management plan seen in the project proposal.

Operability

The output of the analysis aims to be clear and logical for the audience viewing it.

Maintainability

The structure of the project will provide a framework for analysis to be made with different stocks. The main program of the project will also be capable of operating sentiment analysis on other topics such as a business product or a political candidate.

Appendix 3: Code Segments

- Python code to access Twitter API

```
from tweepy import OAuthHandler
from tweepy.streaming import StreamListener
import time

consumer_key = 'arlhD7Cbx1lWH8G71UxTA'
consumer_secret = 'DvppaWwlvVz59V1Qt9y4ichg0S0c681ctuw2fzY1is'
token = '2332481052-Zp6FafDQDsqsWyIvSqWZ1nW5vS7VANDv7tWQq3N'
token_secret = 't0iIx9Amk15DrLFrHHOrJPrxpkcXPMYuzRGJNWBI42X4'

class listener(StreamListener):

    def on_data(self, data):
        try:
            tweet = data.split('","text":') [1].split('","source') [0]
            print tweet

            saveThis = str(time.time())+'::'+tweet
            saveFile = open('tweetDB.csv', 'a')
            saveFile.write(tweet)
            saveFile.write('\n')
            saveFile.close()
            return True
        except BaseException, e:
            print 'failed on data,', str(e)
            time.sleep(5)

    def on_error(self, status):
        print status

auth = OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(token, token_secret)
twitterStream = Stream(auth, listener())
twitterStream.filter(track=["AAPL"])
```

- Python code to scrape words and populate SQLite database

```
import time
import urllib2
from urllib2 import urlopen
import re
import cookielib
from cookielib import CookieJar
import datetime
import sqlite3

cj = CookieJar()
opener = urllib2.build_opener(urllib2.HTTPCookieProcessor(cj))
opener.addheaders = [('User-agent', 'Mozilla/5.0')]
conn = sqlite3.connect('knowledgeBase.db')
c = conn.cursor()
startingWord = 'good'
startingWordVal = 1
synArray = []

def main():
    try:
        page = 'http://thesaurus.com/browse/'+startingWord+'?s=t'
        sourceCode = opener.open(page).read()
        try:
            synoNym = sourceCode.split('<a href="http://thesaurus.com/browse/great" class=common-word data-id="1">')
            x = 1
            while x < len(synoNym):
                try:
                    synoNymSplit = synoNym[x].split('<div class="synonyms-horizontal-divider"></div>')[0]
                    synoNyms = re.findall(r'"text">(\w*)</span>', synoNymSplit)
                    print synoNyms
                    for eachSyn in synoNyms:
                        query = "SELECT * FROM wordVals WHERE word =?"
                        c.execute(query, [(eachSyn)])
                        data = c.fetchone()

                        if data is None:
                            print 'not here yet, let us add it'
                            c.execute("INSERT INTO wordVals (word, value) VALUES (?,?)", (eachSyn, startingWordVal))
                            conn.commit()

                        else:
                            print 'word already here'

                    except Exception, e:
                        print str(e)
                        print 'failed in 3rd try'

                x+=1

            except Exception, e:
                print str(e)
                print 'failed 2nd try'
        |
    except Exception, e:
        print str(e)
        print 'failed in the main loop'

main()
c.execute("INSERT INTO doneSyns (word, value) VALUES (?)", (startingWord))
conn.commit()
```

- Python code to calculate sentiment rating

```
import sqlite3
import time

conn=sqlite3.connect('knowledgeBase.db')
c=conn.cursor()

negativeWords = []
positiveWords = []

sql = "SELECT * FROM wordVals WHERE value =?"

def loadWordArrays():
    for negRow in c.execute(sql, [(-1)]):
        negativeWords.append(negRow[0])
    print 'neg words loaded'

    for posRow in c.execute(sql, [(1)]):
        positiveWords.append(posRow[0])
    print 'pos words loaded'

def testSentiment():
    readFile = open('AALP May 7.csv','r').read()

    sentCounter = 0

    for eachPosWord in positiveWords:
        if eachPosWord in readFile:
            sentCounter += 1

    for eachNegWord in negativeWords:
        if eachNegWord in readFile:
            sentCounter -=1

    if sentCounter > 0:
        print 'this text is pos'

    if sentCounter == 0:
        print 'this text is neutral'

    if sentCounter < 0:
        print 'this text is neg'

    print sentCounter

loadWordArrays()
testSentiment()
```

- Python code to source trading data from Yahoo

```
import urllib2
import time

stockToPull = 'AAPL'

def pullData(stock):
    try:
        fileLine = stock+'.txt'
        urlToVisit = 'http://chartapi.finance.yahoo.com/instrument/1.0/'
                        |+stock+'/chartdata?type=quote;range=1m/csv'
        sourceCode = urllib2.urlopen(urlToVisit).read()
        splitSource = sourceCode.split('\n')

        for eachLine in splitSource:
            splitLine = eachLine.split(',')
            if len(splitLine)==6:
                if 'values' not in eachLine:
                    saveFile = open(fileLine,'a')
                    lineToWrite = eachLine+'\n'
                    saveFile.write(lineToWrite)

        print 'Pulled',stock
        print 'sleeping'
        time.sleep(5)

    except Exception,e:
        print 'main loop',str(e)

pullData(stockToPull)
```

Bibliography

Chen, R. & Lazer, M., 2011. Sentiment Analysis of Twitter Feeds for the Prediction of Stock Market Movement.

Harper, D., 2010. *Forces That Move Stock Prices*. [Online] Available at: <http://www.investopedia.com/articles/basics/04/100804.asp> [Accessed Tuesday May 2014].

Roos, D., 2007. *How to Leverage an API for Conferencing*. [Online] Available at: <http://money.howstuffworks.com/business-communications/how-to-leverage-an-api-for-conferencing1.htm> [Accessed Tuesday May 2014].

Sentdex, 2013. *How to use the Twitter API v1.1 with Python to stream tweets*. [Online] Available at: <https://www.youtube.com/watch?v=pUUxmvl2FE&feature=youtu.be> [Accessed February 2014].

Sentdex, 2013. *Python Charting Stocks/Forex for Technical Analysis Part 2 - How to get free Stock Prices*. [Online] Available at: https://www.youtube.com/watch?v=Eh_E1NqdmLI [Accessed May 2014].

Sentdex, 2013. *Sentiment Analysis and Big Data: Machine Learning, Building Word Values*. [Online] Available at: <https://www.youtube.com/watch?v=9EM7qVnTSVQ&list=PLQVvva0QuDfRO5bQFLcVgvlOIhNUZpZf&index=12> [Accessed February 2014].

Sentdex, 2013. *Sentiment Analysis and Big Data: Machine Learning: Accuracy Testing*. [Online] Available at: <https://www.youtube.com/watch?v=wBRLcOMSpy4&index=13&list=PLQVvva0QuDfRO5bQFLcVgvlOIhNUZpZf> [Accessed March 2014].

Sentdex, 2014. *Who we are*. [Online] Available at: www.sentdex.com/about-us/ [Accessed Wednesday May 2014].

Strickland, J. & Chandler, N., 2007. *How Twitter Works*. [Online] Available at: <http://computer.howstuffworks.com/internet/social-networking/networks/twitter2.htm> [Accessed Tuesday May 2014].

Zhang, L., 2013. *Sentiment Analysis on Twitter with Stock Price and Significant Keyword Correlation*. The University of Texas at Austin.