SENTIMENT ANALYSIS OF SOCIAL MEDIA WITH DATA VISUALISATION

BY

JOHN KELLY

SOFTWARE DEVELOPMENT

FINAL REPORT

5TH APRIL 2017

TABLE OF CONTENTS

ABSTRACT

The purpose of this document is to provide an overview of the project. It will outline the specifications of it, what was achieved, the learning outcomes and the problems encountered. This document reflects the finished product at the end of the project.

1. INTRODUCTION

This document reflects the final product that was created as part of a fourth-year project in the Software Development course at the Institute of Technology, Carlow. The duration of this project lasted 26 weeks between October 2016 to April 2017.

This document aims to give the reader an understanding of the development of this project. This will describe what was required of this project. Next, what was achieved in this project will be discussed. This will state what was completed from the specifications and what the final product is. The problems that were encountered will be discussed and the solutions will be stated. Also, included in this document, is the future features that are planned for this project.

Sentiment Analysis is the extraction of opinions from a piece of text. This opinion can be expressed in different ways, however the chosen method of representing it for this project is using the positive and negative emotions. This project aims to provide a means of examining the sentiment expressed for a chosen topic by the user. The results will be displayed in a readable way for the user to examine.

2. PRODUCT DESCRIPTION

2.1 OPINION MINING EXPLAINED

Opinion Mining is the name of this project. This was chosen because of the generalised scope of this project. Opinion Mining is a tool which enables the user find the public sentiment opinion of a chosen topic. This topic can be anything ranging from a person, product or sports team. This project is implemented using the Twitter Social Network to download tweets and to analyse the sentiment expressed in them. As Twitter is one of the more popular social networks to express opinions on as the character count is restricted to 140-characters. This allows the users to express clear opinions in tweets. This opinion is then examined by a number of different machine learning algorithms and the results are then presented for the user to view in several different ways.

This project also benefits developers, as the project is developed in a generalised way. It allows the developers to change and adapt the application into their specifications. There are several different elements that can be adjusted including the platforms to be analysed, classifiers, analytics type and classification. These elements are described in detail in the design document.

2.2 APPLICATION DESCRIPTION

This application can be separated into several main parts. Each element will be described along with screenshots of the website and phone application. The frontend of this application is implanted in AngularJS. This enabled the website to be transposed onto a phone application using the Ionic framework. This was done with minimal changes to the core code. This allows the application to run on both android and IOS. All devices connect to the backend using an application programming interface (API).

2.2.1 MAIN SEARCH SCREEN

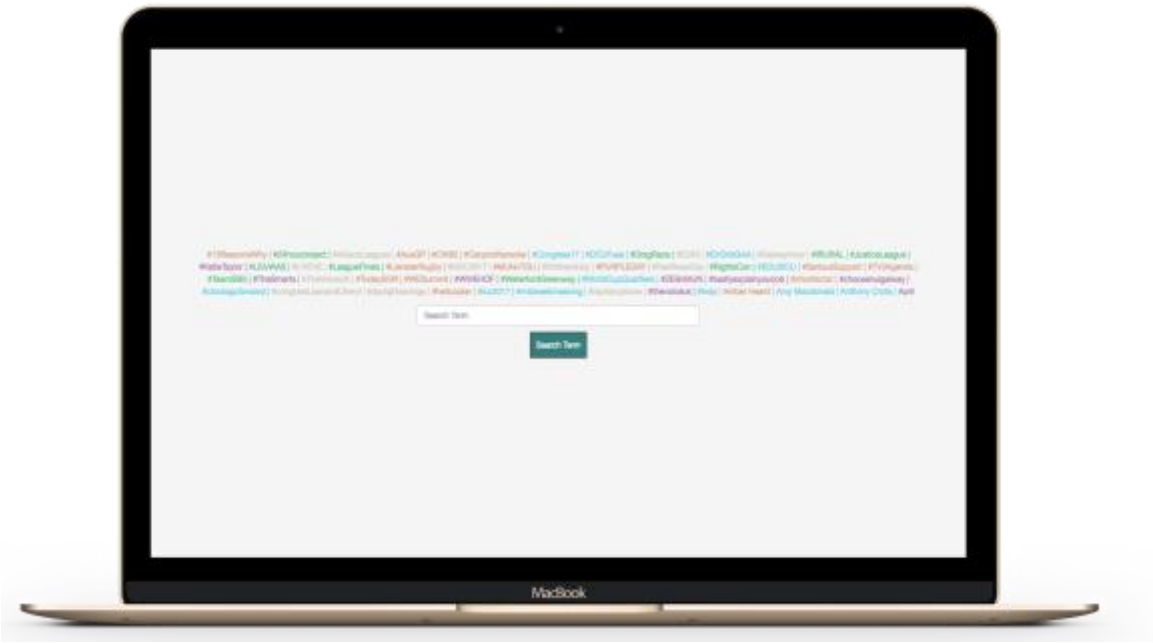

Figure 1. The main search screen on desktop.

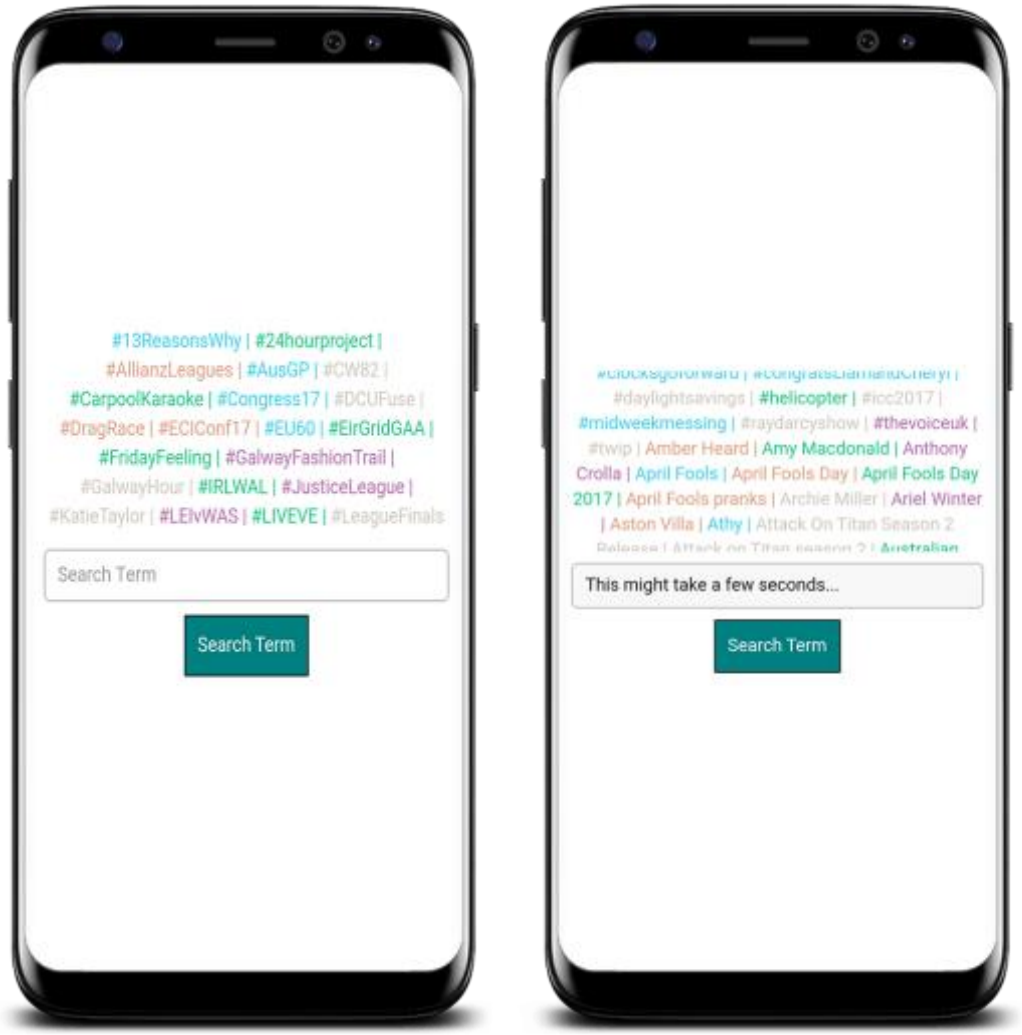

Figure 2, 3. The main search screen on the phone application.

The images from figure 1, 2 and 3 are screenshots from the website and phone application which presents the main search screen. This will be the first screen that will be presented to the user. This screen contains the current trends to help the user choose the term they want to find the sentiment for. As well as this, a search box will be present if the user wishes to search for a topic not currently present in the current trends. When the user submits a term, all fields lock to disable the user from submitting another term. This will be indicated to the user by the text changing in the text box (figure 3).

2.2.2 RESULTS PAGE

Figure 4, 5. The results page of the analysis on desktop.

Figure 6, 7. The results page on the phone application.

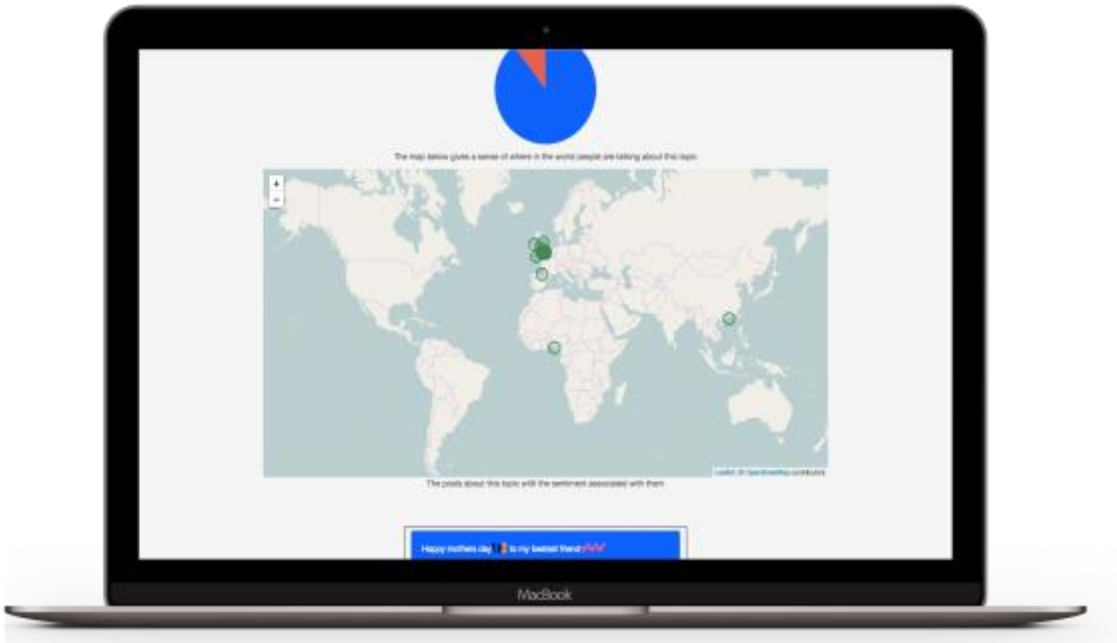

Figure 8. Results page with map to show results. The search term was Mother's Day

Once the user submits a term, the device sends it to the backend. The returned results are then presented in several different ways depending on the device the user is using to view the content. If the user is on a desktop/laptop the results will be presented on a line graph showing the sentiment at certain times. If any of the tweets have coordinates associated with them, they will be presented on a map. This can be seen in figure 8 which shows where tweets are sent from. If no coordinates are present in any of the tweets for the search term, then the map won't appear. This can be seen in figure 5 where the map has been removed.

The phone application provides less information for the user as it was taking too long to render all the graphs that are provided in the web based version. Having a quick response time for a phone application is essential for an application. This suffers severely with this application to begin with, as the analysis phase takes time to complete. It was decided that the pie chart which is present in figure 4 and 6 should remain to present the total of each sentiment expressed in the tweets.

The tweets shown in figure 5 and 7 are classified into the different sentiments expressed as red for negative and blue for positive. This is the trend throughout the entire page. Allowing the users read the classified tweets enhances the trust in the application. This is important as it increases the reliability of the application.

2.2.3 APPLICATION PROGRAMMING INTERFACE

The application programming interface (API) used in this application is used to communicate between the frontend and backend. This is used for both the website and the phone application. Once the backend receives a call from the frontend, it checks the database for the required information and then returns data to the frontend.

When the user requests the sentiment for a specific term the frontend requests the data from the backend. This will trigger the API module to search the database for occurrences of the searched term. If no data is found in the database, the API module will request tweets from Twitter. The tweets will be downloaded, classified and formatted. The formatting of the data adapts to the sentiments/classifications expressed in the data. This feature allows a developer to change the system from a sentiment analysis tool to a spam filtering system for example and the API will automatically change to spam and not spam. This feature was mentioned design document.

2.2.4 CLASSIFICATION ALGORITHMS USED

This project decides on the sentiment of a piece of text by using the classification machine learning method. This requires the use of classifiers to be used. The classifiers used in this project are:

- Multinomial Naïve Bayes
- Logistic Regression
- Support Vector Machines

These classifiers are described in detail in the research document. The first step in using any algorithm is deciding on a dataset which will be used to train and test the accuracy of the classifier. The chosen dataset is a combination of two datasets, one from the University of Michigan and another was a dataset compiled by Niek Sanders. This dataset has a total of 1,578,627 classified tweets. These tweets are classified into positive tweets noted as a 1 and negative tweets noted as a 0. (ThinkNook, 2012)

The next step of using a classifier is training it. This must be done for each of the classifiers being used. This means that the classifiers be presented a selection of the dataset with the elements needing to be classified and the correct classification value. This allows the classifiers to adjust to the data. Once the classifiers have been trained, it must be tested to see how it behaves on unseen data. This decides the accuracy of the classifier.

In this project, the training method was to use 80% of the dataset to train the classifiers and 20% as the test set. The preprocessing used for the classification includes formatting the text to normalise it, this includes the removal of usernames and replaces a hashtag with the word. Next, the sentence is then tokenized, this was explained in the research document. The stop words are removed and finally n-grams are used. N-grams are co-occurring of words. This is used to consider negating words. (Ganesan, 2016) For example, “not”, “good” are examples of unigrams. However, “not good” is an example of bigram. Unigrams, bigrams and trigrams are used to maximize the accuracy of the classification. After the text has been successfully formatted, the term frequency is

calculated and inverted as normally the words that are used less are more crucial in deciding the sentiment of a section of text.

As this application uses three different algorithms, a voting mechanism was implemented to calculate the final sentiment. Each model classifies the text and the voting system will calculate the sentiment with the most votes. This is then selected for the final sentiment for that tweet.

```
Multinomial_Naive_Bayes Accuracy: 78.64 (+/- 0.00)
Logistic_Regression Accuracy: 80.37 (+/- 0.00)
LinearSVC Accuracy: 78.97 (+/- 0.00)
```

Figure 9. Classifier name, Accuracy and standard deviation

The classifiers were developed with the idea of allowing other developers to add and change the models. This allows the application to be used in different ways. The addition of the voting mechanism allows developers join several classifiers together which allows more accurate results.

2.2.5 DOWNLOAD OF TWEETS

Tweets can be downloaded in two different ways. The first way is when no mentions of the term is mentioned in the database, this means it will begin downloading tweets and once they are classified, it will return the data to the user. The second way of downloading tweets is when the server administrator runs the continuous collection of tweets script.

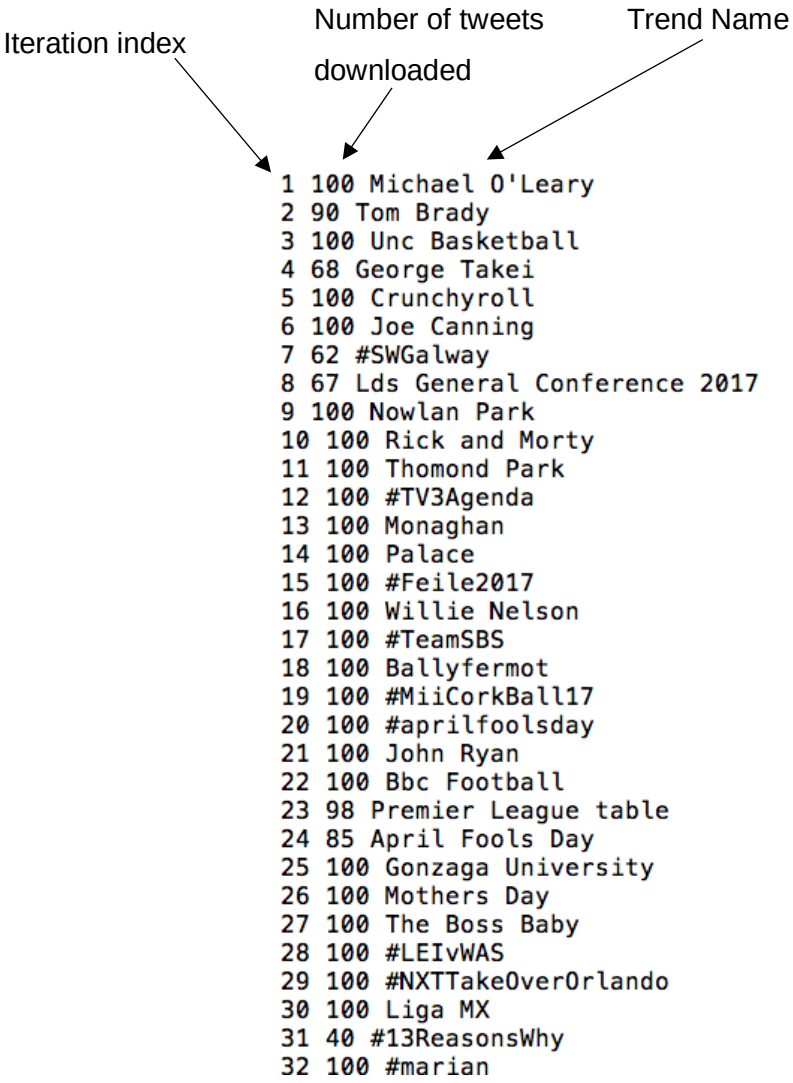

Figure 10. The trending topics on the 2nd April 2017 which were used to download tweets.

When the server administrator runs the script to download the tweets it will first download all the current trends from Google and Twitter in America, Ireland and England. Figure 10 shows an example of what the server administrator will see once they run the script. Examining the first line, the first number expresses the iteration index. Twitter allows a total of 450 requests per 15 minutes, this will be the max iteration index for the download of tweets until it resets and starts again. The second element is the number of tweets that were downloaded for that request. Twitter has a limit of 100 tweets per request. Next is a string, this contains the trending topic name that has been downloaded.

When the tweets have been downloaded, there are a number of features that need to be extracted from it and stored in the database. These features are:

- **Text –** This is the contents of the tweet. This will be analysed to determine the sentiment.
- **Timestamp –** This is when the tweet has been posted. This will be used for the graphing of the sentiment on the line graph.
- **Favourite –** This is the number of people “like” the tweet.
- **Retweet –** This is the number of people who reposted the current tweet to their profiles.
- **Coordinates –** This is where the tweet was sent from. The accuracy of this is dependent on the setting the user has specified. They can give their exact location, for example if there was a rugby match on, the user might want to send a tweet from the stadium. However, the next tweet will have a more generic location. (Twitter, 2017) This will be used for the map feature on the website.

Although this project is implanted using Twitter, using another platform to gather the data would easily accessible for a developer. It was decided for this project to use the trending topics to download tweets. However, if a developer chose to add another platform, it wouldn’t be necessary to follow that standard.

3. CONFORMANCE TO SPECIFICATION/DESIGN

This project has conformed to many of the specifications that were stated in both the functional specifications and the design document. The classifiers shown in figure 9 shows the classifier accuracy between 78% to 80% which meets the requirement of the functionality specified in the functional specification document.

In the design document a wireframe of the user interface was provided. This design is very like the final version which includes all the elements stated however, the difference was that a small description of each graph was provided. This was decided that it needed to be added because although the graphs can be seen to be self-explanatory, user testing showed that a description needed to be added to give the graphs context. The date range was also added to allow the user to see exactly when the sentiment was selected from.

In the functional specification document, it was mentioned that this project would be invaluable for companies to track the sentiment for their products, promotions, etc. To examine this theory and to find what information a company might require from an application like this, I contacted the Assistant Managing Director and the Public Relation Officer from the National Ploughing Association, Anna Marie McHugh and Morag Devins. One of the key features that stood out for them was the sentiment timeline where they can view the sentiment being expressed at certain times. This way they can see reactions to certain events within the organisation. They expressed a lot of interest in a product like this and agreed that it would be invaluable to their business.

I also contacted Emma O'Brien who is the CEO of Emmagine Ireland who specialises in public relations and media marketing for different companies. I showed her a preview of the functionality of this project and she thought the trends were a great way to keep up to date with current topics of conversations. She mentioned that it would be useful for coming up with new novel advertising ideas for the companies she work with.

4. LEARNING OUTCOMES

4.1 PERSONAL

This project provided several main opportunities for personal learning. The first was the time management. This project was a massive undertaking and the specifications mentioned in the functional and design documents added to the complexity. However, the agile method of development taken in this project allowed the development to be on time. A strict schedule was followed throughout the entire project. At the end of each iteration, a functional prototype was created. In the first iteration, as stated in the functional specifications, development began on all aspects of the project with a major emphasis on the downloading of tweets module. The focus of the second iteration was on the development of all sections with an emphasis on the classifiers. Finally, the third iteration was focused mainly on the generalisation of the code, as well as creating more accurate classifiers. Time management was a crucial part on insuring that all the features were developed on time.

Furthermore, due to the time restraints on the project, decisions had to be made fast. This leads to the next personal learning opportunity, decision making. As I could construct all the requirements myself including the technologies used, most of which I had very little experience in using. There were a lot of decisions to be made. Research was a crucial part in the decision-making process, however, due to the lack of knowledge about implementing such decisions, the focus was to implement solutions a general and module so if it needed to be changed, it wouldn’t disrupt the code too much.

4.2 TECHNICAL

This project also enabled many technical skills to be exercised. An example of this was that I have had very little experience with some technologies used in this project. The backend of this project was created completely in Python which I have had no previous experience with. This added to the complexity of the project. Also, I had no experience with natural language processing. This was the main factor in the backend of the project. I have worked with AngularJS on work placement in Aol. This helped with the basic understanding of the uses of this technology and the standards that I needed to follow that enabled me to use this technology in a modular and readable way. I have had limited experience with the Ionic framework for creating the phone application. However, as its based on the AngularJS framework, only some of the code had to be altered to adapt to the change of device. I also have had no previous experience using classifiers or any machine learning algorithms before this project began, so a lot of research had to be conducted before implementing of the algorithms could begin.

There were several issues that occurred throughout the project which allowed me to gain experience troubleshooting the different technologies used. This enabled me to gather a greater knowledge of these technologies. It had allowed me to use problem solving skills and made me think different about different issues that arose.

Documentation and descriptive function and variable names were a major part of the development of the development of this project. This allows other developers to gain a better understanding of what the code does without spending hours reading it. As this tool can also be used by other developers to for different applications, it was necessary to have documentation so they will know how to use the software. This includes the argument types in functions and return types. This gives the reader a greater knowledge of how to use each function and what they require and return. This was essential for them maintainability of the project.

5. PROJECT REVIEW

5.1 WHAT WENT RIGHT?

I would consider the overall project to be a success. The classifiers are considered as accurate as a human at detecting the sentiment of a piece of text. This was a major achievement in this project and can be seen in figure 9.

This project adhered by strict standards in both the frontend and backend of this application. PEP 8 coding conventions for the style guide was implemented for Python and AngularJS follows the ESLint convention stated in the '`.eslintrc.json`' file in the frontend folder. This is enforced by the continuous integration tool TravisCI which is used to test the code and ensure its following the style guides standard. As GitHub is used for the version control system, every time the developer commits a piece of code, GitHub will trigger TravisCI to start a build. If the commit breaks a test or doesn't follow the style guide, the build will fail and will be clearly shown to the developer.

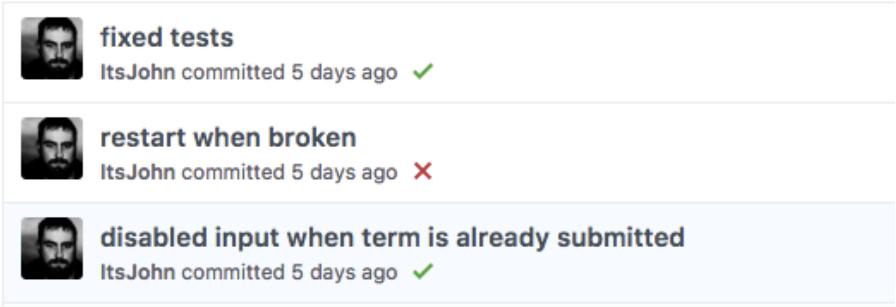

Figure 11. A test failing when the continuous download restart commit was added.

I thought the technologies used were chosen right for this project. AngularJS was used for the frontend of this project which allowed it to be transposed to a phone application with very little changing of code. Using an API to connect to the backend for the website version enhanced the change of device as the same code can be used for each device. The backend technology was Python as it has some essential modules that were used for text processing. These modules are described in detail in the research document.

5.2 CHALLENGES ENCOUNTERED

A project of this size encounters numerous challenges and the primary challenge was encountered with the classifiers. Python has a module called Natural Language Toolkit (NLTK). This module contains the core functionality for processing text. It also contains a function that allows NLTK to communicate with Pythons Scikit-learn module. This is a module used for machine learning algorithms. NLTK's function is called '`SklearnClassifier`' is used as a wrapper for Scikit-learn classifiers. (NLTK, 2015) It provides a more readable solution for the classifiers. Although it enabled me to learn more about the classifiers. It also led me write a lot of code to format the data and construct the correct structure that's accepted by the algorithms. The algorithms weren't classifying the text properly and took roughly four hours to train the classifiers with a dataset of 10,000 tweets.

This gave me two options; one option was to increase the dataset size which would increase the length of time it would take to train the algorithms. The second option was to remove that function altogether and use the raw classifiers provided by Scikit-learn. The latter option was chosen. Doing this required a complete rework on the format of the data for training of the algorithms as well as the testing. However, Scikit-learn provides several functions that were used to format the data. One of which was called '`CountVectorizer`'. This converts the text document to a token count matrix. (Scikit-learn, 2017) this is a less readable solution; however, the computer can process it much faster meaning the dataset size can be resized. This solution allows the classifiers to be trained with the specified amount of the dataset the developer specifies. In the case of this application, the classifiers were trained with 80% of the dataset that had 1,578,627 tweets.

Another challenge encountered in this project was the database. The original choice for the database was MySQL because when the elements were being extracted from the tweet, it was thought that a tabular structure would be the most suitable option for the storage mechanism. This was decided as it conformed to the first normal form. The database structure was constructed and during the implementation of the database

module, it was decided that it wasn't the correct method of storage as the tweets were already in a JSON object and some elements of the tweet may not be present in the object. An example of this was the coordinates attribute. It was decided to change to MongoDB which allowed the database to be defined as it grows. This decision solved the impedance mismatch of forcing the data into a format which didn't suit it.

6. FUTURE FEATURES

6.1 USER INTERFACE

The time constraints on the project allowed for a lot to be done. However, there are a number of features that, if time allowed, would have been added to the project. One feature of this project that could use an upgrade is the user interface (UI). Although the final UI is mostly the same as the original wireframe mentioned in the design document, it isn't very appealing to the user. This is a section of the project that could be upgraded in future development. Time did not allow for a complete remodel of the UI. However, all the elements on each page are clearly labelled for the user to understand the functionality and the meaning of each part of the UI.

6.2 ADDITIONAL PLATFORMS

Another future feature would be adding more platforms. All that would be necessary to add a platform is to create the files required to download the posts from the chosen platform. This would require research and time that was outside the scope of this version of the project. It would be interesting to see how this sentiment analysis project fairs against posts from Facebook, for example, and how it differs from how it behaves on Twitter. As Twitter is a great place for information as the 140-character tweet limit allows users post quickly what's on their mind. This would suggest that the tweets would be an honest judgement of what the publics opinion is for a certain term. Whereas, Facebook posts have no limit and allow for a lengthy expression of opinion.

6.3 ADDITIONAL CLASSIFICATIONS

Classification is a term used to describe the name of the categories that the post can be placed into. In this project, the classifications are either positive or negative. However, a future feature would examine the effects of adding more classifications, for example, neutral. This would add an entire new level of complexity to the classifiers, although, some posts are classified incorrectly as the classifiers can't decide on the correct classification as it is neither positive or negative, it is however, neutral. This was attempted to be added however no dataset could be found that had a large enough number of examples of positive, negative and neutral. Several datasets were used to experiment with different solutions, however it was decided to remain with the positive and negative sentiments for the above reason. A suggested solution is to allow the users to reclassify posts. If a user sees a post that is classified incorrectly, allow them to click on the post and give them several choices of sentiments to choose the correct option from. This would allow for a hand classified sentiment dataset to be created.

7. ACKNOWLEGEMENTS

I would like to thank Dr. Greg Doyle, the project supervisor for his help and advice throughout the year. He helped me keep the project going in the right and always pushed me to do more and asking me “When will you have a prototype?” to ensure I was making the goals that I set out in the beginning.

I would also like to thank my fellow classmates who were always there for a laugh and allowing me to bounce ideas off them.

Finally, I would like to thank all the other lecturers who provided insights into different sections of the project.

8. BIBLIOGRAPHY

Ganesan, K., 2016. *What are N-Grams?*. [Online]
Available at: http://text-analytics101.rxnlp.com/2014/11/what-are-n-grams.html
[Accessed 4 April 2017].

NLTK, 2015. *nltk.classify package*. [Online]
Available at: http://www.nltk.org/api/nltk.classify.html
[Accessed 4 April 2017].

Scikit-learn, 2017. *sklearn.feature_extraction.text.CountVectorizer*. [Online]
Available at: http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html#sklearn.feature_extraction.text.CountVectorizer
[Accessed 4 April 2017].

ThinkNook, 2012. [Online]
Available at: http://thinknook.com/twitter-sentiment-analysis-training-corpus-dataset-2012-09-22/
[Accessed 3 April 2017].

Twitter, 2017. *FAQs about adding location to your Tweets.* [Online]
Available at: https://support.twitter.com/articles/78525
[Accessed 3 April 2017].