**Sentiment analysis of social network project**

Introduction

Social networking and data analytics are a rapidly growing technology area. With billions of people on some sort of social network there is a wealth of data online. In the example of Twitter, an enormous volume of data is posted/ tweeted daily. All of this data in one place provides ample opportunity to extract useful information which can be leveraged to many areas of the technology industry to help develop and improve many businesses. Applying data analytics to the extracted data will allow the development of predictions, trending over time, popularity and many more. In the case of this project, sentiment analysis can be carried out to trend users feeling of certain topic areas allowing the operator to gauge the public’s interest and outlook on certain topic areas.

Business Case

Business Need

By using data from social networks businesses can benefit from information that is already available. Trending the publics feelings towards specific topic areas can be used in areas such as marketing, customer relations, advertising and more. This would add a layer to the business allowing them to access/ reach a larger audience where they before may not have been able to. Automating this process of retrieving data from Twitter, by using a Twitter API, will save time and money. Using hashtags to search for specific tweet will reduce the number of unrelated tweets retrieved for analysis.

Business Objectives

To increase a company’s productivity and to create/ grow the company’s data analytic capabilities by:

* Automating the Tweet retrieval process where the hashtag is the only thing changed.
* Storing all raw data in a series of tables in a sorted database.
* Populating the tables automatically when data has been retrieved.
* Using information gained from the analysis process to further develop the company and put the information to good use.

Project objectives

This project aims to provide a conclusive result formulated from the analysis of Twitter users based on the emotions portrayed in the Twitter content obtained. The objectives to aim towards are as such:

* To be able to extract tweets from Twitter based on text/ hashtag(s).
* To store the retrieved information in a useable fashion in a database system.
* Analysing tweets across time.
* Using sentiment analysis to determine the user’s emotion portrayed in the tweet.
* Graphically plot emotional change over time of tweets.
* Show how opinions of topic areas change over time.
* Determine the overall emotion felt towards topic areas.
* Allowing the extraction of useful information for all that stated above.

Initial Scope

This project is based on social network analytics. Mainly this will focus on sentiment analytics, trends and other interesting analytical outcomes of Tweets by Twitter users. The social network being used is Twitter, Tweets will be retrieved from Twitter using an API call. This API will be made using Twitter4J which is Java based code used to make calls to Twitter in order to retrieve the tweets based on some set criteria. The tweets to be retrieved will be based on specific hashtags and/or text present in the tweet. This data will then be stored in a database and used as the basis of the analytics to be carried out. There are three main components to this project these are as follows;

1. Java code using Twitter4J to carry out API calls to Twitter.
2. Database storage and management to store retrieved tweets and a collection of search phrases and sentiment phrases.
3. Data analytics and the production of graphical representation of the outcomes of the analytics.

Resources and Dependencies

* External hard drive storage if the data volume proves too much for local storage.
* Database storage if the storage for both an external hard drive and locally proves too much, or if it has been determined that using a database facility is needed.
* Access to tools such as Rapid Miner ( https://rapidminer.com/ ) and Tweet Stats (http://www.tweetstats.com/ ).

Method of Approach

Twitter4J is a Java based API program. It will be used, developed and written to meet the needs of this project and to carry out the main task of retrieving the data needed. The API results will be processed, formatted, split and copied to a table in a database which will be explained in more detail below.

The database will consist of a minimum of 3 normalised tables (although this is likely to be more in order to store additional information relating to the data); there will be a table which will contain all the search text needed to filter down which tweets to retrieve (these may be hashtags, keywords or phrases). This data could be anything from a topical current affair in the news or a trend that is currently occurring on Twitter/ the internet. Another table will contain all the sentiment phrases that will be used to determine the nature of each tweet, for example this table may contain words such as hate, love, like, dislike, terrible, horrible, brilliant, great etc. along with whether these words are negative, positive or neutral. The final core table will be used to store the tweets that have been retrieved by the API call and this table will be updated each time the API call is used.

The database will be stored locally on the host computer (the computer that is carrying out the API calls and analytics) and in the event that the database gets too big some data will be compressed and archived. If it is found that the database is still unmanageable the database/ data will be stored on an external hard drive with manageable chunks being accessed/ downloaded to the host computer at any one time.

The database, API and Java code will be used together as follows; the table of search text will be looped through with each record in the table altering the API call code to change which tweets will be returned. If this proves to be too intensive or time consuming the work load will be split across multiple threads and the GPU of the computer being used to increase performance with each thread taking one record from the table. The results returned by the API call will be processed by Java code and then written to the respective database table using SQL which will be embedded within a Java class allowing for ease of access and method calling.

The sentiment table will then be looped through, in the same fashion as the retrieval of tweets above, and compared to the results table where the data retrieved by the API will be stored. Each tweet/ record will then be assigned an overwhelming positive, negative or neutral feeling. These will be determined by the word(s) which the data contains in correspondence with the sentiment dictionary table.

This new dataset will then be used to carry out analytics. The analytics may include, but is not limited to, determining the sentiment of each text phrases and how this changed over time, showing the number of tweets over time containing certain text phrases, showing the most used/ unused emotional word in tweets from those present in the table and how these have changed over time, the number of emotional words used in tweets and the trend of this number relating to the overall sentiment of the tweet. All of the analysis, where appropriate, will be represented in both figures and a graphical representation. In order to do this tools such as TweetStats and Rapid Miner may be used. TweetStats can produce histograms displaying information such as number of retweets (and most common retweeting users), number of replies (and most common replying users), tweet density and interfaces used (Instagram, Twitter for iPhone etc.). The data science tool Rapid Miner can be used for providing graphical outputs of the analysed data.

This project will be carried out under an agile workflow with a series of sprints being planned out in advance and progress evaluated a minimum of twice a week. The results of this will cause planned work to be adjusted as necessary for upcoming sprints. All work will be version controlled and stored on GitHub within a repository under a secure SSH key and all sensitive information (such as twitter login information) to be sure to be omitted in the git ignore file. The Git structure will consist of a Master branch, a feature branch and a test branch. Work will be done on feature branches and tested on the test branch. The test branch will contain the current contents of the Master branch. Once testing has passed this test branch will merged to the Master branch and the feature branch will be deleted. This is to maintain the integrity of the work and to allow back outs and code reverts to be easy while also keeping track of progress of work done.

The findings of the analytics will be briefly written up into documentation with some explanation to what was done and background information.

A stretch goal will be to implement a predictive model to predict outcomes of certain events, such political votes, based on the Twitter users feeling toward each party/ candidate. If this is completed an even further goal is to use the Facebook API on Facebook statuses and carry out the same process as Twitter and compare the results.

Project Plan

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Plan** | | | |
| **Stage** | **Expected start date** | **Expected completion date** | **Products/ Deliverables/ Outcomes.** |
| 1. Initial idea proposal |  | December 9th | Initial write up and meeting on idea |
| 1. PID | January 30th | February 5th | PID |
| 1. Investigation and requirements | February 6th | February 15th | To investigate possible data storages and access e.g. SQLite, Oracle, MySQL etc. and Twitter retrieval APIs (How to use, which to use e.g. Twitter4J) analysis to be carried out and tools to use for this. |
| 1. High level design | February 16th | February 26th | Structure of the system i.e. the API links in to Twitter, Tweets retrieved to database and analyse carried out. |
| 1. Twitter data retrieval | February 27th | March 12th | To have a Twitter API set up and to be able to retrieve the data from Twitter. To be able to pass parameters to search Twitter for and the results retrieved are reliable. |
| 1. Set up database system and integrate into project. | March 13th | March 20th | To initialize the database and tables and to populate the sentiment table. |
| 1. Carry out sentiment analysis. | March 21st | March 31st | To search through populated tables and recognize the sentiment words. Use this to plot graphs. |
| **Easter Break** | **April 3rd** | **April 21st** |  |
| 1. Assemble and complete final report | April 24th | May 5th | Final PRCO304 Report |

Control plan

The following PRINCE2 control techniques will be employed:

1. Highlight reports as dictated by the PRCO304 module

2. Review meetings with project supervisor as dictated by the PRCO304 module (weekly); additional ad-hoc meetings as are necessary

3. Risk management; communication plan; quality plan; exception reports and plans as necessary.

4. Additional meeting with supervisor as needed

5. Record of work carried out kept.

Communication Plan

Review meetings will be held with the supervisor in line with the Control plan. Further ad-hoc communications will take place as needed either via email, SPMS or in the form of additional face-to-face meetings.

Initial risk list

|  |  |
| --- | --- |
| **Initial risk list** | |
| **Risk** | **Management strategy** |
| Schedule overrun. | Contingency has been introduced into the project plan. Highlight reports and review meetings will provide a regular monitoring of schedule. An exception plan will be developed, and approved by the project supervisor, in the event of more than 1 week’s slippage. |
| Difficulty learning/using the development technologies. | A very simple system prototype will be developed during Stage 2 (Investigation and requirements). Alternative technologies can be discussed if needs be with supervisor and other members of staff. |
| Loss of code/work/ data. | All code, work and data (if locally stored) will be version controlled on Git Hub. Commits and pushes will be made very regularly as work is carried out. |
| Lack of data retrieved. | If there is not sufficient data, or an event to retrieve data for, alternative events and search phrases should be used. If the problem continues to occur the analysis of the data will be changed accordingly. |
| Insufficient storage capabilities. | If it is found that the data storage of the data sets is not sufficient then old data will be deleted. Search phrases may also be altered and made more specific to reduce the number of results returned. |
| Insufficient time to complete work stated. | A MVP (minimal viable product) will be agreed upon ahead of time with the supervisor and this will be the new goal to be met. |

Initial quality plan

|  |  |
| --- | --- |
| **Initial quality plan** | |
| **Quality check** | **Strategy** |
| Requirements | Requirements will be checked to ensure that they are correct, relevant (i.e., traceable to the business objectives), complete, achievable and demonstrable. The requirements will also document required product quality criteria (e.g., usability). Prototyping, testing and discussions will be held to ensure requirements are met. |
| Design validation | The design will be checked against requirements compliance, DB normalisation and software design principles (e.g., cohesion, coupling). |
| Sub-system validation and verification | This will be conducted at the end of each stage. |
| Project/ code testing | Tests will be carried out as the project is developed to ensure each stage is fully complete before moving on. |
| System validation and verification | To be carried out when the project has been completed to ensure the full system is working together as expected with the correct outcome being produced. |

Legal, ethical, social and/or professional issues

(Project complies with the universities approved ethics application)