

National College of Ireland
BSc(Hons) in Computing – Data Analytics
2016/2017

Keith Carpenter
X13525447
Keith.Carpenter@student.ncirl.ie

Full Title

Text Analytics – An analysis of Political Media

Technical Report



Declaration Cover Sheet for Project Submission

Name: Keith Carpenter
Student ID: X13525447
Supervisor: Frances Sheridan

SECTION 2 Confirmation of Authorship

The acceptance of your work is subject to your signature on the following declaration:

I confirm that I have read the College statement on plagiarism (summarised overleaf and printed in full in the Student Handbook) and that the work I have submitted for assessment is entirely my own work.

Signature: Keith Carpenter Date: 10/05.2017

Contents

Executive Summary:	4
1 Introduction	6
1.1 Background	6
1.2 Technologies	10
1.3 Structure	11
2 System	13
2.1 Functional requirements	13
2.1.1 Use Case Diagram	13
2.1.2 Requirement 1 <Gather Data >	14
2.1.3 Requirement 2 <Pre process text >	16
2.1.4 Requirement 3 < Perform text analysis >	17
2.1.5 Requirement 4 <Statistical analysis>	19
2.1.6 Requirement 5 <Visualize data>	20
2.2 Non-Functional Requirements	22
2.2.1 Data requirements	22
2.2.2 Environmental requirements	22
2.2.3 Security requirements	22
2.2.4 Reliability requirements	22
2.3 Design and Architecture	23
3 Methodology:	26
4 Implementation	28
5 Results:	37
5.1 Experiment 1: Sentiment Analysis	37
5.2 Experiment 2: Speech classifier	38
5.3 Experiment 3: Political v non-political tweets	39
5.4 Experiment 4: Personality insights	39
5.5 Graphical User Interface (GUI) Layout	46
6 Testing	48
7 Conclusion, Research and Further development	53

8	References.....	54
8.1	References.....	54
9	Appendix.....	57
9.1	Project Proposal	57
	TITLE.....	57
	NLP: An analysis of text through multiple platforms.	57
10	Objectives	58
11	Background	58
12	Technical Approach.....	58
13	Special resources required.....	60
14	Project Plan	60
15	Technical Details.....	63
	Figure 1: Use Case 1	13
	Figure 2: Use case 2.....	15
	Figure 3: Use Case 3.....	16
	Figure 4: Use Case 4.....	18
	Figure 5: Use Case 5.....	19
	Figure 6: Use Case 6.....	21
	Figure 7: Architecture.....	23
	Figure 8: KDD	26
	Figure 9: Boxplot.....	41
	Figure 10: Word cloud of Inaugural Speeches.....	44
	Figure 11: Word Cloud of Donald Trump Text	45
	Figure 12: Sentiment Graph of Speeches by year	46

Executive Summary:

The purpose of this research project is to delve into political media and gain an insight into how politicians use text to communicate with their audience. Politics is a driving force in the world; they use power of persuasion, mass media and influence to gain supporters. The way that politicians do this is through their speeches and other modes of texts. The goal of this project is to use different text analytics tools to explore the domain of political media. Using NLP, analysis will be conducted to determine sentimentality tone, personality insights, similarity of politicians, how audience react and which buzzwords are used as form of influence. Analysis was conducted on presidential speeches and twitter as a source of political debate. An analysis of the emotional tone and psychological factors gathered from a politician's text allow us to see how they apply influence in their media. Sentiment analysis from Twitter shows how powerful the micro blogging platform is as a means of influencing political factors. The steps of this project are discussed in detail through applying the KDD methodology along with suggestions for further work.

1 Introduction

1.1 Background

The idea for the project stemmed from a class in data application development based on sentiment analysis and the power of text analytics. It was interesting to see how a computer could interpret what a person was saying and give back a relevant analysis. Using sentiment analysis, we can see the overall emotional tone of a person's text and we are able to gain an insight on their reactions and behaviour regarding certain topics. After researching into more forms of text analytics and Natural Language Processing, the idea came up of being able to look at how political media influences the world around us, also gather an insight into how these powerful people think as they are in charge of some of the most important jobs in the world. The domain of politics is a dominant source of textual based content that is publicly available online and easily gathered through information retrieval.

Required Reading: Big Five Personality traits:

1. Openness to experience: How open or closed your thinking is. High scores indicate interest in art, high emotionality, vivid imagination and intellect. Low scores tend to be more rational logical thinkers, don't like change.
2. Conscientiousness: High scores in this trait signify discipline and responsibility and good at planning. Low scores indicate impulsiveness and lack of organisation.
3. Extroversion: How people get their energy, high scores reveal extrovert people like going out and socialising and need to feed off other people's energy. Introverts are more interested in solidarity, they are often quieter people.
4. Agreeableness: High scores in this trait indicate a love of doing things for others and wanting everybody to get along. Low scores are suspicious and distant people who care more about self-interest than others.
5. Neuroticism: This trait measures emotional stability, highly neurotic people are easily stressed out. They are prone to anxiety, depression and anger. Low scores indicate a more calm emotional stable person.



Figure 1: Big 5 Traits

2 Literature Review:

Approaching this project it was vital to first research related works to what has been done in this field as it is a relatively new field in data analytics, the application of Natural Language Processing techniques is a fast and largely growing area. The three texts used in this review explore the use of IBM Watson to gather insights into personality traits, the use

of Twitter to analyse sentiments and the application of the Naïve Bayes algorithm to build a twitter sentiment classifier.

(Meysam Alizadeh et al, 2017) conducted a study into the personality traits of political extremists. “Is the appeal a rational response to material conditions or is it the expression of psychological and personality disorders associated with aggressive behavior, intolerance, conspiratorial imagination, and paranoia?” this report aims to discuss the mental state of extremists by comparing twitter messages written by political extremists and comparing them with non-extremist users. The report uses IBM Watsons personality insights API to gain a look at the emotional tendency of users using the Big Five personality traits. The users were categorized by gathering a list of extremist groups and finding twitter users who openly follow these groups, that allowed them to identify right wing extremists, left wing extremists and non extremist followers. Tweets were then extracted from these users and this allowed a personality profile to be created on the users, the personality profile was conducted using LIWC and IBM Watson. The results of the study indicated that left wing extremists show more negative and less positive than random twitter users where right wing extremists only appear to show more negative motion and do not differ in other factors compared to a random twitter user. The study used personality traits to gain an accurate portrait of left wing and right wing extremists.

.(Alec Co, et al, 2009) of Stanford produced a paper regarding the automatics classification of tweets by using sentiment analysis to assign them to negative, positive or neutral. They propose the approach of classifying the tweets with emoticon, unigrams and bigrams; their study shows that in applying three machine-learning algorithms (Naive Bayes, Maximum Entropy, and SVM) it is possible to achieve an accuracy measure of over 80%. This paper applies the KDD model to great effect and is a source of inspiration in how I will approach my project. They start off by extracting tweets and identifying what is positive and negative sentiment. The data is cleaned by removing usernames, links and repeated letters to make the data more accessible. The tweets are retrieved using the twitter API, the tweets are collected in a space of two months. The results show that analyzing tweets using unigrams and bigrams together it performs better with the Naïve Bayes and Maximum Entropy algorithms but shows a slight decrease in performance for SVM compared to unigram analysis alone. The paper demonstrates an

efficient framework for gathering tweets and running machine learning models to gain an accuracy measure for sentiment classification, however the study could have implemented more algorithms such as decision trees and neural networks as a means of contrasting the accuracy score to a further degree.

.(Andranik Tumasjan, et al, 0000) used text analysis of political sentiment from the German federal election to investigate is a micro-blogging social media platform an accurate source of political deliberation. The analysis was conducting using a dataset of over 100,000 tweets, the tweets were then analysis using LIWC(Linguistic Inquiry and Word Count). “An analysis of the tweets’ political sentiment demonstrates close correspondence to the parties' and politicians’ political positions indicating that the content of Twitter messages plausibly reflects the offline political landscape.” The names of the 6 main political parties were used as key terms and tweets containing these terms were stored in a dataset, along with gathering tweets from the timeline of prominent politicians in these parties. The sentiment of the tweets were analyzed in LIWC which is a text analysis software developed by Pennebaker that calculates a psychological and linguistic profile based on the sample text it’s given. The results of study show that they find twitter to be a plausible platform to not on spread political opinions but as a discussion tool for political agendas.

Aims

The main objective of this project is to use Natural language Processing techniques and text analysis to analyse multiple forms of political media in order to gain an understanding of how politicians communicate and how the publics sentiment reacts to political parties. To implement an accurate measure of this three main topics need to be looked at, authorship attribution, sentiment analysis and similarity of text. Different algorithms will be

implemented to find the similarity between different text corpora.. An authorship attribution algorithm will be implemented to verify authorship of text. Sentiment analysis will be used to view the overall emotional tone of text analysed.

Aim 1: Obtain relevant text resources from a variant to analyse

Aim 2: Use NLP to clean the data for analysis.

Aim 3: Perform analysis on text.

Aim 4: Use Algorithms to allow us to classify and predict data.

Aim 5: Visualise and report finding to see if there is correlation between different platform modes of texts

1.2 Technologies

Python/R: Python and R will be where most of the pre-processing of the data will be done.

The text will be gathered from URLs using a scraping script in R, this will allow the acquisition of multiple forms of text from varying websites, once the script is run the data will be stored in csv, txt or a mongo DB. Social media platform APIs will be used to stream tweets to allow analysis of sentiment tone in users tweets and also to view the emotional reaction to current events based on the hashtags and keywords. Once the data is gathered it will be imported and read into R or Python, which will then be used to clean the data. Once the text is effect enough to perform analysis on, NLP libraries and toolkits will be used to perform multiple features to gain an insight into the data and report relevant results.

MongoDB: A NoSQL document based database that will be used to stream live tweets to allow keeping up with current trends while they happen.

Watson: Watson IBM is a number of useful tools and API by IBMs blue mix team, the feature that will be used in this project is the personality insights tool that will allow a

profile to be created on political figures based on their twitter account, the profile is generated by Watsons API applying the big five personality traits.

RShiny: R shiny is a package in R that allows data visualisation in a web app, the package applies both a user interface and server script that allows the user to create live updating dashboards, interactive plots and other visualisations on a web page all through Rstudio.

Excel: Microsoft excel will be used to create and modify csv files also to generate statistical tests using the data analytics tool pack.

1.3 Structure

Text Mining Techniques	Key Considerations
Organising and Structuring data	<ul style="list-style-type: none"> • Clustering • Categorizing • Classification • Taxonomy
Text Processing	<ul style="list-style-type: none"> • NLP • Parsing • Tokenization • Stemming • Term Reducing • Part of Speech Tagging
Statistical Analysis	<ul style="list-style-type: none"> • Term Frequency • Keyword Frequency • Distribution • DTM • TF-IDF • Document Indexing
Machine Learning	<ul style="list-style-type: none"> • Clustering • Classification • Association Rules • Predictive Modelling

Classification Methods	<ul style="list-style-type: none"> • Native Bayes • Support Vector Methods • K-nearest neighbour
Model Evaluation	<ul style="list-style-type: none"> • Precision • Recall • Accuracy • Relevance
Visualisation	<ul style="list-style-type: none"> • Visualise data

This table shows the process of analysing text, it demonstrates how we can use text mining techniques in different stages of the KDD.

The overall structure of this document will apply the KDD framework and show how the initial plan came into effect through this application pipeline. The requirements for the project will be discussed, use cases will be created to demonstrate how the system will accomplish its functional requirements. The methodology behind each step of the project along with the algorithms to be used in the data modelling will be described in detail, next each step of the implementation will be documented. This will include how this project applied the KDD framework principles to create the analysis we set out to. The results of the implementation will be discussed along with evaluating the accuracy of the data models. The conclusion will end the document with a reflection of the project as a whole.

2 System

2.1 Functional requirements

1. Gather Data (Scarping, API) – the data will be taken from a source and stored into data store
2. Clean, Pre process data – the data will be imported into R and NLP libraries will be run to clean text data so it is ready for analysis
3. Perform text analysis – This is text data will be turned into quantitative data allowing creation of term document matrices and sentiment analysis
4. Statistical tests and creating data models – Here we can perform statistical tests and create data models with the transformed text
5. Visualize data – here the data will be plotted and visualised to help us interpret the results of the tests.

2.1.1 Use Case Diagram

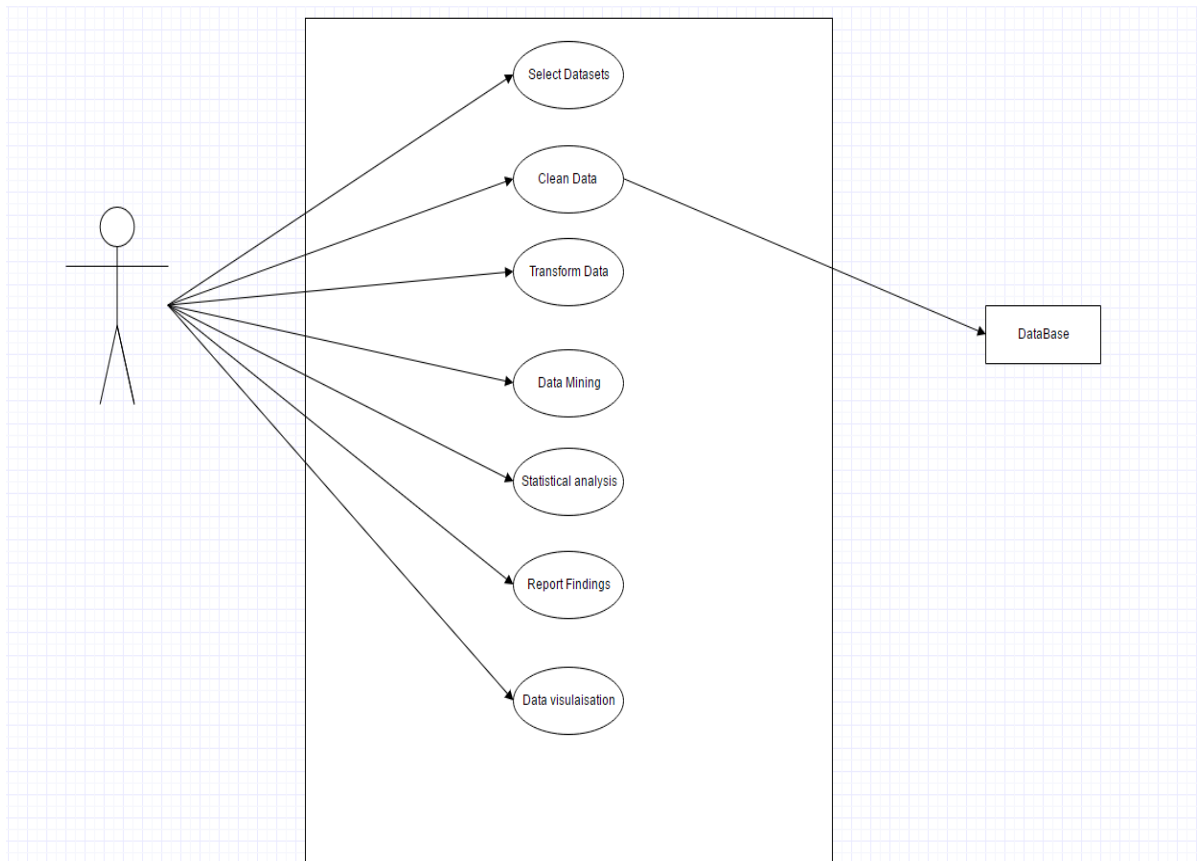


Figure 2: Use Case 1

2.1.2 Requirement 1 <Gather Data >

2.1.2.1 Description & Priority

This is the first and most important part of the system as without data it is impossible to achieve any relevant information

2.1.2.2 Use Case

Scope

The scope of this use case is to find appropriate datasets, websites containing useful information and social media content and pull the contents into a data store to import into a chosen IDE.

Description

This use case describes the acquiring of text from various sources, the data will be downloaded, scraped or pulled from REST APIs

Use Case Diagram

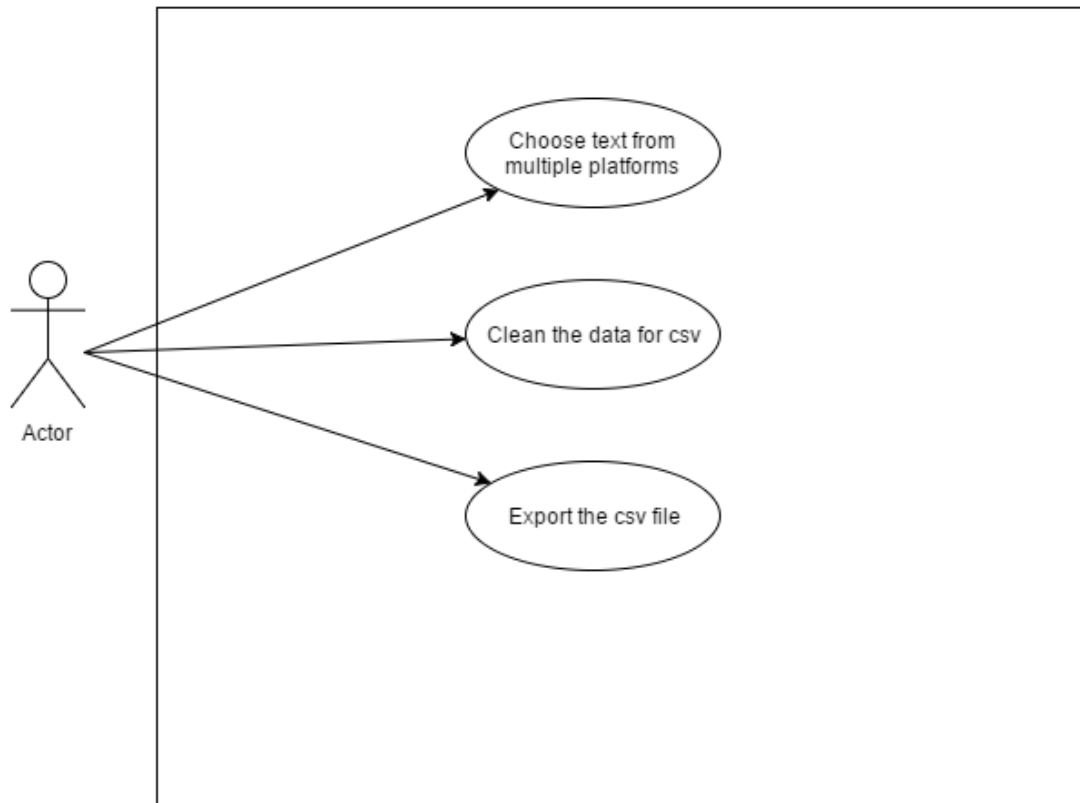


Figure 3: Use case 2

Flow Description

Precondition

The system has no data in data stores.

Activation

This use case starts when an <Actor> runs a script to scrap data, read csv files or pull data from a REST API.

Main flow

1. Data is taken from source and stored in a data store
2. The data is read into the IDE enviroirment

Post condition

The data file is imported into IDE enviroirment.

2.1.3 Requirement 2 <Pre process text >

2.1.3.1 Description & Priority

Here is will the textual data will be cleaned and created into a corpus for analysis

2.1.3.2 Use Case

Scope

The scope of this use case is to clean text and create a corpus

Description

This use case describes the cleaning and pre-processing of text

Use Case Diagram

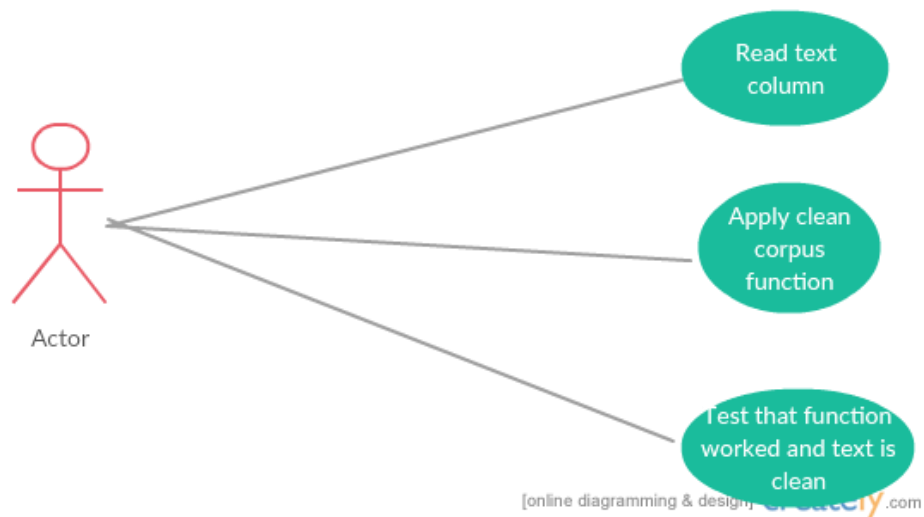


Figure 4: Use Case 3

Flow Description

Precondition

The text data is noisy and unfit for statistical analysis

Activation

This use case starts when an <Actor> imports text to R and Python

Main flow

3. Text files are gathered by the user
4. The user enters the text into Python/R to pre-process
5. The text is cleaned and ready to use for analysis

Post condition

The text is imported and ready to analyses

2.1.4 Requirement 3 < Perform text analysis >

2.1.4.1 Description & Priority

This is where we will transform text into numeric values, using a term document matrix, so we can perform statistical analysis and also allow us to plot text on graphs.

2.1.4.2 Use Case

Scope

The scope of this use case is to import data into the chosen programming scripts and use NLP libraries to gather results on the given text corpus. These will then be reported and the data will be used for statistical analysis, the term document matrix will allow us to create data models using bag of words.

Description

This use case describes the import of data into R/Python and the analysis of text by creating a TDM converting text into quantitative values

Use Case Diagram

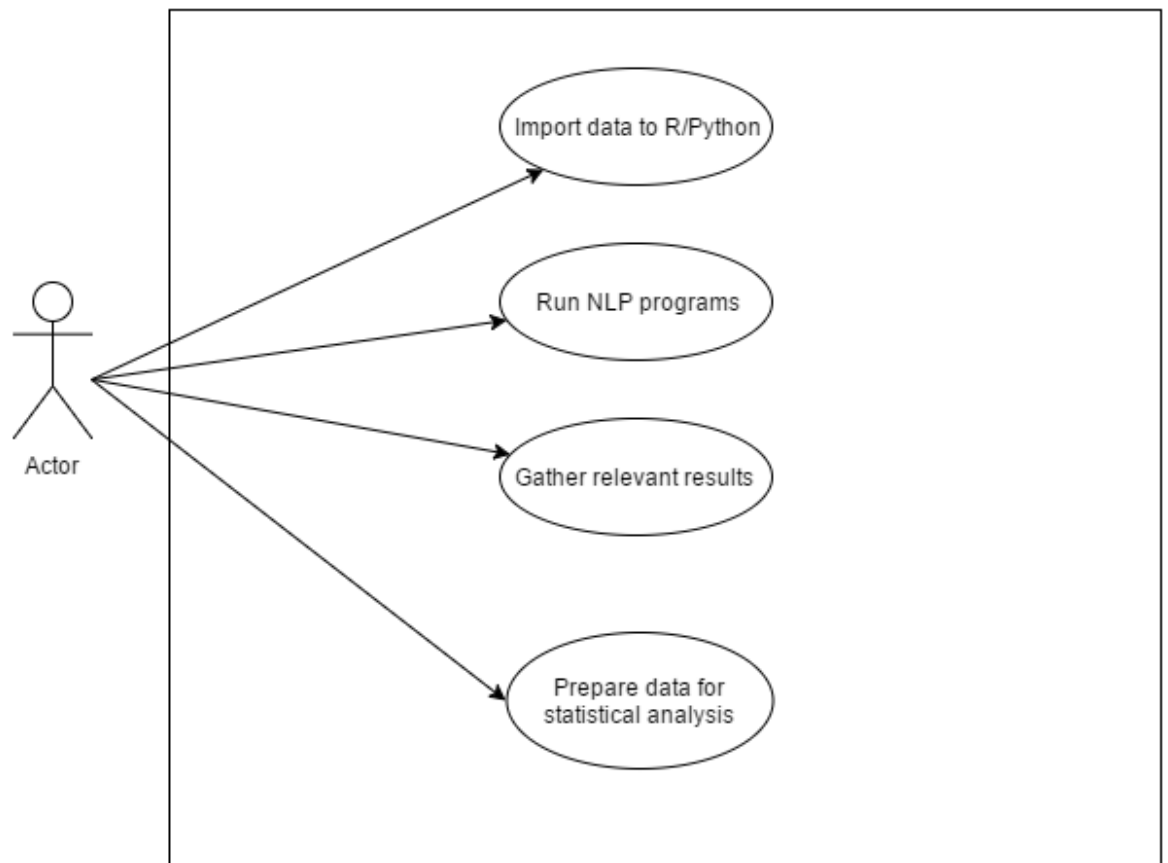


Figure 5: Use Case 4

Flow Description

Precondition

The text files are imported into the programming environment and cleaned accordingly to remove noisy data.

Activation

This use case starts when an Actor runs NLP libraries.

Main flow

1. The system identifies the text is in the programming environment
2. The actor will initialize NLP libraries
3. The system will return analysis of linguistic and lexicon structure

2.1.5 Requirement 4 <Statistical analysis>

2.1.5.1 Description & Priority

Use Machine learning and statistical algorithms to find out information on given text

2.1.5.2 Use Case

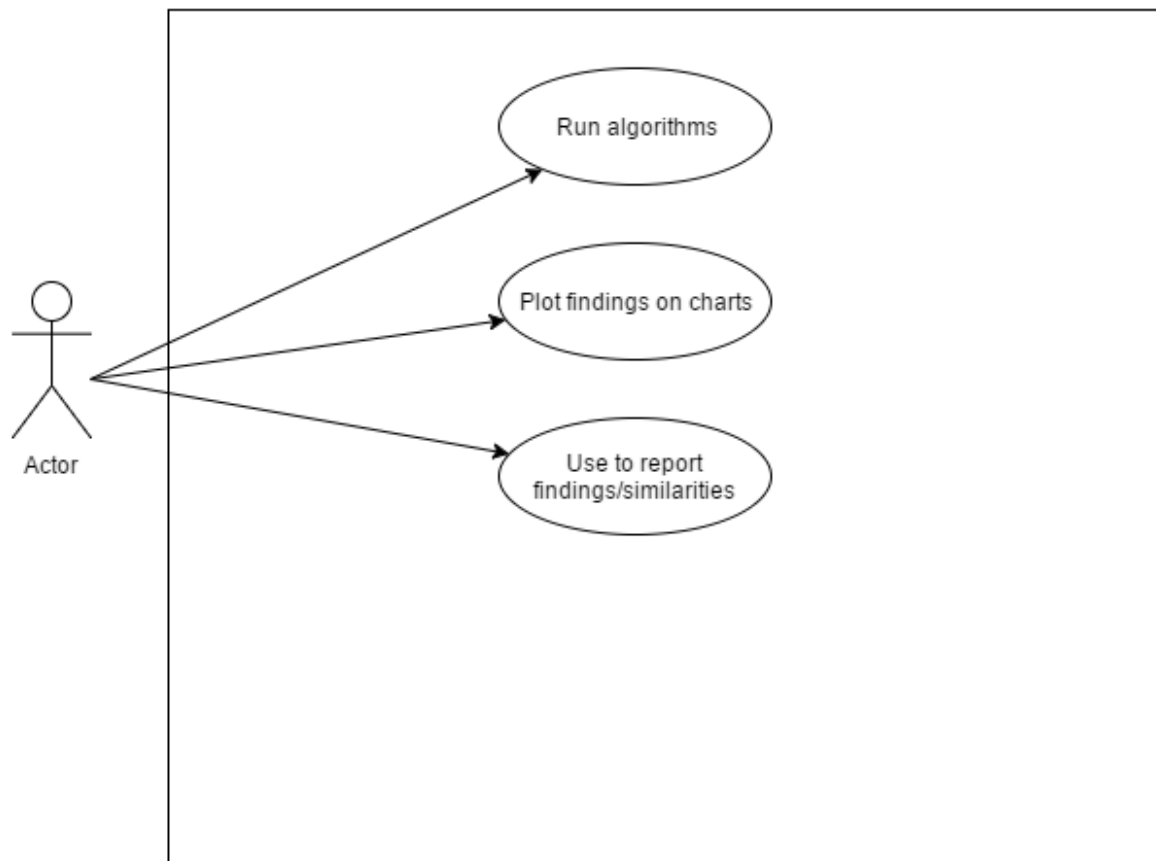


Figure 6: Use Case 5

Scope

The scope of this use case is to use statistic tests to find out information on the text and create models based on the text to allow us to categorize, classify and predict authors. We will also be running the personality insights to gain some stats on the text.

Description

This use case describes the use of algorithms in the system

Flow Description

Precondition

The text is tokenised and tagged ready for analysis

Activation

This use case starts when an <Actor> runs machine learning techniques

Main flow

6. The system identifies the text is inputted
7. The <Actor> uses machine learning algorithms to learn about the variants text.

2.1.6 Requirement 5 <Visualize data>

2.1.6.1 Description & Priority

Data will be adequately visualised to help the user understand more about the statistical analysis and help us to interpret the results of the tests

2.1.6.2 Use Case

Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.

Scope

The scope of this use case is to visualize data using wordclouds, plots, histograms and other relevant visuals

Description

This use case describes the visualisation of data using various packages in R.

Use Case Diagram

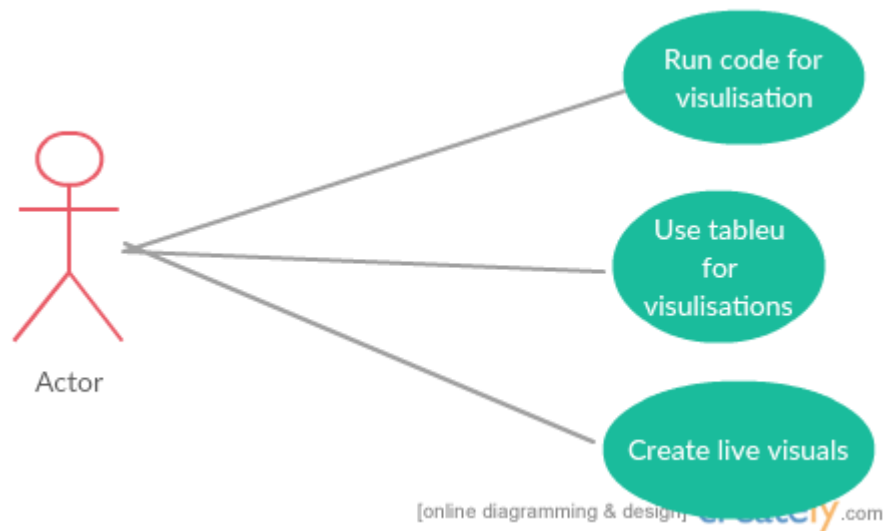


Figure 7: Use Case 6

Precondition

The system has taken transformed data and returned a visualisation to help the user to interpret the data results

Activation

This use case starts when an <Actor> runs the lines of code written that will output visualisations.

Main flow

8. The Actor plugs runs the script to visualise data
9. The system outputs the visualisation.

2.2 Non-Functional Requirements

Specifies any other particular non-functional attributes required by the system. Examples are provided below. **Remove the requirement headings that are not appropriate to your project.**

2.2.1 Data requirements

For the purpose of this project, data will be extracted from each possible platform of media available. Sufficient data will be required on political topics and will be gathered from URLs, Articles, book and social media APIs.

2.2.2 Environmental requirements

To run this project an environment of windows 7 or greater will be required. Python 2.7 or 3.4 will be required with the necessary packages installed. R studio 3.3.1 or higher will also be needed with the required packages. It is recommended that the performance of the computer used be of 4gb RAM or higher and use an SSD as the analysis of large datasets can cause problems in performance.

2.2.3 Security requirements

The software should never disclose any personal information of Twitter users. It is of high importance to keep the API keys secure. Twitter uses OAuth authentication as a means of privacy, to access twitter it is necessary to have a twitter account along with a valid phone number

2.2.4 Reliability requirements

The software will meet all of the functional requirements without any unexpected behaviour. The data models will be evaluated to show how reliable they are.

2.3 Design and Architecture

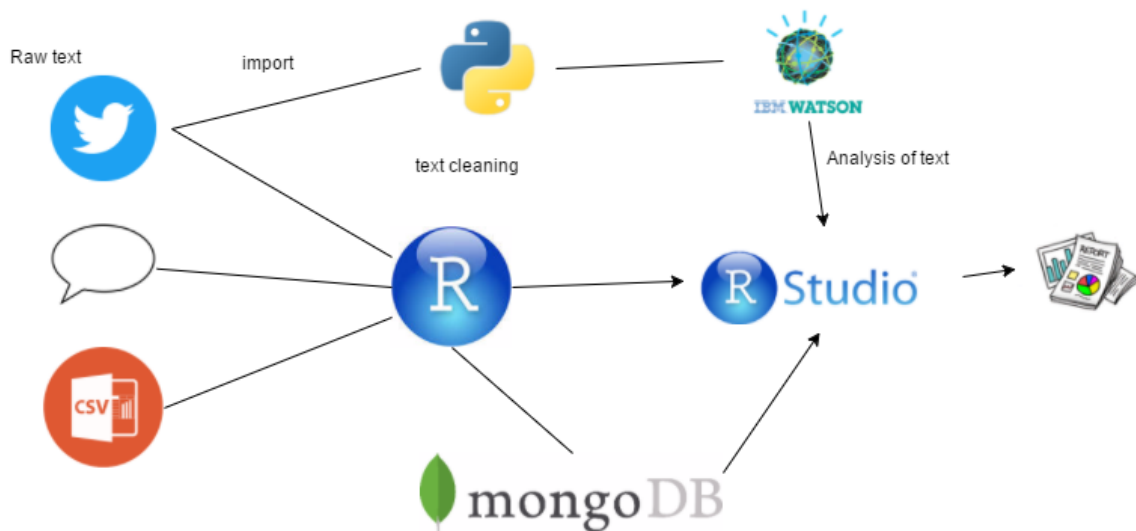


Figure 8: Architecture

This architecture diagram shows the process of how the system will work, firstly the user will gather text from online resources, the data will be parsed into a data store such as a csv or database then imported into R studio for text pre-processing and transformation. When the text is cleansed in a meaningful manner it is possible to use a plethora of NLP libraries, sentiment analysis and machine learning programmes on it to obtain relevant statistics. The transformed text will then be statistically analysed and then reported in this document.

- Naïve Bayes:

The Naïve Bayes classifier is a hugely popular and widely used classifier; it is based on using probability to classify items. It comes from the Bayes Probability Theorem, it is

widely used because of its fast and easy application. It is a supervised learning method meaning that we know the result we want, this makes the algorithm easy to evaluate. It works by feeding the algorithm training data based on classified information, then feeding it non-classified data. The algorithm will then predict the classification for the new data. This will be experimented with using twitter data, we will try to classify whether a tweet is written by a politician or a regular person based on the lexicon structure of the tweet.

posterior probability=conditional probability prior probability evidence

- K-Nearest Neighbour:

The K nearest neighbour algorithm is a clustering algorithm. “NN classifiers are defined by their characteristic of classifying unlabeled instances by assigning them the class of the most similar labelled instances.” The KNN model is called a lazy learner, as it does not create any more fields of data as it just classifies non classified information into classified information based on the already classified data. It is a simple but effective algorithm, it is used when data is complex and hard to categorize. The KNN model works in three steps, firstly measure the distance between X and each data point using the Euclidean distance measure, next it chooses the data points closest to X and then assigns X to the class of its closest data points.

- Decision Trees:

“A decision tree is a structure that includes a **root** node, branches, and leaf nodes. Each internal node denotes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label. The topmost node in the tree is the **root** node.” The decision tree in text mining concepts uses the root node as a term, and will use decision making to classify what class the decision will be based on the frequency of classes with that term. Eg. If the root node is the term “happy” it would be possible to classify text into

positive and negative categories by using a decision tree. The tree would create a structure and predict that if the text contains the word “happy” it would more than likely classify it as positive.

3 Methodology:

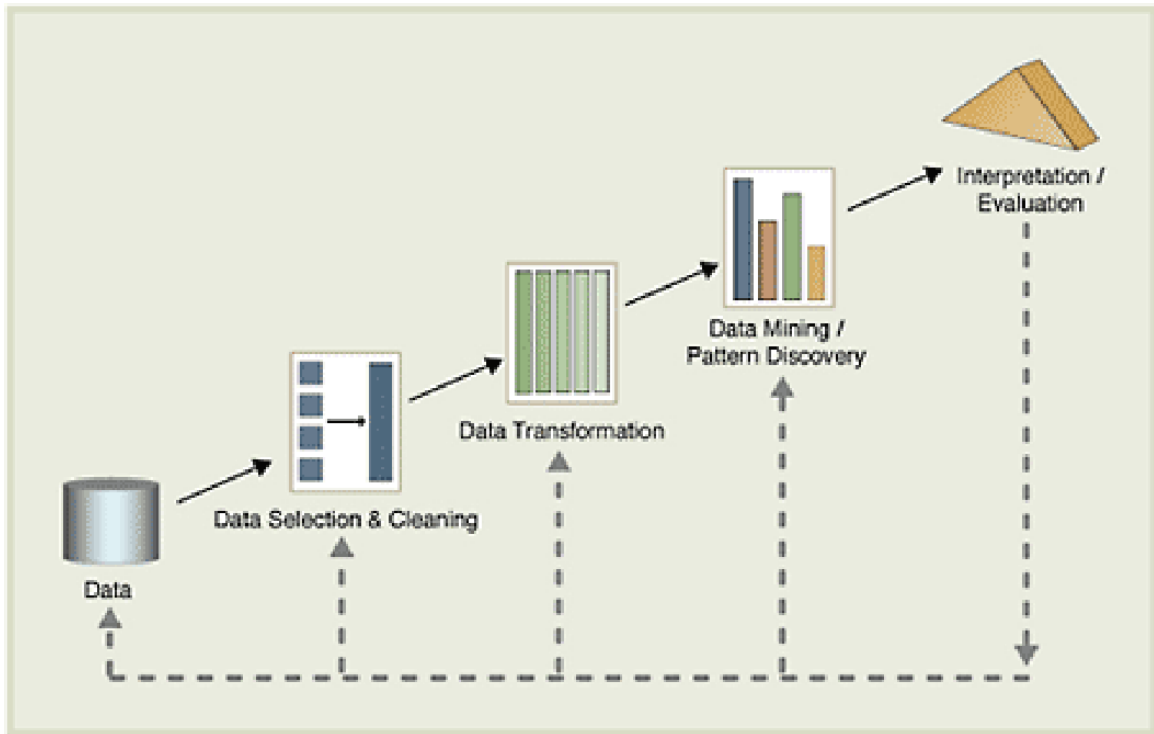


Figure 9: KDD

Data Selection:

It will be necessary to pick a variant with sufficient text to be able to analyse for relevant results. We will look at politicians, authors, celebrities to get a valid text corpus to analyse. The text gained from the internet can sometimes be in very raw format so we must use structuring methods such as clustering and categorizing.

Data Pre Processing:

In the pre-processing of data we must use techniques such as tokenising, POS tagging, stemming and chunking text in order to give us something we can use to analyse efficiently

Transforming Data:

The data will be used to run statistical procedures like Document term matrix, TF-IDF and term frequency to look at how many terms appear and a given text and this will return text is numerical values used as a weighting factor to see how important words are in text and

allow us to look further into machine learning. With the data transformed we can look at importing it into R, LIWC, Watson for sentiment analysis, personality insights, word clouds and similarity measures.

Data Mining:

Artificial intelligence, machine learning, statistics, and database systems will be used to look for trends and patterns in this data. In this part we will apply algorithms to look at the authorship, sentiments and document similarity.

Data Interpretation:

It is vitally important for the data to be interpreted and reported correctly, to insure this the data will be sampled accordingly to the domain in which is being investigated and ran multiple times to gain a adequate confidence level in the results which are reported. The statistical test will impose a hypothesis test that will allow a story to be told on the data,.

4 Implementation

Main features of the project:

A comparison was done on a collection of tweets using the key word 'ww3', 'brexit' and 'repeal' to gain an insight on the public reaction to current events. These tweets were then scored and plotted against each other to analyse.

.The next step in the implementation of this project also used twitter data. The idea was to implement a model that could predict whether a tweet was written by a politician or a normal everyday twitter user talking about other events. To do this test it was necessary to collect tweets from users that are non political and users that are in politics. To gather these tweets the handles of various presidents and current political heads were gathered and placed into a csv file with the label politics. In contrast a number of tweets were gathered using generic key words to drift away for political tweets to give a view of other activities such as sport and media. These tweet were placed in the same csv file and labelled non politics. Once the data was in the csv it was read into RStudio.

Another model was created to see if we can predict who said a speech by collecting numerous speeches by presidential elects and feeding them into a KNN algorithm. The speeches were gathered from online sources and stored into text files using both automated and manual scraping. The documents were then stored in a folder with the presidents name, this file path was then read into R to allow us to perform our analysis.

.To implement the IBM Watson personal insights feature a python script was developed to allow us to interact with the BlueMix technology. The idea was to pull tweets from a users timeline and gain an insight into their personality using the big five traits, we can then compare this user with another user and measure the difference between them. This application is useful in looking at the personality of political v non political, or comparison of two political figures.

.After acquiring a dataset from kaggle containing historical inaugural presidential speeches an analysis was conducted comparing the use of terms from presidents throughout history. To do this analysis a kernel was forked from kaggle containing a base of code for

analysis that we can work off. The author of this kernel has been referenced accordingly, the analysis was used for comparison bases.

.Live streaming twitter analysis: (work in progress)

Currently trying to build a live streaming twitter analysis that will be able to display live updating sentiment analysis based on a keyword. This will be implemented in an RShiny web app that will be hosted through R and allow user interface.

DOMAIN KNOWLEDGE:

The preliminary step in starting a data analysis project is to know which domain you are analysing and what you are trying to find out in analysing this field. The chosen field of political media is a wide topic but the goal of the analysis is broken down into a sections. In doing this projects the aims are to analyse current trends in politics using sentiment analysis, compare world leaders in personality and to build models that will be useful in determining whether media is considered political or not. The purpose of the implementation of this project is to research what makes world leaders tick, what the public thinks of the current global situation based on political agendas and to apply knowledge of data science principles to the domain of politics. The implementation process will take place using the steps of the KDD.

DATA GATHERING:

Data gathering was done in a numerous amount of ways as for this project a lot of text based data was needed, firstly twitter was mined for tweets about political agendas along with tweets from politicians themselves. To mine from twitter firstly a twitter account was to be set up along with a valid phone number, secondly a twitter app must be created on the app twitter page in order to be given access to the twitter api which is secured with outh authentication. In doing this four keys and secrets are received, these allow acces to the twitter api and streaming can begin. Secondly an R script must be created to access the api through R studio to do this the API keys must be logged into R studio as variables and then the package twtteR must be loaded in R studio which then allows communication between

R and twitter. Next is the Oauth setup which allows a secure connection to the twitter api to gather tweets. Once the tweets are gathered, they are created into a data frame and copied to a csv file.

DATA SCRAPING:

Data scraping took place using the R package rvest, this allows scraping of web pages where data is not easily downloadable. This package was implemented to gather some president speeches and export them into txt files. Another way data was gathered was through manual copy and pasting of president speeches into a text file for the speech classification model. This was done by locating collections of presidential speeches on the internet and then creating folder with these speeches with the name of the candidate, this file directory was then imported to RStudio for analysis.

KAGGLE:

Kaggle is a data science website where thousands of datasets are uploaded for analysis purposes, kaggle was scanned for datasets relevant to the project. Two datasets were the acquired from kaggle. Inauguration speeches from United States Presidents dating back to George Washington and a dataset comprised of Donald Trump and Hilary Clinton speeches. The Inaugural speeches dataset was a very interesting find as it saved time scraping the web for past speeches and it provided a large volume of text to create a corpus. The tweets dataset was combined with data that had already been scrapped from the twitter api to analyse political sentiment further. A recent data set uploaded in April was taken which contains the inaugural speeches of presidents dating back to George Washington, this dataset was added to the project to gain an insight into the changing of presidential speeches throughout history.

DATA PRE-PROCESSING:

Dealing with raw text data can sometimes be very noisy and messy. Multiple NLP processes were used to clean the textual data. The main package used was tm package in R. Using this package, it was possible to remove stop words, remove numbers, white space, convert documents to lower case, remove stem words and convert the corpus to a plain text document, this format is needed. This code snippet creates a function for cleaning text, this was applied to multiple scripts where the cleaning of the text was needed to be transformed to a term document matrix.

```
cleanCorpus <- function(corpus) {  
  clean_data <- tm_map(corpus, removePunctuation)  
  clean_data <- tm_map(clean_data, stripWhitespace)  
  clean_data <- tm_map(clean_data, tolower)  
  clean_data <- tm_map(clean_data, PlainTextDocument)  
  clean_data <- tm_map(clean_data, removeWords,  
stopwords("english"))  
  return(clean_data)  
}
```

This function creates a temporary corpus in a variable to store the cleaned corpus. This function makes use of the tm library in Rstudio.

. Setting up the environment for twitter sentiment analysis: In order to access twitter first you must have a twitter account with a valid phone number in order to get API tokens. Once the authentication is complete, it is possible to pull tweets. To further analyse we must make a dictionary of words to compare tweets to that will give us positive and negative sentiment. Once we have this we need a corpus of text to compare. To implement a score of sentiment based on the positive and negative words, a function must be created to allow this analysis. Our dictionary of positive and negative words is acquired from th hu

liu dictionary, these are well known repositories and are widely used in text mining, this will give a dictionary for us to match the positive and negative functions. We will create a score sentence function, a score sentiment function and a collect and score function. The score sentence function takes a sentence and analyses whether the words match to our positive and negative dictionaries and if there is a match their sentence is score accordingly. In the score sentiment function we are seeing if a sentence doesn't have a positive or negative word it is give a score of neutral. The collect and score function interacts with the Twitter API and pulls tweets based on a given handle, hashtag or keyword, the tweets are then analysed and scored accordingly to their sentiment. A data frame is then created with the collected tweets and scored sentiment for us to analyse. Once the twitter data was obtained and the sentiments were applied, it became a decision of which topics to analyse, to apply this scrip to the political domain a range of experiments were conducted.

```
collect.and.score <- function (handle, code, title, pos.words,
neg.words) {

  tweets = searchTwitter(handle, n=1500)
  text = lapply(tweets, function(t) t$text())

  score = score.sentiment(text, pos.words, neg.words)

  return (score)
}

#here we invoke the function above for each of our users
trump.scores = collect.and.score("@realDonaldTrump", "trump", "president",
pos.words, neg.words)
```

DATA TRANSFORMATION

One of the most effective was to transform data in text analytics is using a term document matrix, this is then process of turning your text based data into a numeric value. This is done by generating a matrix of the frequency of terms used in a corpus along with the speech it appears in. The first step in implementing the term document matrix is to create a function that will allow us to apply this code to multiple corpuses if necessary.


```

generateTDM <- function(pres,path){
  speech_dir <- sprintf("%s/%s",path,cand)
  speech_corpus <- Corpus(DirSource(directory = speech_dir, encoding= "UTF-8"))
  speech.cor.cleaned <- cleanCorpus(speech_cor)
  speech.tdm <- TermDocumentMatrix(speech.cor.cleaned)
  speech.tdm <- removeSparseTerms(s.tdm, 0.7)
  result <- list(name = pres, tdm = speech.tdm)
}

```

To generate the TDM in this Speech classification script it is necessary to pass the candidate name and the file path to locate the speech text files. In the function a speech directory is created the will concatenate the file path and the candidate name, meaning it will return the path with the speech text and then put the candidate name at the end of this to show which will point to the directory the speeches are found in. The next step is to create a corpus based on the path we just created. The text is encoded at UTF-8. Then the clean corpus function that was created previously is applied to our new corpus. Each term that appears across all the speeches is on one axis and the speech it appears in is on the other matrix, this process quantifies our text data. Next sparse terms are removed, sparse terms are not useful in classifying text data as the more frequent a term the less unique it becomes in the context of identifying the author. The result is returned in a list where the first name is the candidate and the second is the term document matrix. We then append the name of the candidate to the term document matrix. To do this a function is created. We pass the function the two tdms, firstly the tdm is converted into a numeric document matrix, also the row and column variables will be switches each speech will be a row and each term a column. In this step the tdm is changed into a data frame so it is easier to work with. Next cbind is used to input a column for the name of the candidate who said the speech. Then the name of the column will be 'target candidate'.

Next the tdms are stacked together, to this the rbind function from the dplyr package is used. This allows analysis of the two candidates as a whole and will allow us to create a KNN model using training and test samples of the current tdm to train a model to predict who wrote which speech. To start this process it is necessary to create sufficient training and test samples. It is good practice to use approximately 70% or more of your full dataset as a test dataset to feed the machine learning algorithm enough information for the prediction. The test dataset is then the remaining number of speeches left. Once we have these samples we can begin implementing the KNN algorithm.

.Sentiment Analysis: This step was implemented by using the three functions that were created earlier, once the functions are called then we use the keywords to collect the tweets and score them according to the sentiment.

DATA ALGORITHMS:

For the speech classification script, the model that was built used the K nearest neighbours algorithm.

```
knn.pred <-knn(tdm.stack.2[train.set, ], tdm.stack.2[test.set, ], tdm.pres[train.set])
```

We take out the name of the candidate and store it in its own variable, another variable will be created with everything except the target candidate. We then create the KNN model, the model is fed in the training data and the test data. Then we give it the names of the candidate for the training set to allow the algorithm to learn. After running this algorithm it is now possible to look at the accuracy of the model. One way of doing this is to build a confusion matrix, also we can get the accuracy score using the accuracy formula.

Politician v Non Politician Tweets:

This was a test to see if a classifier could be created to analyse if a tweet was of political deliberation by implementing machine-learning algorithms. To build this it was first necessary to choose the data we would use to conduct the study. It was necessary to first establish

sh what constitutes political and non political subjects. To do this a sample of 5000 tweets were taken from politicians to account for the political tweets and a sample of 5000 tweets were taken from twitter using keywords that do not relate to politics. Topics that were used were 'Kardashian, nba, movies and Rihanna'. The combined set of these tweets were stacked into a csv file to perform the analysis. First it was necessary to set up test and training data subsets, to do this the data needed to be randomly listed using the set.seed function. Then the data was split using 50% of the data for testing and 50% for training.

```
Train <- createDataPartition(y=pol_data$label, p=0.50, list = FALSE)
train_pol <- pol_data[Train,]
```

```
Testdata<- pol_data[!Train,]
```

```
Test <- createDataPartition(y=Testdata$label, p=0.50, list=FALSE)
```

The next step was to create a corpus from the train data, this corpus was cleaned using the same tm.map functions as before in the speech classification script. Once the corpus is cleaned a term document matrix was created from the training set. The matrix proved to be too large for sufficient analysis so another cleaning function needed to be applied, remove sparse terms. These takes out terms that will be no use in the classification. Once the text is clean we use decision trees and naïve bayes algorithm to attempt to classify the text.

```
#naivebayes
library(e1071)
bow_Naive_pol <- naiveBayes(y ~ ., data = train_data_pol)
summary(bow_Naive_pol)
testlpred = predict(bow_Naive_pol, newdata = test1_data_pol)

mmetric(testlpred, test1_data_pol[,1], c("ACC", "TPR", "PRECISION", "F1"))
```

Python Script:

The first step in creating this script was to download the PyCharm IDE and then use the pip function to install the necessary packages needed in this script using pip. The script was created following the learn Watson API tutorials from code academy. The data used in this script is gathered using the twitter api. To use this we must first create the api tokens in order to gain access to the twitter api, once we are in we gather the tweets from a users timeline. The twitter api gives a lot of meta data when streaming tweets so we must indicate that we only want text. We must ensure that the tweets were written in English and encoded in the UTF-8 format. This script will take two users timeline and apply the PI function to

compare the 2 users tweets and give an insight into their big five traits. To analyse these tweets with the personality insights feature, first it is necessary to create an account with IBM BlueMix, this allows us to access multiple features from the Watson api. Once we create an account we are give a token that allows us to access the api. We then create a function that called analyse that will take the twitter handle of the user and then apply the personality insights feature. Next a flatten function is created to separate the PI tree of personality, needs and values into just traits. The traits are then stored in a dictionary, a function is then created to compare these dictionaries against each other to allow us to see the distance between the two users traits, these are then stored in another dictionary. This produces results to give us insights to our politician's social media accounts.

MongoDB:

A mongoDb was used in this project to store live steaming tweets and perform up to date analysis, however the implementation of the database proved to not be as useful as hoped. The process of streaming live tweets and using it for real time analytics proved to be too much processing for this machine to handle. It caused shutdowns many times and for that reason it was not used to its full extent as I would have liked.

```
# Load the package
library(Twitter2Mongo)
# Specify the two collections to use for storing the tweets
setMongoDbCredentials(type = "hometimeline", collection =
"hometimeline", db = "twitter",
                      url = "mongodb://localhost")
setMongoDbCredentials(type = "tweetstream", collection = "tweetstream",
db = "twitter",
                      url = "mongodb://localhost")

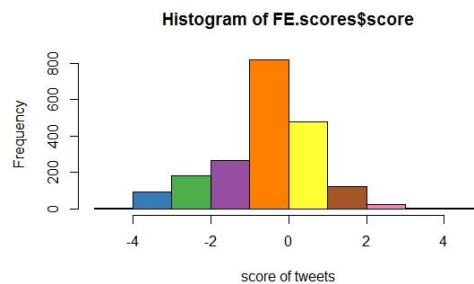
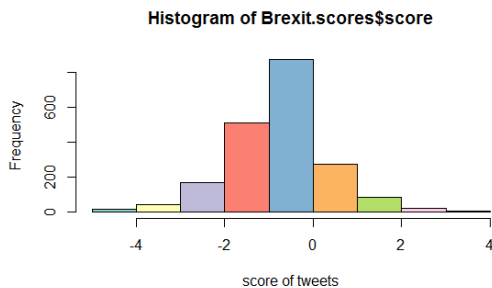
# start the streaming process with a specified query_url
startStreaming(60, "https://stream.twitter.com/1.1/statuses/filter.json?track=ww3")
```

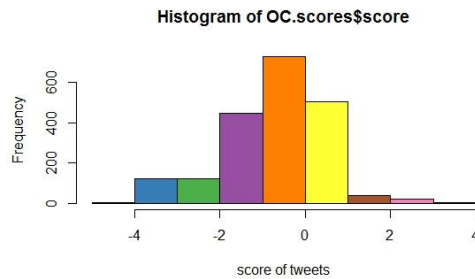
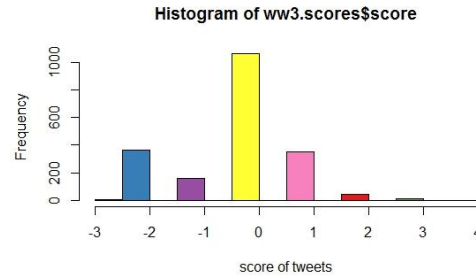
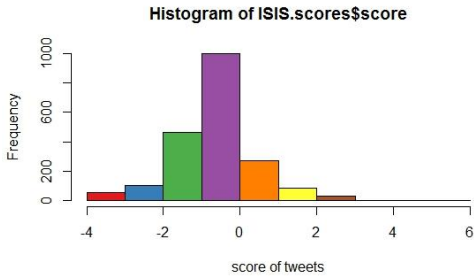
5 Results:

5.1 Experiment 1: Sentiment Analysis

In this experiment took samples of current topics that are currently in discussion around the world and gather the collective sentiment of each topic to assess what the overall tone towards these topics are. The test was comprised by collecting 2000 tweets from each topic. The topics chosen were “ww3”, “Brexit”, “ISIS”, “FrenchElection” and “Obamacare”. It was ensured that the tweets streamed be in English also they be adequately cleaned. To clean the tweets usernames, links and RT were removed to gain a more accurate representation of public opinion.

```
> summary(Brexit.scores$score)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-4.00  0.00   0.00  -0.08  0.00   2.00
> summary(ww3.scores$score)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-3.0000 -1.0000  0.0000 -0.2075  0.0000  4.0000
> summary(OC.scores$score)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-5.000  -1.000   0.000  -0.212   1.000   4.000
> summary(FE.scores$score)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-5.000  -1.000   0.000  -0.052   1.000   5.000
> summary(ISIS.scores$score)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-4.000  -1.000   0.000  -0.154   0.000   6.000
```





The results of the sentiment analysis show in general the mean public opinion is negative in emotional tone regarding political topics. In comparison, tweets that deal with elections and divide in political party opinions are more neutral than tweets that deal with war and terrorism.

5.2 Experiment 2: Speech classifier

In this experiment, the goal was to create a model that would allow us to feed a KNN model a list of speeches from different presidents and then the machine-learning algorithm would be able to predict which president said the speech. The test was initiated by taking 60 speeches, 30 by trump and 30 by Obama. The text was gathered from websites that make have the text of the speeches available for the public to see. Each individual speech was then stored in a folder with the name of the president who said the speech. The idea behind this experiment was to see if a model could identify the author of a text based on processing other text by the same author. The results show that the classifier model was perfectly accurate. This model could be applied to many other domains and could possibly be used as a base of a plagiarism detector.

Predictions	Actual	
	Obama	TRUMP
Obama	10	0
TRUMP	0	20

Accuracy = 100%

5.3 Experiment 3: Political v non-political tweets

This is the results of the two algorithms used to attempt to classify the text to with a political tweet or non-political. The results show that the accuracy of the naïve Bayes model performs slightly better although it's only slightly better than a 50/50 classifier which means that we have a weak prediction model. A reason for this may be the length of twitter text, at only 140 characters per tweet it is harder to classify the text compared to the speech classifier shown before which uses full speeches as a means of training data. As a further study it could be possible to use speeches of politicians and non politicians rather than the tweets to attempt to create a more accurate classifier

```
> mmetric(test1pred, test1_data_pol[,1],c("ACC","TPR","PRECISION","F1"))
      ACC      TPR1      TPR2 PRECISION1 PRECISION2      F11      F
12
50.63241  0.00000  50.63241  0.00000  100.00000  0.00000  67.2
2645
```

```
> mmetric(test1pred, test1_data_pol[,1],c("ACC","TPR","PRECISION","F1"))
)
      ACC      TPR1      TPR2 PRECISION1 PRECISION2      F11
F12
62.450593  0.000000  62.450593  0.000000  100.000000  0.000000  7
4.783951
```

5.4 Experiment 4: Personality insights

A python script was run to make use of the IBM personality insight function. This script allows to get the big 5 traits from two users and compare them. The results of the Python script return multiple personality facets that stem from the 5 traits, for the purpose of testing politicians we will use facets that apply to a political personality.

	Trump	Cameron	May	Trudeau	Clinton
Dutifulness	93	64	82	74	97
Emotionality	8	44	27	70	77
Orderliness	77	17	28	67	65
Intellect	96	89	89	82	87
Trust	74	90	97	98	57
Assertiveness	99	98	99	99	99

Once the data was collected from twitter and analysed using the Watson API, a sample group of five political figures were selected. The personality traits for each were displayed in the output, 6 traits were taken from each in order to create a statistical test based on this data. An ANOVA test was conducted to investigate if there is a statistically significant between the politicians and the personality traits. To set up this test the data is reported into excel. The test will be run using the data analysis tool pack.

$H_0: \mu_{\text{Trump}} = \mu_{\text{Cameron}} = \mu_{\text{May}} = \mu_{\text{Trudeau}} = \mu_{\text{Clinton}}$

The null hypothesis states that the mean values of the traits is equal across all of the politicians.

$H_1: \mu_{\text{Trump}} \neq \mu_{\text{Cameron}} \neq \mu_{\text{May}} \neq \mu_{\text{Trudeau}} \neq \mu_{\text{Clinton}}$

The alternate hypothesis states that the mean values is different in at least two or more politicians.

The test will be conducted at an alpha value of 0.05, at a confidence level of 95%.



Figure 10: Boxplot

This boxplot visual that displays the variance of each trait. The bold black bars show the mean value, the slim black bars represent the min/max values. We can report with 95% confidence that the majority of values lie within the white box.

Dutifulness	Emotionality	Orderliness	Intellect
Min. :64	Min. : 8.0	Min. :17.0	Min. :82.0
1st Qu.:74	1st Qu.:27.0	1st Qu.:28.0	1st Qu.:87.0
Median :82	Median :44.0	Median :65.0	Median :89.0
Mean :82	Mean :45.2	Mean :50.8	Mean :88.6
3rd Qu.:93	3rd Qu.:70.0	3rd Qu.:67.0	3rd Qu.:89.0
Max. :97	Max. :77.0	Max. :77.0	Max. :96.0
Trust	Assertiveness		
Min. :57.0	Min. :98.0		
1st Qu.:74.0	1st Qu.:99.0		
Median :90.0	Median :99.0		
Mean :83.2	Mean :98.8		
3rd Qu.:97.0	3rd Qu.:99.0		
Max. :98.0	Max. :99.0		

Now that we have our descriptive we can conduct the One way ANOVA to test our hypothesis.

Anova: Single Factor

SUMMARY

<i>Groups</i>	<i>Count</i>	<i>Sum</i>	<i>Average</i>	<i>Variance</i>
Trump	6	447	74.5	1166.7
Cameron	6	402	67	1002.4
May	6	422	70.33333333	1137.467
Trudeau	6	490	81.66666667	195.4667
Clinton	6	482	80.33333333	292.2667

ANOVA

<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>	<i>F crit</i>
Between Groups	951.866667	4	237.966667	0.313584	0.866159	2.75871047
Within Groups	18971.5	25	758.86			
Total	19923.3667	29				

$F(4) = 0.313, p = 0.866$

The results of the ANOVA give an F statistic of 0.313 at 4 degrees of freedom. The obtained p value is 0.866 which is greater than the set alpha value of 0.05 meaning we lack sufficient evidence to reject the null hypothesis. This means that there is no statistically significant difference between the means of the personality traits of the politicians.

Donald Trump personality analysis using Watson:

Openness:

- Imagination: 0.0678636999748
- Artistic Interests: 0.579190066366
- Intellect: 0.967554296326
- Emotionality 0.085682998745

- Adventurousness 0.872067736446
- Liberalism 0.7222062766

High scores indicating Trump is a person who is open to new experiences and likes to live freely. Very low imagination scores indicate a preference to facts than fantasy.

Extraversion

- Friendliness 0.837504562933
- Gregariousness 0.556578150068
- Assertiveness 0.990366122032
- Activity level 0.838303234966
- Cheerfulness 0.370930774039
- Excitement-seeking 0.106075429638

Low scores in excitement seeking indicates an adverse to thrill seeking, although high scores in other areas indicate an outspoken, active individual who is a well-spoken energetic socialiser.

Agreeableness

- Trust 0.754196306886
- Altruism 0.841474086546
- Morality 0.818317850428
- Cooperation 0.669028607987
- Modesty 0.35496311328
- Sympathy 0.984706515461

Low scores in modesty show that Trump may feel himself as a superior than others and seen as arrogant to others. High scores in other areas indicate a person who is interested in self-fulfilment also a frank and outright individual.

Conscientiousness:

- Self-efficacy 0.95133093881
- Orderliness 0.767594184423
- Dutifulness 0.93519513974
- Achievement striving 0.962993934103
- Self-discipline 0.978174917753
- Cautiousness 0.978909164751

Very high scores show that Trump is highly motivated, disciplined, responsible and likely to plan ahead.

Neuroticism:

- Anger 0.0303128957249
- Anxiety 0.0297506165177
- Vulnerability 0.00789058986195
- Immoderation 0.0189333606472
- Depression 0.0410377681224
- Self-consciousness 0.0170243577832

Very low scores indicate a emotionally stable, calm and collected individual.

Visuals and other results



Figure 11: Word cloud of Inaugural Speeches

This is a word cloud of inaugural speeches dating back to George Washington with Donald Trump's speeches removed, this word cloud is an indication of the frequent terms used by past presidents, in comparison below we see a word cloud of tweets by Donald trump. These visuals are a quick insight into the difference into speeches and social media as a means of addressing the public.

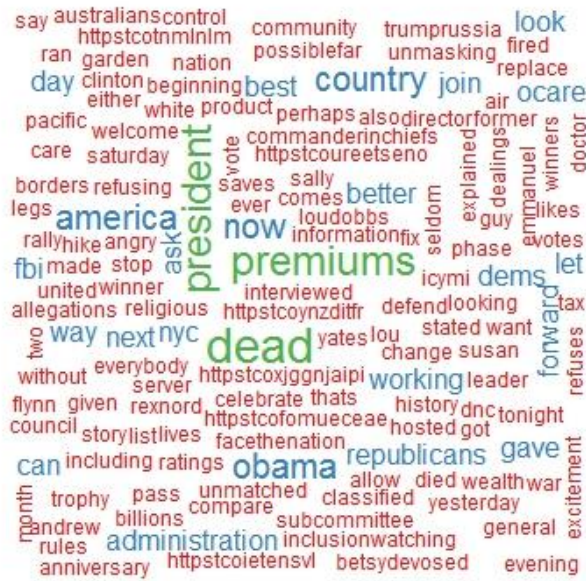


Figure 12: Word Cloud of Donald Trump Text

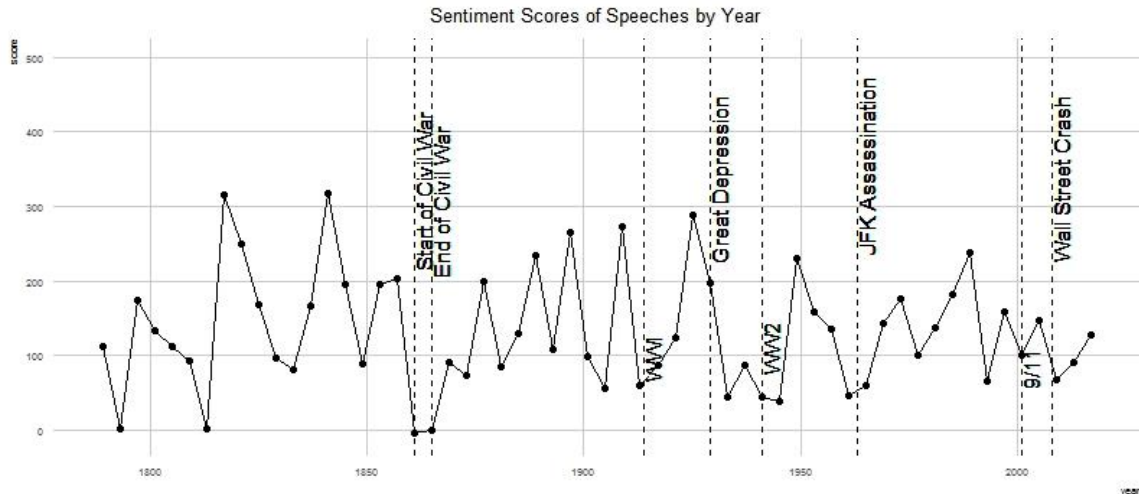


Figure 13: Sentiment Graph of Speeches by year

This graphic is based off of the inaugural speeches data set acquired from kaggle, The source code for creating graph was taken from a Kernel by Pradeep Adohskaja. The graph displays the time series sentiment of inaugural speeches with labels from key event in American history. This diagram fits in nicely as a comparison to the sentiment analysis of recent topics.

5.5 Graphical User Interface (GUI) Layout

This project is designed for statistical analysis so the GUI element of the project is not of much importance although there is a need for data visualisation which may make use user interface. In further study, a web app will be developed in R Shiny that will allow the user to look at live streaming sentiment analysis of current topics based on ley words. The graph should update in real time, the user interface comes from allowing the user to change the keywords in the search bar to allow them to look at the topics in which they would like to see analysed. The web app is strictly for visualisation purposes only. Early stages of app, source code was gathered from github tutorials. This app is in development and will be modified for showcase.

RStudio

File Edit Code View Plots

Shiny

http://127.0.0.1:6455 Open in Browser Publish

Project: (None)

Sentiment Analysis

Search on Social Media:

#donaldrump

Compare

Plot Sentiments

Language:

All

#donaldrump

Sentiment Chart

You can click on each dot to read the social media post.

Search

● #donaldrump

Popularity

Date Time

Trends

#donaldrump

5.0

2.5

2 4

Console

```

C:/Users/keith/Desktop/R
sew_dataj , wroch -399
"-2.05778146597692," bott
59994951585,"bottom":32
pping":{})),"trends":{"
map":[{"domain":{"left":
ght":599,"bottom":399,"
[]

```

10/05/2017 02:22

6 Testing

Through each phase of my system evolution I will implement different test cases in each phase to ensure each procedure is fit for use. A test case will be created for each of the requirements to ensure that the system is up to standard. These cases will include white box testing, black box testing and end user testing.

Test ID	Test_Requirement1
Purpose of Test	To Ensure that: It is possible to engage with the twitter API It is possible store tweets in datastore
Test Environment	The test environment is as follows: Client Hardware: HP pavilion laptop running windows 10 and R Studio latest edition Test was to store 200 tweets from Theresa Mays timeline into .csv file
Test Steps	From R studio the tester should: Ensure that the working directory is set to the folder they wish to store the data Run script
Expected Result	On completing the above steps, the twitter API should be activated and it should return 200 tweets from the selected handle and store them in a variable. This variable should then be read into a csv file that will be stored in a selected directory.
Actual result	The csv file was created with 167 tweets stored.
Suggested action	The dataset was short because of the limitations of the twitter APIs limit of only pulling tweets from up to two weeks old
Resolution	Pull tweets again two weeks later for more available tweets.

Test ID	Test_Requirent12
Purpose of Test	To Ensure that: It is possible to read a file into R It is possible to clean the text in the file
Test Environment	The test environment is as follows: Client Hardware: HP pavilion laptop running windows 10 and R Studio latest edition Test data was a csv with 200 tweets
Test Steps	From R studio the tester should: Ensure that data set "TweetTest.txt" is in working directory Run script Small test from line 1-58
Expected Result	On completing the above steps, the global environment in RStudio should have cleaned with data. There should be a cleaned Tweets variable created with text cleaned to lower, no punctuation, stop words or numbers.
Actual result	The variable was created and text was cleaned.
Suggested action	N/A
Resolution	N/a

Test ID	Test_Requirent3
Purpose of Test	To Ensure that: TDM is generated for a text corpus
Test Environment	The test environment is as follows: Client Hardware: HP pavilion laptop running windows 10 and R Studio latest edition Test data was csv file with tweets
Test Steps	From R studio the tester should: Ensure that data set .csv file is in working directory Run script line 15 - 25
Expected Result	A Term Document Matrix should be generated and stored in global environment
Actual result	Returned Term Document matrix with 0 terms
Suggested action	Remove line 'Remove sparse terms' or set a 0.99 percentage for the remove sparse terms command
Resolution	TDM was generated with 175 terms

Test ID	Test_Requirent4
Purpose of Test	To Ensure that: Algorithm returns accuracy measure
Test Environment	The test environment is as follows: Client Hardware: HP pavilion laptop running windows 10 and R Studio latest edition Test data was two file directories of text files
Test Steps	From R studio the tester should: Ensure that working directory is set to desktop/R/text Ensure text files are located in the in the folders Run test_KNN script from start to finish
Expected Result	Accuracy measure is returned
Actual result	Accuracy measure was returned with 100% rate
Suggested action	n/a
Resolution	n/a

Test ID	Test_Requirent5
Purpose of Test	To Ensure that: Word cloud is formed on the selected text
Test Environment	The test environment is as follows: Client Hardware: HP pavilion laptop running windows 10 and R Studio latest edition Test data was csv file with president speeches
Test Steps	From R studio the tester should: Ensure that data set inaugural_speeches.csv file is in working directory Run script medium test from line beginning to end
Expected Result	A word cloud should be formed with the 20 most frequent words
Actual result	Word cloud is generated with 20 most frequent words
Suggested action	n/a
Resolution	n/a

7 Conclusion, Research and Further development

Further Development:

The projects initial plan was to explore the area of text analytics in the political domain while also looking at interesting trend in what politicians are saying through their speeches and social media and see how people are reacting to current political topics. In further development, the algorithms applied in this project can be used as a start for an authorship detection program in which we can train a model to test if the text is written by the said author or if it's a case of plagiarism. With more resources and computational power it'd be possible to live stream tweets online and gain a constant sentiment analysis with live updates. I would like to continue the development of this app and try to complete it as my own personal project as I think it would be a good development tool and also an area in which I am interested.

The KDD framework allowed me to apply a very structured approach to this data project; I feel this played a vital role in the project lifecycle. The KDD ensured efficient data analysis principles were adhered to in completion of this project. The application of this method allowed the project to be planned effectively and laid out a step by step guide of how to get from the initial planning through the application pipeline of pre processing text, transforming it into numeric values then machine learning applying models to gain informative results.

In developing this project through the year it has become a great source of learning as it has opened me up to us of text base analysis along with broadening my scope as a data analyst. Recently I have had a job interview for Bank of Ireland in which I was asked in detail about my final year project and the inspiration behind it, in which the interviewers were very impressed. I have succeeded in securing the job offer off the back of the NCI course load and my final year project, which is a testament to the base content of this course.

8 References

8.1 References

- En.wikipedia.org. (2016). *Software testing*. [online] Available at: https://en.wikipedia.org/wiki/Software_testing [Accessed 11 Dec. 2016].
- Ganesan, K. (2016). *What is Text Similarity?*. [online] Text-analytics101.rxnlp.com. Available at: <http://text-analytics101.rxnlp.com/2015/11/understanding-text-similarity.html> [Accessed 11 Dec. 2016].
- Gurusamy, V. (2016). *Preprocessing Techniques for Text Mining*. [online] ResearchGate. Available at: https://www.researchgate.net/publication/273127322_Preprocessing_Techniques_for_Text_Mining [Accessed 8 Dec. 2016].
- Liwc.wpengine.com. (2016). *LIWC | Linguistic Inquiry and Word Count*. [online] Available at: <https://liwc.wpengine.com/> [Accessed 11 Dec. 2016].
- Nlp.stanford.edu. (2016). *The Stanford Natural Language Processing Group*. [online] Available at: <http://nlp.stanford.edu/> [Accessed 11 Dec. 2016].
- Nltk.org. (2016). *Natural Language Toolkit — NLTK 3.0 documentation*. [online] Available at: <http://www.nltk.org/> [Accessed 11 Dec. 2016].

9 References

- Anon, (2017). *Psychological and Personality Profiles of Political Extremists*. [online] Available at: <https://arxiv.org/ftp/arxiv/papers/1704/1704.00119.pdf> [Accessed 5 May 2017].
- Citeseerx.ist.psu.edu. (2017). *CiteSeerX — Unknown file type*. [online] Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.153.9654&rep=rep1&type=pdf>
https://www.researchgate.net/publication/273127322_Preprocessing_Techniques_for_Text_Mining [Accessed 10 May 2017].
- Codecademy. (2017). *Learn the Watson API*. [online] Available at: <https://www.codecademy.com/learn/ibm-watson> [Accessed 10 May 2017].

- GitHub. (2017). *ampaho/TweetyFeely*. [online] Available at: <https://github.com/ampaho/TweetyFeely> [Accessed 10 May 2017].
- GitHub. (2017). *ProjectTw/TwitteR2Mongo*. [online] Available at: <https://github.com/ProjectTw/TwitteR2Mongo> [Accessed 10 May 2017].
- Go, A., Bhayani, R. and Huang, L. (2017). *Twitter Sentiment Classification using Distant Supervision*. [online] amazonaws. Available at: http://s3.amazonaws.com/academia.edu.documents/34632156/Twitter_Sentiment_Classification_using_Distant_Supervision.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1494382123&Signature=kccEQaA%2Fw3OTxu6UghLLVZLoSUY%3D&response-content-disposition=inline%3B%20filename%3DTwitter_Sentiment_Classification_using_D.pdf [Accessed 10 May 2017].
- learn data science. (2017). *Demystifying Text Analytics part1 — Preparing Document and Term Data for Text Mining in R*. [online] Available at: <https://blog.exploratory.io/demystifying-text-analytics-preparing-document-and-term-data-for-text-mining-in-r-4f858feb4b77> [Accessed 10 May 2017].
- Psychometric-success.com. (2017). *The Big 5 Aspects of Personality*. [online] Available at: <http://www.psychometric-success.com/personality-tests/personality-tests-big-5-aspects.htm> [Accessed 10 May 2017].
- Ohri (2017). *Twitter analysis by Kaify Rais*. [online] Slideshare.net. Available at: <https://www.slideshare.net/ajayohri/twitter-analysis-by-kaify-rais> [Accessed 10 May 2017].
- R and Beyond. (2017). *Scraping the web for Presidential Inaugural Addresses using #rvest*. [online] Available at: <https://pradeepadhokshaja.wordpress.com/2017/03/31/scraping-the-web-for-presidential-inaugural-addresses-using-rvest/> [Accessed 10 May 2017].
- Rstudio.github.io. (2017). *Tutorial: Building 'Shiny' Applications with R*. [online] Available at: <http://rstudio.github.io/shiny/tutorial/> [Accessed 10 May 2017].
- Tumasjan, A., Sprenger, T., Sandner, P. and Welpe, I. (n.d.). Election Forecasts with Twitter - How 140 Characters Reflect the Political Landscape. *SSRN Electronic Journal*.

YouTube. (2017). *How to Build a Text Mining, Machine Learning Document Classification System in R!*. [online] Available at: <https://www.youtube.com/watch?v=j1V2McKbkLo> [Accessed 10 May 2017].

YouTube. (2017). *Rvest Tutorial*. [online] Available at: <https://www.youtube.com/watch?v=gSbuwYdNYLM> [Accessed 10 May 2017].

YouTube. (2017). *Text Mining: Sentiment Analysis with Bag of Words*. [online] Available at: <https://www.youtube.com/watch?v=M1rR8V1-UpA> [Accessed 10 May 2017].

10 Appendix

10.1 Project Proposal

Project Proposal

TITLE

NLP: An analysis of text through multiple platforms.

Student Name: Keith Carpenter

Number: x13525447

email address: Keith.Carpenter@student.ncirl.ie

Degree Programme Name e.g. BSc (Hons) in Computing

Specialisation: Data Analytics

Date : 19/10/2016

11 Objectives

The objective of this project is to gather information on a chosen individual and use as much text based information as possible to identify trends, similarities and distance in their words from different modes of text.

The project will evolve in in 3 main steps.

1: Choose person/people with a variety of text based publications eg. Bob Dylan. Pull data from the web including social media, lyrics, books, poems, speeches etc. Parse and clean data to be entered into a database.

2: Use NLP Toolkit to pre-process and break down text to be analysed. Run data through a NLP tool eg. LIWC to gain information, identify trends and valuable information.

3. Use statistics and visualisation tools to analyse and display data.

12 Background

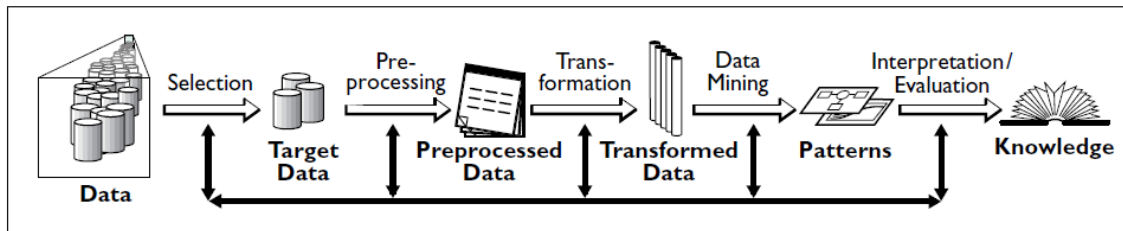
The idea of this project stemmed from interest in the area of text analysis. The purpose of the project was to explore in depth the area of NLP. It was planned to make a project that would take multiple forms of text from the same author and analyse to gauge a sense of how they portray themselves through different mediums of text. As the idea developed it became about the questions that could be applied to the project and how much information you could derive from these questions. The plan is to create a software system that will extract and analyse this data through NLP to allow for statistics to be applied to answer such questions.

13 Technical Approach

Brief description of the approach to be followed (Max. 1 Page), Research, literature review, requirements capture, implementation etc...

KDD data model

Figure 1. Overview of the steps constituting the KDD process



Data Selection:

It will be necessary to pick a variant with sufficient text to be able to analyse for relevant results. We will look at politicians, authors, celebrities to get a valid text corpus to analyse

Data Pre Processing:

In the pre-processing of data we must use techniques such as tokenising, clustering, stemming and chunking text in order to give us something we can use to analyse efficiently

Transforming Data:

The data will be used to run statistical procedures such as plotting into graphs and charts to allow use to look for trends.

Data Mining:

We will use artificial intelligence, machine learning, statistics, and database systems to look for trends and patterns in this data.

Data Interpretation:

Reporting of valid information and findings

14 Special resources required

If applicable, e.g., books, hardware, etc.

In this project information will be obtained from as many sources necessary to be able to analyse text correctly and efficiently. In order to this as much resources will be used as possible such as books, poems, lyrics etc.

Resource books from library.

Natural Language Processing in Python.

15 Project Plan

Gantt chart using Microsoft Project with details on implementation steps and timelines

Task Mode	Task Name	Duration	Start	Finish	Predecessors
Manually Scheduled	Project Proposal & Project Plan Documents	25 days	Mon 19/09/16	Fri 21/10/16	
Auto Scheduled	Project Brainstorming	8 days	Mon 19/09/16	Wed 28/09/16	
Auto Scheduled	Project Pitch	1 day	Wed 05/10/16	Wed 05/10/16	
Manually Scheduled	Journal Entry	1 day	Fri 07/10/16	Fri 07/10/16	
Auto Scheduled	Project Proposal	5 days	Thu 06/10/16	Wed 12/10/16	3
Manually Scheduled	Documentation Review & Upload	1 day	Fri 21/10/16	Fri 21/10/16	
Manually Scheduled	Requirement Specification	17 days	Thu 20/10/16	Fri 11/11/16	
Auto Scheduled	Begin searching for text documents on web	10 days	Fri 21/10/16	Thu 03/11/16	

Auto Scheduled	Requirement Specification Document/pull data from web	14 days	Mon 24/10/16	Thu 10/11/16	
Manually Scheduled	Documentation Review & Upload	1 day	Fri 11/11/16	Fri 11/11/16	
Auto Scheduled	Journal Entry	1 day	Thu 03/11/16	Thu 03/11/16	
Manually Scheduled	Project Analysis & Design	21 days	Sat 12/11/16	Fri 09/12/16	
Manually Scheduled	parse data	5 days	Mon 14/11/16	Fri 18/11/16	
Manually Scheduled	put in mysql database	7 days	Fri 18/11/16	Sun 27/11/16	
Manually Scheduled	Organise dataset for text analysis	5 days	Sat 19/11/16	Thu 24/11/16	
Manually Scheduled	Run unit tests	5 days	Thu 24/11/16	Wed 30/11/16	
Manually Scheduled	Project Prototype	3 days	Sat 01/10/16	Tue 04/10/16	
Manually Scheduled	journal Entry	1 day	Fri 09/12/16	Fri 09/12/16	
Manually Scheduled	Mid-Point Presentation	1 day	Fri 16/12/16	Fri 16/12/16	
Manually Scheduled	Post Mid-Point	23 days	Sat 17/12/16	Tue 17/01/17	
Manually Scheduled	Review Feedback from panel and makes changes	1 day	Sat 17/12/16	Sat 17/12/16	
Manually Scheduled	obtain more data from web and begin automating software	5 days	Sun 18/12/16	Thu 22/12/16	

Manually Scheduled	Christmas Break	4 days	Thu 22/12/16	Tue 27/12/16	
Manually Scheduled	journal Entry	1 day	Wed 04/01/17	Wed 04/01/17	
Manually Scheduled	Study Break for exams	15 days	Wed 28/12/16	Tue 17/01/17	
Manually Scheduled	Project Final Documentation & Code	90 days	Tue 17/01/17	Mon 22/05/17	
Manually Scheduled	Model Finalization	27 days	Tue 17/01/17	Wed 22/02/17	
Manually Scheduled	Finish of mining data	7 days	Tue 17/01/17	Wed 25/01/17	
Manually Scheduled	Discuss with supervisor final steps	1 day	Wed 25/01/17	Wed 25/01/17	
Manually Scheduled	Begin data mining and running different test and algorithms	15 days	Thu 26/01/17	Wed 15/02/17	
Manually Scheduled	visualization of data	5 days	Thu 16/02/17	Wed 22/02/17	
Manually Scheduled	Testing Phase	6 days	Thu 23/02/17	Thu 02/03/17	
Manually Scheduled	Collect testing feedback	3 days	Thu 23/02/17	Mon 27/02/17	
Manually Scheduled	Address any newly flagged issues	3 days	Mon 27/02/17	Wed 01/03/17	
Manually Scheduled	Final testing phase	3 days	Thu 02/03/17	Mon 06/03/17	
Manually Scheduled	Technical Report, Findings	49 days	Fri 10/03/17	Wed 17/05/17	
Manually Scheduled	Review	1 day	Wed 17/05/17	Wed 17/05/17	

Manually Scheduled	Upload Documentation	Final	1 day	Wed 17/05/17	Wed 17/05/17	
---------------------------	-----------------------------	--------------	--------------	---------------------	---------------------	--

16 Technical Details

Implementation language and principal libraries

REST API to pull data from websites. Data will be parsed and put into a MySQL database. Then a NLP process will be implemented (python NLTK, LIWC) resulting data will be analysed using SPSS or R. The stats will then be visualised using Tableau.

Python

R

Python NLTK

rNLP

LIWC

17 Evaluation

Describe how you will evaluate the system with real technical data using system tests, integration tests etc. In addition, where possible describe how you will evaluate the system with an end user.

After extracting data from the web I will be implementing unit tests to ensure the quality of the code.

The software process will be automated to allow for easier use of app.

User testing will be implemented to ensure this project can be used by end user.

Signature of student and date: Keith Carpenter 19/10/16

Task Mode	Task Name	Duration	Start	Finish	Predecessors
Manually Scheduled	Project Proposal & Project Plan Documents	25 days	Mon 19/09/16	Fri 21/10/16	
Auto Scheduled	Project Brainstorming	8 days	Mon 19/09/16	Wed 28/09/16	
Auto Scheduled	Project Pitch	1 day	Wed 05/10/16	Wed 05/10/16	
Manually Scheduled	Journal Entry	1 day	Fri 07/10/16	Fri 07/10/16	
Auto Scheduled	Project Proposal	5 days	Thu 06/10/16	Wed 12/10/16	3
Manually Scheduled	Documentation Review & Upload	1 day	Fri 21/10/16	Fri 21/10/16	
Manually Scheduled	Requirement Specification	17 days	Thu 20/10/16	Fri 11/11/16	
Auto Scheduled	Begin searching for text documents on web	10 days	Fri 21/10/16	Thu 03/11/16	
Auto Scheduled	Requirement Specification Document/pull data from web	14 days	Mon 24/10/16	Thu 10/11/16	
Manually Scheduled	Documentation Review & Upload	1 day	Fri 11/11/16	Fri 11/11/16	
Auto Scheduled	Journal Entry	1 day	Thu 03/11/16	Thu 03/11/16	

Manually Scheduled	Project Analysis & Design	21 days	Sat 12/11/16	Fri 09/12/16	
Manually Scheduled	parse data	5 days	Mon 14/11/16	Fri 18/11/16	
Manually Scheduled	put in mysql database	7 days	Fri 18/11/16	Sun 27/11/16	
Manually Scheduled	Organise dataset for text analysis	5 days	Sat 19/11/16	Thu 24/11/16	
Manually Scheduled	Run unit tests	5 days	Thu 24/11/16	Wed 30/11/16	
Manually Scheduled	Project Prototype	3 days	Sat 01/10/16	Tue 04/10/16	
Manually Scheduled	journal Entry	1 day	Fri 09/12/16	Fri 09/12/16	
Manually Scheduled	Mid-Point Presentation	1 day	Fri 16/12/16	Fri 16/12/16	
Manually Scheduled	Post Mid-Point	23 days	Sat 17/12/16	Tue 17/01/17	
Manually Scheduled	Review Feedback from panel and makes changes	1 day	Sat 17/12/16	Sat 17/12/16	
Manually Scheduled	obtain more data from web and begin automating software	5 days	Sun 18/12/16	Thu 22/12/16	
Manually Scheduled	Christmas Break	4 days	Thu 22/12/16	Tue 27/12/16	
Manually Scheduled	journal Entry	1 day	Wed 04/01/17	Wed 04/01/17	
Manually Scheduled	Study Break for exams	15 days	Wed 28/12/16	Tue 17/01/17	
Manually Scheduled	Project Final Documentation & Code	90 days	Tue 17/01/17	Mon 22/05/17	

Manually Scheduled	Model Finalization	27 days	Tue 17/01/17	Wed 22/02/17	
Manually Scheduled	Finish of mining data	7 days	Tue 17/01/17	Wed 25/01/17	
Manually Scheduled	Discuss with supervisor final steps	1 day	Wed 25/01/17	Wed 25/01/17	
Manually Scheduled	Begin data mining and running different test and algorithms	15 days	Thu 26/01/17	Wed 15/02/17	
Manually Scheduled	visualization of data	5 days	Thu 16/02/17	Wed 22/02/17	
Manually Scheduled	Testing Phase	6 days	Thu 23/02/17	Thu 02/03/17	
Manually Scheduled	Collect testing feedback	3 days	Thu 23/02/17	Mon 27/02/17	
Manually Scheduled	Address any newly flagged issues	3 days	Mon 27/02/17	Wed 01/03/17	
Manually Scheduled	Final testing phase	3 days	Thu 02/03/17	Mon 06/03/17	
Manually Scheduled	Technical Report, Findings	49 days	Fri 10/03/17	Wed 17/05/17	
Manually Scheduled	Review	1 day	Wed 17/05/17	Wed 17/05/17	
Manually Scheduled	Upload Final Documentation	1 day	Wed 17/05/17	Wed 17/05/17	

Monthly Journals

Reflective Journal

Student name: Keith Carpenter

Programme (e.g., BSc in Computing): BSc in Computing

Month: September

My Achievements

This month, I was able to secure an approved project idea.

Start to think of a way to plan my project and the technologies to be used

My Reflection

During this month I had not thought of a topic, the project pitch was scheduled and I didn't know what I wanted to do as I hadn't done many classes specific to the data analysis stream. In one of my lectures the lecturer spoke about sentiment analysis and text analytics and it caught my interest. I had no solid idea of my project idea so I went into my pitch and told the lecturers I wished to do an Natural Language Processing type of application. They were really helpful and Michael and Simon suggested I do a cross platform NLP project to research authorship of content. When I left I began to do research on this topic and have contacted Simon in hopes to talk more about this topic.

Supervisor Meetings

No supervisor as of yet but have met with Simon Caton and discussed the best paths of action for this project.

Reflective Journal

Student name: Keith Carpenter

Programme (e.g., BSc in Computing): BSc in Computing

Month: October

My Achievements

This month, I started thinking about my project idea and how im going to approach it.

I started to research python nlp toolkits and how to effectively examine authorship in documents.

My Reflection

During this month I got to start to visualise how I'm going to start working on my project and researching the necessary techniques such as text clustering as I have never done them in class. Due to overwhelming projects in other modules I did not start working on a prototype but start looking at my requirements and what needs to be done and how I am going to achieve these goals.

Supervisor Meetings

Met once with Keith Maycock

Student name: Keith Carpenter

Programme (e.g., BSc in Computing): Data analytics

Month: November

Upload one journal every month. Expected word count 300 words (of you own words).

My Achievements

This month, I was able to start looking at what is needed for my mid point presentation. I worked on my requirement specification and also looked into a

framework for to begin building my prototype for the mid point. I will be creating a sentiment analysis tool using the R programming language.

My Reflection

I felt, it worked well to take large strides in my coursework and am beginning to get a feel for data analytics as a whole. I am focused on the tasks ahead and am looking forward to completing this semester on a high.

Intended Changes

Next month, I will try to use my time to complete the next stages of my report and begin to start coding my project.

Supervisor Meetings

Date of the meeting: 7/11/2016, 14/11/2016

Discussed different ways to perform text analytics. Keith advised to use the Stanford NLP

Student name: Keith Carpenter

Programme (e.g., BSc in Computing): Data analytics

Month: December

Upload one journal every month. Expected word count 300 words (of your own words).

My Achievements

This month, I was able to finish off semester one along with all of my projects. Currently studying for the exams and was able to complete my final draft for the mid point presentation, I have met with Keith on 2 occasions to discuss progress. This month has been very stressful as many deadlines were all due together but I feel I have performed well.

My contributions to the projects included building a working sentiment analysis prototype along with finishing off my tech report upload.

My Reflection

I felt, it worked well to finish this semester in a strong way while juggling multiple projects while studying for CAs and exams.

However, I was not successful in gathering as much time as I would have liked to work on my project as I would've liked to flesh out the tech report.

Intended Changes

Next month, I will try to use my time in studying material needed to progress my project.

Supervisor Meetings

Date of Meeting: 2/12/2017, 9/2/2017

Items discussed: What is needed for the mid point.

Student name: Keith Carpenter

Program (e.g., BSc in Computing): Data analytics

Month: January

Upload one journal every month. Expected word count 300 words (of you own words).

My Achievements

This month, I was able to complete the first semester exams and also got my marks for my midpoint presentation. I took into account what Keith said about my project direction and began to look into different ways to approach the sentiment analysis I wanted to do as he said it was too broad to to this and similarity of texts along with authorship attribution.

My Reflection

I felt, it worked well to finish the exams while researching different parts for my project. I need to focus my attention on sentiment analysis.

However, I was not successful in getting a meeting with Keith to discuss further plans. I will try to take further steps in developing my sentiment analysis tool

Intended Changes

Next month, I will try to use my time in studying material needed to progress my project.

Supervisor Meetings

Keith has been busy and was not able to meet

Student name: Keith Carpenter

Programme (e.g., BSc in Computing): Data Analytics

Month: February

Upload one journal every month. Expected word count 300 words (of you own words).

My Achievements

This month, I was able to get some help from my new supervisor Frances on the direction my project should be heading in, after I got assigned Frances after Keith was pronounced absent for the duration of the semester I went immediately to meet with her and tell her my situation with the lack of supervisor interaction and that I had no feedback. Frances got the report that Keith had wrote an my feedback forms. However I have decided to stick with my original project idea and began to work towards this. I contributed to my project by working on the authorship attribution problem, to do this I'm using a classification machine learning algorithm.

My Reflection

I felt, it worked well to make a decisive decision on my project, to stick with what I intended to do although at first it seemed to much. I will proceed working on my three main requierements this month.

Intended Changes

Next month, I will try to free up time to dedicate to the coding of the project along with finish up the tech report.

I realised that I need to focus on the wider aspects of text analytics to achieve my goals rather than focus on just sentient analysis.

Supervisor Meetings

Date of Meeting: 2/20/2017

Items discussed: Feedback from mid point, next steps, tech report

Action Items: Implement a politic domain to my project. Focus on gathering data to analyse. Work on usability in tech report.

Student name: Keith Carpenter

Programme (e.g., BSc in Computing): Data Analytics

Month: March

Upload one journal every month. Expected word count 300 words (of you own words).

My Achievements

This month, I was able to meet with Frances before the final exams, we discussed the importance of continuing to work on the project. I was able to implement a mongoDB and analyse live streaming tweets as well as build a classifier using the naïve Bayes model.

My Reflection

I felt, although the final year exams were looming I got some vital work done on my project that will stand going into the final months. I have implemented another bit of functionality to my project which I feel will allow me to expand on this and keep the momentum going. Meeting with Frances was a great success this month as it kept me focused going into the final hurdle.

Intended Changes

Next month, I will finish the final year exams and begin to focus my full attention to my project. There has been a lack of attention to the project documentation and I will need to put more effort into this. Also I need to finish off my profile for the project showcase and prepare for this by making a poster.

Supervisor Meetings

Date of Meeting: 22/03/2017

Items discussed: Keeping focused on the project as well as exams

Action Items: Work on implementing more algorithms in my project as well as try to implement a statistical test. Work on finding a way to present results at presentation.