Appendix A. Software Requirements Specification

Overview

Product Functions

The System must be able to collect Tweets on the topic of “Britain leaving the European Union”.

The System must be able to semantically classify tweets that is has received.

The System must be able to visually display the semantic views of the public on the topic of “Britain leaving the European Union”.

The System must collect, classify and visualize the tweets within sufficient real time.

The System must be autonomous in it’s operations.

User Classes and Characteristics

Operator: An Individual who has a reasonable technical expertise and wants to interface with the System directly via command line tools and code itself.

General User: An individual who wishes to inspect the visual analysis that the system computes. This user is encompasses all the possible characteristics of a miscellaneous member of the general public wishing to engage with the system.

Operating Environment

The System will operate on a remote virtual private server with an operating system of a Ubuntu Linux Distribution.

The System will require a data store for permanent and temporary storage of data with the data store being maintainable, scalable and available at all times.

Design and Implementation Constraints

There are a number of limiting factors that will constrain the Design and Implementation which are highlighted as follows:

External Storage Size: due to the system operating remotely the data store will also be remote and thus the amount of data stored will be a consideration.

RAM: the system may be operating on large amounts of data and thus must be conservative towards RAM usage at any one time.

Execution Time: the system operations must occur within sufficient real time.

User Documentation

The System will include an in detail user manual for an **operator** on how to run additional System commands.

It will not be required to provide a user document on the visual component of the System as it will be designed in an intuitive manner with some explanation on the usage of the website included within the site itself.

Dependencies

The systems main dependency is Twitter, the system source it's from Twitter and thus the system, its successful delivery and continuous operation will require a reliable interface with Twitter itself.

External Interface Requirements

Interfaces

The System will contain a command line interface and a visual interface for which the operator and general user will interact with respectively.

Command Line Interface

The System will provide a command line interface for every component within the system. Each component will have at least one runnable command e.g. for tweet collection component there will be commands for fetching historic tweets and fetching real time tweets.

The main requirements of the set of these commands is that they must ensure user can view the progress of the command and that the user is presented with error messages where incorrect usage has occurred.

Progress

The execution time of commands may vary dependent on command and parameters given thus the system's operational commands must provide the user with a visual indication of progress.

Errors

Due to the commands being manually run the system will provide means of prompting the user with the correct parameter inputs i.e. a -h option for each command that informs the user of the required and optional parameters. The System should also inform the user of incorrect usage with commands.

Visual Website Interface

The System's main user interface will be the visual website interface which will display all the tweets collected and the analysis conducted.

Home View

This is the view in which new users will access the website from, the home page will introduce the key concepts of the site via text i.e. that the site is on the display of sentiment analysis and the site is looking at the discussion of "Britain leaving the European Union" via the forum of Twitter. The home page will also include the tweets for which it has most recently collected, the main navigation overlay to traverse the site and a button prompt to encourage users to read more on what the site is about.

About View

This is the view in which users will access when wanting to read more into how the site conducts it's analysis. The about page will clarify how the analysis was conducted from a high level and what important API's it uses in a tiled display fashion.

Analysis View

This is the view in which users will access when wanting to view the analysis conducted by the site, the analysis page will allow users to view analysis from the different classifications the system offers and give explanation behind each of the classifiers.

Classifier Analysis View

This is the view in which a user has selected a specific classifier they wish to view analysis for and are now presented with the analysis results. This view will display all the classifications for a given classifier and is viewable by time unit e.g. day/week/month/year. The view will allow for the navigation by time unit using a time switcher group of buttons with a button for each of the aforementioned time units. The time switcher will also have forward and backward buttons to allow a user to go forward/backward a time unit for instance forward by one day would then present the user with the analysis of the next day after the currently selected one. Each time unit that is made up of a lower granularity time unit that is provided by the system i.e. a week is composed of days, will provide the analysis of it's composite time units and provide buttons to view the analysis of each of them.

Hardware Interfaces

The Systems visual website component will be developed using a Mobile First approach and will therefore support all common device types from small constrained screen size devices such as mobile up to larger desktops.

Software Interfaces

¡Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.¿

Communications Interfaces

HTTP: The System will require the use of HTTP connections for the consumption and exposure of data.

SCP: The System will require the use of the secure copy protocol for transferring the application files to the hosting destination of the remote virtual private server.

System Features

Below a number of features and their associated priorities are stated, for each feature its reflective priority is how essential it is within the execution and delivery of the system.

1. Tweet Collection

Description and Priority

High Priority: The System must be able to collect, parse and save tweets that satisfy a selection criterion.

Stimulus/Response Sequences

Using a command line interface; a user can give a selection criterion e.g. set of hashtags that tweets must contain and the system will collect tweets that satisfy that selection criterion.

Functional Requirements

REQ-1.1: The System must allow the user to specify hashtags for which to collect tweets for.

REQ-1.2: The System must collect tweets using its given selection criterion.

REQ-1.3: The System must permanently store the tweets it collects.

2. Tweet Tokenization

Description and Priority

High Priority: The System must be able to tokenize tweets as part of a preprocessing step for classifications.

Stimulus/Response Sequences

Using a command line interface; a user gives a tweet selection criteria, and the system will tokenize all tweets that satisfy the selection criteria and save the resulting tokenized tweets to the database.

Functional Requirements

REQ-2.1: The System must provide an interface to allow users to manually tokenize tweets.

REQ-2.2: The Systems tokenization interface must accept the user configurable input of a tweet selection criteria.

REQ-2.3: The System must be able to tokenize a plaintext tweet into a classifiable form.

REQ-2.4: The System must permanently store tokenized tweets that it has computed.

3. Tweet Classification

Description and Priority

High Priority: The System must be able to classify tweets against configurable classification models and store the resulting classifications for later use.

Stimulus/Response Sequences

Using a command line interface; a user gives a selected classification model and tweet selection criteria, and the system will classify all tweets that satisfy the selection criteria and save the resulting classifications to the database.

Functional Requirements

REQ-3.1: The System must provide an interface to allow users to manually classify tweets.

REQ-3.2: The Systems classification interface must accept user configurable inputs of a tweet selection criteria and classification model.

REQ-3.3: The System must classify tweets on whether they are in favour (leave) or against (remain) Britain Leaving the European union using brexit stance models.

REQ-3.4: The System must classify tweets on whether they are positive or negative towards Britain Leaving the European union using a sentiment polarity model.

REQ-3.3: The System must permanently store classifications that it has computed.

4. Visualization

Description and Priority

High Priority: The System must allow for the visualization of the tweets for which it has collected and the classifications for which it has computed.

Stimulus/Response Sequences

Using the visualization interface a user can;

- view tweets that the System has collected.
- view the classifications that have been computed by each classification model within the System.
- view the classifications the System has computed for a given classification model within a given time unit period.

Functional Requirements

REQ-4.1: The System must display all Tweets that it has collected.

REQ-4.2: The System must provide a means of visually querying the resulting classifications by classification model.

REQ-4.3: The System must provide a means of visually querying the resulting classifications by time unit periods e.g. day/week/month/year.

REQ-4.4: The System must display all classifications and their associative tweets

5. Automation

Description and Priority

High Priority: The System must collect, tokenize, classify and visualize tweets autonomously.

Functional Requirements

REQ-5.1: The System must conduct Tweet Collection autonomously.

REQ-5.2: The System must conduct Tweet Tokenization autonomously.

REQ-5.3: The System must conduct Tweet Classification autonomously.

REQ-5.4: The System must conduct Tweet Visualization autonomously.

Other Nonfunctional Requirements

Performance Requirements

NFREQ-1: The System must operate in real time and thus background operations must execute within an a max of an hour time window.

NFREQ-2: The System must respond to HTTP Requests within 3 seconds.

Security Requirements

The System will source its tweet data from the set of tweets which are publicly available or were publicly available at time of collection and thus the System abides by the twitter terms of service. However, to ensure the System has the upmost ethical grounds the System can give the option to allow users who the System has collected tweets from to delete their data.

NFREQ-1: The System must allow users from which their tweet data has been collected to remove their data from storage.

Software Quality Attributes

NFREQ-1: The System must operate cross platform on the common operating systems e.g. Linux Distributions, Windows and Mac OSX.

NFREQ-2: The System visual interface must work across common browsers e.g. Safari, Chrome, Firefox and Opera.

NFREQ-3: The System must be flexible so that it operates as a framework allowing for classification models to be easily plugged in.

NFREQ-4: The operation of System commands must be intuitive and easy to execute by a user who has moderate technical experience.

NFREQ-5: The System visual interface must be intuitive to use and navigate by a general public user with little to no technical experience.

NFREQ-6: The System must be extensible in that the data models it stores are extendible.

NFREQ-7: The System should be reliably up and operating (and only down for when a new version is being deployed).

NFREQ-8: The System must favor efficiency within efficiency vs memory complexity trade offs in the system.