**Software Requirements Specification (SRS)**

**for Sentiment analysis**

**Rev 3**

**DEC, 2020**

**International IT University**

**Management IT projects**

**Ulan Kassabulatov, Aiasem Koshimbay, Zharas Kyyakhmetov**

**Contents**

1. Objective
2. References
3. System Description
   1. User application
   2. Twitter Server
   3. Hardware Interface
   4. Communications Interface
4. External Interface Requirements
   1. Operating System
   2. I/O Hardware
5. Functional requirements

5.1. Retrieving Input

5.2. Real-Time Processing

5.3. Sentiment Analysis

5.4. Output

1. Non-functional Requirements

6.1. Performance Requirements

1. Safety requirements
2. Module Interface Requirements
   1. User Inputs
   2. Table
   3. Error notifications
   4. Popular Tweet Notification
3. Design Constraints
4. Change History

1. **Objective**

The document defines the Software Requirements Specification (SRS) for the sentiment analysis. The sentiment analysis will extract twitter data using twitter public API and apply some techniques for detecting sentiments or emotions for a tweet.

In general, the purpose of an SRS is to define software requirements, not software design details. In this project, a small amount of high-level design has been performed to decompose the software into logical modules, as described in the System Description.

The External Interface Requirements define the interaction of the entire MRC-II Software with external hardware and software.

For each module, the requirements are then separated into the following categories:

• Functional Requirements

• Performance Requirements

• Safety Requirements

The Module Interface Requirements define the internal interfaces between the software modules.

The Design Constraints specify certain conditions that are imposed on the design.

1. **References**

Twitter: <https://twitter.com/about>

Twitter Developer Terms of Service: <https://dev.twitter.com/terms/api-terms>

1. **System Description**

The web application for sentiment analysis is the system for retrieving data from Twitter API and then apply some methods in order to determine sentiments or emotions for a tweet by users. Initially, retrieve data is considered a keystone in the project as it detects and extracts topic titles from the tweets. Using hashtags is not informative enough about the topic of sentiment the author mentions in his/her tweet. Then perform an intelligent processing technique to extract the true meanings of the people’s comments and to decide and classify them in terms of positive, negative or neutral thus to know the majority of people like or dislike the desired topic. More specifically providing people's feelings regarding certain topics with high accuracy will lead to better decision making.

The figure 1 provides architecture diagram of the tool, which will provide the following functions:

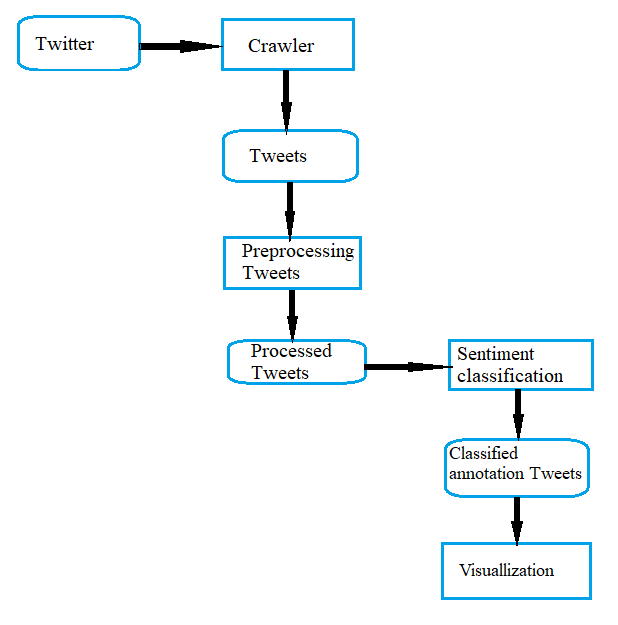


Figure 1. Architecture diagram of method for sentiment analysis

* 1. **User Application**

The interface will meet the following requirements to conform to the users’ needs. It will be simple and easy to understand. Controls which allow the user to interact with the application will be clear and imply their functionality within the application. The interface will include user inputs as well as tables, outlined below. The Tables displayed to the user will provide a visual representation of the output produced, is the message positive or negative.

* 1. **Twitter Server**

The Twitter server provides data from Twitter API. This API is directly accessible to local User Applications and indirectly to remote User Applications.

* 1. **Hardware Interface**

The application is intended to be a stand-alone, single-user system. The application

will run on a web application. No further hardware devices or interfaces will be required. Only one device will be included:

* Windows and Linux users’ computers.
  1. **Communications Interface**

Internet connection and a web browser are required in order to make use of several functions and to be executed such as searching, viewing and downloading.

**4.      External Interface Requirements**

**4.1.     Operating System**

**4.1.1. Portability**

The software will run on the web application, specifically used Python language for backend (machine learning)

**4.2. I/O Hardware**

**4.2.1. Inputs**

The user interface will supply the keywords and the analysis session duration, while the Twitter API will supply the Tweet text.

**4.2.2 Outputs**

The output will portray the current mood of the Twitter community on a given topic in the form of a simple table, which represent positive or negative result. In general, it will indicate the percentages of all positive and negative information.

**4.2.3. Operating System**

The software will run on the python jupyter compiler, specifically python version 3.6 and above

**5.** **Functional Requirements**

**5.1 Retrieving Input**

The software will receive three inputs: keywords, analysis session duration, and Tweets.

● Keywords will be entered by the user for each topic.

● The analysis session duration will be set by the user before each session.

● Tweets will be retrieved with the Twitter Streaming API

**5.2. Real-Time Processing**

The software will take input, process data, and display output in real-time. This will enforce that the snapshot provided by the simple gauge is a current view of the Twitter community’s mood on the chosen topic.

**5.3. Sentiment Analysis**

Sentiment analysis will be performed on the user-specified keywords within the Tweet to determine the overall mood of the Tweet relative to the topic. The sentiment analysis will provide a negative or positive numeric sentiment value.

**5.4. Output**

The software must output real time data in the form of a simple gauge. In addition, the software may output a table of mood trends over time, as well as additional statistics pertaining to a topic (average sentiment over all analysis sessions and total number of tweets processed). This output should be clear and easy to understand.

**6.** **Non-functional Requirements**

**6.1. Performance Requirements**

**6.1.1. Real-Time**

The software will provide information by analysing messages only on Twitter Social Network. The output should display the real time mood of the user by processing the last messages and be updated at least once a day.

**6.1.2. System Resource Consumption**

Resource consumption of this application should not exceed the amount that device remains usable. The application should be capable of operating in the background if the user wish to use other applications.

**7. Safety requirements**

* The software should never disclose any personal information of Twitter users, and should collect no personal information from its own users.
* For the safety requirements nothing but an operation of weekly backups for the data base should take place.

**8.      Module Interface Requirements**

The interface will meet the following requirements to conform to the users’ needs. It will be simple and easy to understand. Controls which allow the user to interact with the application will be clear and imply their functionality within the application. The interface will include user inputs and tables with outputs. The table illustrated to the user will provide a visual representation of the output produced.

**8.1 User Inputs**

The user will do sentiment analysis by editing, adding and deleting keywords for each tweet.

**8.2. Table.**

The table will consist of the original tweet, the date and twitter account of the users.

**8.3. Error Notifications**

Error notifications will be required within the application, presenting the user with appropriate messages which describe the error that has taken place. If applicable, error messages should suggest possible solutions to the problem.

**8.4. Popular Tweet Notifications (Optional)**

A notification system will be set up to inform the user when any popular Tweets have been processed through the application.

**9. Design constraints**

This application will be implemented in HTML and Python and will be designed to work on all internet browsers and support at least following Web Browsers:

• Chrome

• Microsoft Edge

• Firefox

• Safari

**10.  Change History**

|  |  |  |
| --- | --- | --- |
| **Rev** | **Date** | **Description** |
| 1 | 13/11/20 | Initial Version (changes tracked by date for now) |
| 2 | 19/11/20 | Modified Version |
| 3 | 28/12/20 | Modified version(add non-functional requirements) |