The Tweetables

Ada Bulgur, Amir Crutchfield, Sydney Johnson, Micaiah Steplight, Isaac Stinnette



Requirements

Table Of Contents

[**1.0 Introduction 3**](#_heading=h.2s8eyo1)

[**1.1 Goals and Objectives 3**](#_heading=h.1y810tw)

[**1.2 Scope 3**](#_heading=h.4i7ojhp)

[**1.3 Software Context 3**](#_heading=h.2xcytpi)

[**1.4 Major Constraints 4**](#_heading=h.1ci93xb)

[**2.0 Usage Scenario 5**](#_heading=h.3whwml4)

[**2.1 Overview 5**](#_heading=h.2bn6wsx)

[**2.2 Special Usage Considerations 5**](#_heading=h.qsh70q)

[**3.0 Design Model and Description 6**](#_heading=h.3as4poj)

[**3.1 UI Design 6**](#_heading=h.1pxezwc)

[**3.2 API Usage 6**](#_heading=h.49x2ik5)

[**3.3 Algorithm Design 6**](#_heading=h.2p2csry)

[3.4 Executable 7](#_heading=h.260x85d76lt6)

[**4.0 Functional Model and Description 8**](#_heading=h.147n2zr)

[**4.1 Data Acquisition 8**](#_heading=h.3o7alnk)

[**4.2 Description of Algorithm 8**](#_heading=h.23ckvvd)

[**4.4 Description of Application 8**](#_heading=h.ihv636)

[**4.5 Additional Features 9**](#_heading=h.32hioqz)

[**5.1 Restrictions 10**](#_heading=h.w9koppg5drv)

[**5.2 Limitations 10**](#_heading=h.nc2ny9ybzsnh)

[**5.3 Constraints 10**](#_heading=h.95p2dvyvk6wl)

[**6.0 Revision Log 11**](#_heading=h.41mghml)

# 1.0 Introduction

Tweetables has been given the task of creating a complete Sentiment Analysis system. Sentiment Analysis applications work by measuring the emotional tone of certain products or subjects. Our program will work in the same fashion by using data from social media. By doing this we will be able to achieve our task of being able to measure the emotional tone of certain products. This document will outline all the requirements for this task.

## **1.1 Goals and Objectives**

The goal of the project is to create a running application that will help corporations determine the emotional tone of a selected product. The application will do this by sorting people’s feelings for a product into three different categories. These categories are positive, negative and neutral.

## **1.2 Scope**

Tweetables plans to use a Lexicon-based approach in order to build a Sentiment Analysis application. This will be done by using APIs from X in order to build a Lexicon-Based dictionary that will be used to measure people’s emotional tone towards a product. The application will include components that will allow it to gather data, clean data, analyze data and then display the results of the data. Our application will not be using any ML-based techniques. A synopsis of the tasks that must be completed to create the application are:

* Utilize X’s APIs in order to gather data.
* Implement data cleaning so that the analysis focuses only on meaningful words.
* Implement Lexicon-Based sentiment analysis within the application so that certain words get tagged with the correct polarity (positive, negative, or neutral).
* An installation script will be provided to install the app so that it can be executed with a single click on the installed file.

## **1.3 Software Context**

| Software | Description | Type |
| --- | --- | --- |
| Google Drive/Google Document | Cloud-based storage service that allows the team to store and collaborate on documents. | File Sharing |
| Visual Studio Code | An integrated development environment used for all of coding purposes for the project | Application |
| Python | The Programming language that the application will be created using. | Language |
| Jira | Allows tasks to be created and assigned to team members to be completed for each iteration. | Application |
| X (Twitter) | The social media platform where we will be using its APIs in order to build the program. | Application |

## **1.4 Major Constraints**

Implementation: Learning how to properly use X’s API. This is a major constraint because it is integral that we understand how the API works in order to get our application to run. It could take a lot of time to learn how to use X’s APIs which could lead to delays with our application.

Lexicon-Based Techniques: Not everyone is familiar with how to implement techniques related to Lexicon-Based searches. As a result this could cause delays as people have to take the time to properly learn how to use Lexicon- Based techniques.

# 2.0 Usage Scenario

The program will be utilized by companies and others in order to find out how certain products or subjects are received by the general public. By doing so companies will have a better idea on how the public feels about certain things. This could help companies with making decisions such as whether to market a certain product better. It could also help companies decide whether to improve a certain product.

## **2.1 Overview**

A user would enter a product or subject into the interface. Once this is done the application will display the results of the emotional tone towards the subject or product. It will display a chart showing how much the product or subject is associated with a positive, negative or neutral tone. The application is meant to be user friendly so that it can be used by a wide range of people.

## **2.2 Special Usage Considerations**

The application won’t be limited to just English. Users will have the ability to use the application in multiple languages. As a result, the application can be used across different cultures. The application will also be compatible with multiple operating systems. By doing this the application can become more accessible to a wider range of people.

# 3.0 Design Model and Description

## **3.1 UI Design**

The user interface (UI) will be designed with simplicity and ease of use in mind, ensuring that users can quickly access and understand the sentiment analysis results. The core feature of the UI will be a chart that visually displays the sentiment distribution for a given product or subject. This graph will show how much the product or subject is associated with positive, negative, and neutral sentiments, making it easy for users to interpret the emotional tone at a glance. To further enhance the user experience, the UI will include an additional filter option that allows users to select the language in which they wish to view the sentiment analysis results. This will enable the application to support multiple languages, ensuring that users from different regions and linguistic backgrounds can access the sentiment data in a language they are comfortable with.

## **3.2 API Usage**

The application will utilize X’s Search Tweets API to gather real-time tweet data based on user-specified keywords, hashtags, tweets, or mentions. This will allow the system to analyze current sentiment related to a specific product or subject. To efficiently manage the data retrieval process and stay within X’s rate limits, the system will implement request queuing and batch processing, enabling multiple tweets to be fetched at once and reducing the likelihood of exceeding API limits. Additionally, the application will leverage X’s language detection feature to identify the language of each tweet automatically. This allows users to filter and view sentiment analysis results in their preferred language, supporting a diverse, multilingual user base. By including this language filtering option, the application ensures its accessibility to users across different regions and cultures. To maintain secure access to X’s API, OAuth 1.0a authentication will be used, ensuring the application can safely interact with X’s services while protecting user data. This approach will help provide reliable, real-time sentiment analysis based on the most up-to-date public opinions.

## **3.3 Algorithm Design**

The core of the sentiment analysis in this application will be a lexicon-based algorithm. This method involves the use of a predefined dictionary of words, each associated with a sentiment polarity (positive, negative, or neutral) and a sentiment strength. When a tweet is processed, it will be broken down into individual words or tokens, a process known as tokenization. Next, common words that do not contribute to sentiment, such as “and” or “the”, will be removed through a step called stop-word removal. The remaining words will be compared against the sentiment lexicon, and a polarity score will be assigned to each word based on its sentiment. These scores will be aggregated to determine the overall sentiment of the tweet. The final step will involve calculating the sentiment for the entire dataset by aggregating the individual sentiment scores of all the tweets. The algorithm will also be capable of handling multilingual tweets, using language-specific lexicons where necessary to ensure accurate analysis regardless of the language of the tweet.

## **3.4 Executable**

Users can easily run our product by launching the standalone executable file generated using PyInstaller. This approach provides a seamless user experience, as it removes the need for manual setup or configuration. PyInstaller plays a critical role in this process by converting our Python program into a self-contained executable. It bundles the Python interpreter, code, and all necessary dependencies into a single file, main.exe allowing users to run the application without installing Python, setting up virtual environments, or managing external libraries. This not only simplifies distribution but also ensures consistency and reliability across different systems.

# 4.0 Functional Model and Description

## **4.1 Data Acquisition**

The application will allow users to specify a product, subject, or keyword to analyze public sentiment. By leveraging X’s Search Tweets API, the system will fetch tweets in real-time. To ensure efficient data retrieval, the system will incorporate mechanisms like queuing and batch requests, enabling it to handle X’s rate limits effectively. Collected tweets will include metadata such as timestamps, and language, ensuring comprehensive sentiment analysis. Data will be temporarily stored in a database for processing to maintain a smooth flow between collection and analysis.

## **4.2 Description of Algorithm**

The core of the application’s functionality is a lexicon-based sentiment analysis algorithm. This method uses a predefined dictionary of words, where each word is associated with a polarity (positive, negative, or neutral) and a sentiment strength. The process is as follows:

* Tokenization: Tweets are broken down into individual words.
* Lexicon Matching: Tokens are matched against the sentiment dictionary, and a polarity score is assigned to each word.
* Score Aggregation: Polarity scores of all tokens in a tweet are combined to determine the overall sentiment of the tweet.
* Dataset Sentiment Calculation: The system calculates the sentiment for the entire dataset by aggregating the sentiment scores of all tweets.

To support multilingual capabilities, the algorithm will employ language-specific lexicons to ensure accurate sentiment analysis regardless of the tweet’s language.

4.3 Data Cleaning

To produce accurate results, the raw tweet data will undergo a rigorous cleaning process. This involves:

* Removing unnecessary elements: URLs, user mentions (@user), hashtags, emojis, and special characters.
* Standardization: Converting all text to lowercase to avoid inconsistencies.
* Tokenization: Splitting each tweet into individual words or tokens.
* Stop-word removal: Excluding common, non-informative words like “and” or “the”.
* Stemming or Lemmatization: Reducing words to their root forms for consistency.

## **4.4 Description of Application**

The application is designed to provide a seamless user experience. Upon entering a keyword or topic, users will receive a visual representation of the sentiment analysis in the form of an interactive graph. This graph will categorize sentiments into positive, negative, and neutral, allowing users to interpret the emotional tone easily. A filter option will enable users to select their preferred language, ensuring the application is accessible across various linguistic backgrounds. The application will perform the following tasks:

* Collect tweets in real-time based on user inputs.
* Process and clean the data to remove irrelevant elements.
* Analyze the cleaned data using the lexicon-based algorithm.
* Present results visually through graphs, allowing users to interpret the sentiment distribution quickly.
* Provide options to export sentiment analysis results in formats like CSV or PDF for further use.

## **4.5 Additional Features**

Multi Language Support: Users can select the language of their choice, ensuring the application caters to a global audience.

Real-Time Analysis: The application will fetch and analyze tweets in real time, providing users with up-to-date sentiment insights.

Data Export Options: Users will have the ability to save results for offline review or integration into other systems.

Username and Password Encryption: This allows our users username and passwords to be more secure preventing them from being exposed during a data breach

5.0 Restrictions, Limitations, and Constraints

## **5.1 Restrictions**

* Due to the strict time limits, it is possible that some jobs will be completed hurriedly or incompletely to ensure that everything is finished by the planned delivery date.
* X’s API free tier: While X’s free API tier will allow us to access the data we need to successfully complete a sentimental analysis there are restrictions like a monthly tweet retrieval limit at 1500 tweets a month and there are also data storage limitations.

## **5.2 Limitations**

* Data limitations: There are limitations to the amount of data we can fetch due to the API’s free tier restrictions. Also there is potential bias in data on X making it difficult to detect tweets that are serious or trolls.
* Model limitation: Our model could struggle sensing sarcasm in Tweets making it difficult to tell if certain words have a positive or negative tone.

## **5.3 Constraints**

* Incomplete Data: A major feature of X is threads which are connected tweets that allows users to add more context past the 280 character limit. When fetching data, only reviewing one part of a thread could lead to missing context which will negatively impact our sentimental analysis.

Time: To reach our project's deadline, we're employing timesheets to keep our team organized and on track. To ensure the project is completed as quickly as possible, several team members who are proficient in Python are splitting up the coding responsibilities among themselves.

# 6.0 Revision Log

| Version | Date | Document |
| --- | --- | --- |
| 1.0 | 1/24/2025 | Requirements Document Version 1.0 |
| 2.0 | 2/13/2025 | Requirements Document Version 2.0 |
| 3.0 | 3/6/2025 | Requirements Document Version 3.0 |
| 4.0 | 3/27/2025 | Requirements Document Version 4.0 |
| 5.0 |  | Requirements Document Version 5.0 |