Social Media Posts Sentiment Analyzer

By: GROUP 19-5

DEPARTMENT OF NETWORKS

SCHOOL OF COMPUTING AND INFORMATICS TECHNOLOGY

A methodology submitted to the School of Computing and Informatics Technology For the Study Leading to a Project Proposal in Partial Fulfillment of the Requirements for the Award of the Degree of Bachelor of Science in Software Engineering of Makerere University.

Supervisor: Dr. Ntanda Moses

Department of Networks

School of Computing and Informatics Technology, Makerere University

mntanda@cis.mak.ac.ug/kmntanda@gmail.com, +256-70293-8609

|  |  |  |
| --- | --- | --- |
| **NO** | **NAMES** | **RE NO** |
| 1 | ATWINE NICKSON | 15/U/4058/EVE |
| 2 | MUGABO AMUZA | 15/U/7798/PS |
| 3 | TURYAHIIRWA HAPPINESS | 15/U/19330/EVE |
| 4 | MULINDWA HENRY | 15/U/8235/EVE |

# 3.0 Methodology

## 3.1 Introduction

In this chapter we discuss the methods that were used in the research study of the proposed system. The methods that were applied to achieve the objectives of the system included the techniques and tools that were used in collecting, analyzing, designing, implementing, testing, verification and validation of the system.

We also discuss the datasets collected that were used to build and test the classifier and the performance measures used to determine how accurate the model is working.

We present all the information that the reader requires to replicate the study and to understand that all steps in the methods are done to ensure the readability and validity of the study.

The project’s design and development stages were completed in small iterations. This is a core part in the agile methodology which was extensively applied throughout the project. The time available for development was divided into smaller chunks resembling the iterations, each having a deadline and a set of tasks to be completed [15]. A mini-plan covering a summary for each iteration can be found in ​ Appendix A​ [To be put].

Furthermore, GitHub was used as a version control platform. At the end of each iteration a new branch containing the tasks done was created. If the code passed the tests, the current branch was merged with the master branch. If the code did not pass the tests, the code remained unmerged until errors / conflicts were resolved. To help with the progress tracking, JIRA, a third party software offering an implementation of the “Tasks Board” agile practice, was used.

## 3.2 Data sets and corpus collection

Using Facebook API, we collected a corpus of Facebook posts with their respective comments. We collected a corpus of posts and formed a dataset of three classes: positive sentiments, negative sentiments and neutral sentiments.

The collected corpora were used to train the classifier to come up with an algorithm that was able to classify sentiments as positive, negative or neutral.

In order to collect a corpus of objective posts, we retrieved posts from Facebook accounts of Uganda’s leading newspaper publishing companies namely; New Vision, Daily Monitor and The observer.

To be able to collect datasets, we designed a python script that was able to scrap posts and the respective comments from the mentioned Facebook accounts. A total of 300000 posts were collected with their respective comments to be used in training the classifier.

We perform a linguistic analysis of our corpus and we show how to build a sentiment classifier that uses the collected corpus as training data. We distributed the collected corpus to selected people who hand labeled training data into positive, negative or neutral.

We used 66% of collected posts and reactions for training the classifier, 33% for testing the classifier and 1% for validating the classifier.

## 3.3 Requirements Gathering

Requirements, both functional and non-functional, were gathered at different stages during the development process, contrary to a classical upfront approach. However, defining a base set of requirements and expected behavior was necessary to begin development. As such, the engine had to be scalable, allowing integration of new modules without affecting its performance. In addition, standard structure of the processed data was defined, using JSON (JavaScript Object Notation) formatting, where data objects were stored using attribute-value pairs.

A non-functional requirement prior to the development stage, was researching the state of the art in sentiment analysis. From this perspective the accuracy of the algorithms used for classification, the flexibility of implementing heuristics on such algorithms, and the time required for implementation and testing were assessed. For example, a system using Neural Networks requires more time for implementation compared to one where Naïve Bayes is used. In addition, Neural Network algorithms require in depth understanding in order to perform heuristics, while a probabilistic model like the Naïve Bayes algorithm is more flexible [13].

### 3.4 Algorithm Requirements

Most of the requirements for the engine are of a functional nature. Hence, heuristics on improving the machine learning algorithm were required, as well as the use of third party software. An initial list of functional requirements ranked by priority and number of hours required for completion is presented in ​ Table 3.1

In order to achieve the behavior desired, a Facebook API had to be used to collect data that would be used to develop the classifier.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Requirements | Priority | Hours | Development Stage |
| 1 | Train the main classification algorithm | Very High | 15-20 | Early |
| 2 | Output and store the model after the training phase of the classification algorithm | Very High | 5-10 | Early |
| 3 | Test the classification algorithm proposed | Very High | 15-20 | Early |
| 4 | Clean the noise from the data mined | High | 10-15 | Middle |
| 5 | Store the acquired data depending on the component in which the engine is used | High | 20-24 | Middle to late |
| 6 | Train an additional machine learning algorithm for comparison with the main algorithm | Low | 24 | Late |
| 7 | Based on the selected topic and a reaction, classify the reaction as negative, positive or neutral | Low | 10-15 | Late |

Table 3.1: Algorithm - functional requirements

Another important requirement was data storage which was implemented with Sqlite3.

A more detailed list of non-functional requirements is outlined in ​ Table 3.2

|  |  |  |
| --- | --- | --- |
| No. | Non-functional requirements | Priority |
| 1 | Extensibility – the algorithm should be able to implement new modules without performance issues | Very High |
| 2 | Efficiency - the algorithm should be able to support large intakes of data without affecting its performance | High |
| 3 | Speed - The algorithm should perform sentiment analysis in seconds from the time the request was made | Very High |
| 4 | Robustness - the system should be able to run multiple instances of algorithm | Medium |
| 5 | Scalability - the algorithm should scale the output size based on the size of the input | Very High |

Table 3.2: Algorithm - non-functional requirements

## 3.5 Data Gathering

During this step, we considered all the current priorities that would be affected and how they should be handled. We used the following and they include: Interviewing, observations, questionnaires, and existing literature as elaborated below.

### 3.5.1 Interviewing

We interviewed a total of random 30 people, selected 50 students from 3 institutions, 2 politicians and 3 companies.

The 50 students were from Makerere University, Kyambogo University and Kampala International University while the other 30 were got from around the streets of Kampala.

The interviewed politicians introduced us to their social media handlers who gave us a clear view on the process they go through to analyze posted reactions

The visited companies which included The Uganda Human rights commission, Coca Cola and Airtel also introduced us to those in charge of handling social media applications all who we interviewed and gave us details on how they analyze data and the challenges they face.

The data collected assisted us to understand how they manually operate, the problems they face and the user requirements for the proposed system. See attached Appendix A (Interview guide)

The reasons for using this method included a possibility of obtaining supplementary information, ability to obtain quick feedback from respondents and being able to collect additional view from different people about the problems of the existing system and the requirements of the new system.

Below is a summary on how we carried out the interview process

|  |  |
| --- | --- |
| Random Selected People | Age bracket |
| 20 | 19-30 |
| 10 | 30+ |

|  |  |
| --- | --- |
| Institution | Number of Interviewees |
| Makerere University | 30 |
| Kyambogo University | 10 |
| Kampala International University | 10 |

|  |  |
| --- | --- |
| Politicians and Companies | Number of Interviewees |
| Airtel | 1(Social Media Handler) |
| Coca-Cola | 1(Social Media Handler) |
| Politician 1(Ministry of Justice) | 1(Social Media Handler) |
| Politician 2(Ministry of Education and Sports) | 1(Social Media Handler) |

### 3.5.3 Questionnaires

Questionnaires were issued to people to be able to get information from them on how they use social media through reactions and how they analyze other people’s reactions. See Appendix B (Questionnaire).

A total of 600 questionnaires were issued to different categories of people and we received back 500 questionnaires which were used to analyze the challenges faced and new requirements for the proposed system.

We used questionnaires because they are easy to analyze, cheap to use to contact a large sample of data, easy to receive responses in the shortest time possible, they are easy to analyze, and information is always collected in a standardized way.

### 3.5.4 Existing Literature

Included reading online related literature like magazines, journals, books to know the existing trends in the market. This technique helped us not to avoid repeating already existing systems. Review of the existing literature and documents on sentiment analysis of social media posts was carried out to enlighten us on the way people react to social media posts. It enabled us to investigate problems and benefits of the existing system. This technique helped us to understand other relevant written material on the problem area and how other researchers and system developers have tried to solve similar problems. Information generated from manual reports was used to identify the inputs, processes and outputs.

### 3.5.5 Observation

We interacted with over 20 people and went through the process of reacting to posts with them. We were able to find out that people react differently to same posts. Some like the content of the post and other like the post because of their own personal reasons. This case was common especially on sports posts where someone liked the post because he supports the club it is about while the other one likes the post because of what it was all about. This was a big challenge as it is hard to find out why users liked or disliked some posts.

There were also cases where posts and reactions conflicted. A post would be negative but to someone, it would be positive in its negative way.

This method helped us to identify the problem areas. The process people took to analyze reactions to social media posts. This technique enabled us to identify the weaknesses of the existing system.

## 3.6 Data Analysis

The data collected from questionnaires and interviews was edited and inserted into CSV files. The data was analyzed using RStudio by importing the files with edited data to discover the trends and relationships among the data. From this analysis, we were able to infer a number of system and user requirements.

## 3.7 System Design

The data that was obtained at the analyzing phase was used to derive requirements of the new system. The requirements included application and database designs.

### 3.7.1 Application Design

The technique of Data Flow Diagrams (DFDs) was used to illustrate the processes involved in achieving the objectives if the system. [We can use sequence diagrams, class diagrams etc more research]

### 3.7.2 Database Design

This was carried out in three phase’s namely conceptual, logical and physical design.

**A conceptual design** typically describes an entire enterprise. In conceptual design, we defined entities like article, reaction, authors, user and publishing company. We also defined the relations among the entities.

**A logical data model** is derived from and or linked back to objects in a conceptual data model. Typically, logical design describes data requirements of our project.

**Physical design** is dependent on a specific version of a DBMS, data storage location or technology. We used SQLite since we used python in PyCharm.

The database stores the downloaded/fetched comments and posts from Facebook API. The database stores posts against their specific comments. The posts and comments are then fetched from the database for classification.

## 3.7 System Architecture

The product was developed using three-tier architecture. The architecture includes the following sections;

**Interface/presentation**: This included software development tools suitable for interface development, business logic implementation. We used FLASK, HTML, CSS and JavaScript for developing the user interface.

FLASK

Flask - a third party micro framework in Python, which offers a build-in development server with restful request dispatching [20]

“Micro” in micro framework means Flask aims to keep the core simple but extensible.

Flask stands out from other frameworks because it lets developers take the driver’s seat and have full creative control of their applications. Flask supports relational databases, which was needed to save our classified data.

HTML

We used html to develop static web pages for user interface.

CSS

We used CSS for page formatting and design.

JavaScript

JavaScript enables interactive web pages and thus is an essential part of our web application part product.

**Application logic/middle tier**: This connects the presentation and the back end. We used python to write the middle tier and to connect the back end and the presentation.

# PYTHON

Why Python?

Given the context of processing large amounts of data, memory management became a prioritary. Consequently, a programming language which is capable to handle processing and storage of these amounts of data was imperative. An iterative system approach, when processing a list of items, has to firstly store it, which requires memory. In these regards, Python provides generators, which are particularly useful when processing large amounts of data, passing the source data through the processing chain, one item at a time, storing only the results of the processing chain [11].

Considering the above argument, Python proves capable of efficiently managing memory, a task crucial for the ‘real time’ component of the project. A drawback of Python is that it is an interpreted language, which by contrast, is slower than compiled languages (such as C, Java, etc.). The developers’ community considered the disadvantage, and proposed different ways to improve Python’s speed. As such, projects like Numba and PyPy are viable solutions. The creator of Python, Guido van Rossum recognizes the improvements added to Python and states that PyPy is the best way to obtain high-performance systems while using Python [12].

Furthermore, advanced libraries for data processing such as NumPy and SciPy were developed by the scientific community and domain experts [29]. Such tools proved to be helpful during the development stage, and reinforced the reasoning of choosing Python.

**Database/Back end**: This included database designing. We designed it using SQLite

Loading the Data

As usual, we first downloaded our datasets locally, and then we will load them into data frames using Python.

Preparing the Corpus

In linguistics, a corpus or text corpus is a large and structured set of texts (nowadays usually electronically stored and processed). They are used to do statistical analysis and hypothesis testing, checking occurrences or validating linguistic rules within a specific language territory.

In our particular case, we are talking about the collection of reaction fragments that we want to classify in positive, negative or neutral sentiment. Working with text corpora involves using natural language processing techniques. Python is capable of performing really powerful transformation with textual data. However we will use just some basic ones.

The requirements of a bag-of words classifier are minimal in that sense. We just need to count words, so the process is reduced to do some simplification and unification of terms and then count them. The simplification process mostly includes removing punctuation, lowercasing, removing stop-words, and reducing words to its lexical roots (i.e. stemming).

So, we process our reactions and create a corpus. We also extract important words and establish them as input variables for our classifier.

### Algorithm Used

We build the classifier using the multinomial Naïve Bayes Classifier.

Naïve Bayes classifier is based on Bayes theorem which uses the theorem to build a probabilistic classifier.

We basically sidelined with Naïve Bayes classifier because it is simple, trains faster than other classifiers, and can work well when we don’t have sufficient training data.

We can also create solid baselines with little efforts and help us to explore more complex solutions.

We also tried Linear Regression however; Naïve Bayes yielded the best results.

## 3.8 System Testing

This chapter presents the testing methodologies that were applied during development, under the test driven development recommendations [26]. As such, unit testing was exhaustively used for each component module of the engine and the graphical user interface. In addition, end to end testing was performed on the major parts of the system

The resultant system was tested with the users to see if it works as expected and to provide the stake holders with information about the quality of service being proposed. It was aimed at meeting the business and technical requirements that guided design and development.

Before actually implementing the new system into operation, a test run of the system was done removing all the errors that existed. It is an important phase of a successful system. After codifying the whole programs of a system, a test plan was carried out and run on a given set of test data. The output of the test run was matched with the expected results. System testing is considered as part of implementation process.

Using the test data, the following test run are carried out:

### 3.8.1 Unit Test

For the algorithm, unit testing was done in a systematic manner, each module being tested during or immediately after it was developed. In these regards, “pytest” - a third party library for Python was used. ​ Figure 5.1 is the example code comprising of one test case for unit testing the tokenizer module. An expected behavior is defined and tested against the input. Consequently, all the modules developed were tested using the same format [28]

Figure 5.1: (To be put here) Unit testing (engine) - tokenizer module

This approach was useful in discovering errors which affected the engine. In addition,​Table 5.1 presents a set of malfunctions were found for the modules tested. The ordering in table was done based on the error prone of each module. As such, the stream module and the graphical user interface required a significant amount of time for debugging compared to the other modules. While the GUI seemed challenging due to the limited knowledge in JavaScript, the stream module was affected by the use of the Facebook API which for the purpose of the project required a multi-threaded implementation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Module | Source | Error-Behavior | Fixed |
| 1 | Stream | Facebook API | We were denied access to public pages as Facebook changed their API . | Yes but as we mined data from Facebook, we would be denied access constantly as we had to change code and disable JavaScript and also use m.facebook.com all the time we needed to scrap data. |
| 2 | GUI | FLASK | Assessing the database and servicing requests was challenging | Yes |
| 3 | GUI | FLASK | Setting up APACHE was very challenging and also uploading the classifier | Yes |

Table 5.1: Sample of unit testing form

### 3.8.2 Program Test

When the programs were coded, compiled and brought to working conditions, they were individually tested with the prepared test data.

### 3.8.3 System Test

While most of the erroneous behavior was discovered and solved using unit testing, end to end system testing was necessary to establish the reliability and the performance of the system as a whole, so that interaction between modules is as expected. In addition, end to end testing was required to ensure that the tests employed cover a significant amount of the code produced [29].

After carrying out the program test for each program of the system and errors removed, then the system test was completed. At this stage the test was done on the actual data. Then the complete system was executed on the actual data.

**The other types of testing that were conducted include;**

**Compatibility testing** – this was done to ensure compatibility of the application with different browsers, operating system and hardware platforms.

**Integration testing** – this was done to ensure that the system components integrate and work together efficiently.

**Usability testing** – was carried out to determine the convenience of the system for example user friendliness of the interface, appeal and navigation to the intended users.

## 3.9 Verification and validation

### 3.9.1 Verification

This was mainly carried out to remove defects from the model we designed.

Verification testing was done using V-Model. The artifacts such as test Plans, requirement specification, design, code and test cases were evaluated.

Verification was done using three methods;

* Reviews; where peers and stake holders challenge the correctness of the work being reviewed. This testing will be completed when presenting the model to the final year project assessment panel.
* Desk Debugging; where we constantly checked the accuracy and completeness of the model we were developing.
* Requirements tracing; where we always ensured requirements were not lost during development of the model.

### 3.9.2 Validation

The through put of the processes within the system is faster. In cases of something wrong, an error message will be displayed.

### 3.9.3 Performance Measures Used

After designing the model, we needed to find out how good the model is at labeling users’ reactions. This section was very important because it delineated how good our predictions were.

We used a confusion matrix to describe the performance of the classification model on the set of data for which the true values were known as shown in the table below.

|  |  |  |  |
| --- | --- | --- | --- |
| Actual | | | |
| Predicted |  | Positives | Negatives |
| Positives | TP | FP |
| Negatives | FN | TN |
|  |  |  |

True Positives (TP) and True Negatives (TN) are the observations that are correctly predicted. We minimized False Negatives (FN) and False Positives (FP).

True Positives: these are the correctly predicted values which mean the value of actual class is yes and the value of the predicted class is also yes. E.g. when a user comments a positive reaction and the predicted class tells you the same thing.

True Negatives: these are the correctly predicted negative values which means that the value of actual class is also no. E.g. when a user comments a negative reaction and the predicted class tells you the same thing.

False Positives: these come up when the actual class is no and the predicted class is yes. E.g. the user comments a negative reaction and the model classifies it as positive.

False Negatives: these come up when the actual class is yes but the predicted class is no. E.g. the user comments a positive reaction and the model classifies it as negative.

Basing on these four parameters, we measured the performance of the classifier using Accuracy and Precision methods.

Accuracy: this is the ratio of the correctly predicted score to the total observations. For our model, we got 0.89 which means our model is approximately 90% accurate. Accuracy can be got from the equation below.

Accuracy = (TP + TN) / (TP+FP+FN+TN).

We used accuracy because the target variable classes in the data are nearly balanced.

Precision: this is the ratio of correctly predicted positive observations to the total predicted positive observations. The question this metric answer is of all reactions that were labeled as positive, how many were actually positive? We got a 0.888 precision which is pretty good for the classifier.

Precision = (TP) / (TP+FP).

We also used precision because we want to be precise on how good our model works.