ASHESI UNIVERSITY

AN AUTOMATED EMAIL CLASSIFICATION SYSTEM FOR THE ASHESI SUPPORT CENTER

APPLIED PROJECT

B.Sc. Computer Science

Kwame Owusu Boahene

2019

ASHESI UNIVERSITY

An Automated Email Classification System for the Ashesi Support Center

APPLIED PROJECT

Applied Project submitted to the Department of Computer Science, Ashesi University in partial fulfilment of the requirements for the award of Bachelor of Science degree in Computer Science.

Kwame Owusu Boahene

April 2019

DECLARATION

I hereby declare that this applied project is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

…………………………………………………………………………………………

Candidate's Name:

…………………………………………………………………………………………

Date:

…………………………………………………………………………………………

I hereby declare that preparation and presentation of this applied project were supervised in accordance with the guidelines on supervision of applied projects laid down by Ashesi University.

Supervisor's Signature:

…………………………………………………………………………………………

Supervisor's Name:

…………………………………………………………………………………………

Date:

…………………………………………………………………………………………

ACKNOWLEDGEMENT

To God, my family and friends whose constant motivation and support helped me through my years at Ashesi.

To my supervisor, Mr. Dennis Owusu whose guidance help me complete this project I express my sincerest gratitude.

ABSTRACT

The widespread usage of the internet has made email an indispensable tool for communication within organizations. Today, email is used by support centers as one of the mediums for providing solutions to the daily internal problems' organizations face. An example is the Ashesi Support Center which is the hub for solutions for all problems and questions relating to IT, facilities, logistics, and other issues on the Ashesi University campus. In dealing with problems, the Ashesi support center classifies emails as either an IT related issue or an operations related issue. However, the support center does not have a way to automatically classify the emails. Hence, a support personnel manually sifts through the emails to group them. This can be a cumbersome process considering the support center receives over 40 emails daily during peak periods. Harnessing the power of machine learning, a classification model is built to automatically group emails the Ashesi support center receives.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1: Introduction

1.1 Background

The Ashesi Support Center is the hub for solutions for all problems and questions relating to IT, facilities, logistics, and other issues on the Ashesi University campus. In their quest to solve problems, the Ashesi support center has an email address which students, staff and faculty can send their problems to. In this email address, the support center classifies reports as either an IT or Operations issue. Thus, the support center has two folders in their inbox where IT and Operations emails are placed.

However, the support center has no way to automatically classify emails. Therefore, a support personnel in dealing with the problems manually places the emails in the appropriate category. This can become a problem as the support center receives over 40 emails daily in peak periods and support personnel have other responsibilities apart from responding to emails [14]. To further add to the problem some of these emails may not report a problem and can be emails everyone in the Ashesi mailing domain receives.

This presents an excellent opportunity for technology to be used to automate the process of classifying emails. Utilizing machine learning approaches to automatically classify emails, this project seeks lessen the burden placed on support personnel and make the problem-solving journey simpler and more efficient.

1.2 Related Work

This chapter explores previous work in building email classification systems. Extensive work has been done in building email classification systems. Particularly in building of spam classification systems [4,5,6] as well as multi folder categorization systems [1,2,3]. As such, it is necessary to review these works to understand the various processes and procedures involved in building an email classification system.

In [1], the authors build an email classification system for a contact center which plays a role similar to the Ashesi Support Center. Their classification system categorizes emails into 3 categories or folders. The classifier groups email the contact center receives as root emails which represents the start of a thread between a customer. Next, leaf emails are grouped which ends the thread and inner emails which represent emails between the root email and leaf emails. The authors use archived data from a web forum to train their classification system.

The authors preprocess the data by taking out stop words, symbols and punctuation marks. They also take out signature blocks from the emails and use non-inflected form of words in the emails. In their classification system, the authors use the tools Rainbow and SvmLight which are implemented using Naïve Bayes and Support Vector Machines respectively. In testing their system, the authors present tables showing the accuracy as well precision and recall scores achieved by the system.

In [2], the authors design and implement an email grouping system as well as an email summarizer. Unlike [1], the authors used a ruled-based approach in building their classification. The authors classify emails based on the user’s activity. Thus, they extract common words in a new email and compare it to common words in the existing user folders to classify the new email. To ensure that only words meaningful to the email are extracted, the authors preprocess the emails to remove stop words.

Similar to [1], the authors in [2] present accuracy, precision and recall scores in the testing of their system. However, in addition to that, they also compare their scores to emails classified by human beings as well as results from an email classification software called LibTextCat. They achieved an accuracy score of 98% whereas LibTextCat achieved 90% showing an improvement over the software.

In [3], the authors build an email classification system which automatically groups user emails into categories without the user predefining any folders. Their system is also able to determine if an email is generated by a machine or human being. The data used consisted of emails by Yahoo users who agreed to be a part of such research studies. The authors used both manual labelling and a ruled-based approach to group emails for their training data. They grouped emails as either human, career, shopping, travel, finance and social.

Like [1] and [2], the authors in [1] remove stop words from their data. Next, the authors build their classification which is a combination of an online and offline system. The online system uses a rule-based approach to automatically classify new emails users receive. The offline system which is built using logistic regression periodically classifies user's inbox into the 6 groups mentioned earlier. After testing their system, the authors achieved recall and precision scores close to 90%.

To have a better understanding of the various email classification methods and the situations where they work best [4,5,6] were reviewed.

In [4], the authors review approaches used to classify emails into a hierarchy of folders, as well as filter spam. First, the authors discuss the purpose of an email classification system which they defined as either classifying spam or classifying emails into its respective folder. Based on this purpose, the authors determine that the classification method used will determine if the system performs efficiently.

Next, the authors discuss the various email classification methods and the situations they work best. They examine Naïve Bayes, TF-IDF, Support Vector Machines (SVM) as well as rule-based approaches in classifying emails. They conclude that SVM typically outperforms most classification algorithms. In addition to classification approaches used, the authors discuss the importance of preprocessing emails as well as the various metrics for evaluating classification algorithms.

[5] discusses the effect of feature selection methods on machine learning classifiers for detecting email spam. In understanding the effect of feature selection methods, the authors describe in detail some classification approaches which include SVM, Naïve Bayes, Bayesian Classifiers and Genetic Classifiers. They present tables showing the accuracies achieved from these approaches from previous work and these reveal that SVM always outperforms the other algorithms in terms of accuracy.

From their test of the effect of feature selection on classifiers, the authors also perform evaluation of the various classification approaches used. They achieved results similar to earlier work they reviewed even after trying different feature selection approaches like genetic and greedy step-wise feature selection. SVM and Naïve Bayes yielded great results in terms of accuracies and false positives with SVM out performing Naïve Bayes.

The last paper reviewed is [6]. The authors discuss in-detail and present a comparative analysis of the various supervised learning techniques used in Spam classification. The paper is divided into dataset description section, machine learning methods and results section.

In the machine learning methods section, the authors discuss in detail three classification algorithms. The first is multilayer perceptron which is a feed forward artificial neural network model which maps a set of input data onto a set of appropriate output [6]. Multilayer perceptron uses back propagation algorithm which is a supervised learning pattern recognition process. The next algorithm they examine is K-Nearest Neighbour which uses a similarity measure like a distance function to classify new data points. The authors examine SVM.

Finally, the authors evaluate these algorithms by looking at the computational time, accuracy and error rate. The results revealed SVM and K-Nearest Neighbour to have the highest accuracy and lowest score of 77% and 22% respectively. However, SVM recorded the highest computation time which was 30 milliseconds slower than multilayer perceptron. K-Nearest Neighbour fell between SVM and multilayer perceptron scoring average points between the two.

1.3 Significance

The above research work reveals that building an email classification system can be grouped into the following activities building the dataset; preprocessing the email data, building a learning and classification system as well as a system to interact between the email client of the user and the classification system.

In addition, the research work also reveals that supervised learning approaches particularly Support Vector Machine and Naïve Bayes usually produces better results than unsupervised learning approaches in text classification.

Chapter 2: Requirements

2.1 Overview

In this chapter, a detailed description of how requirements are gathered and analyzed to produce specified requirements is presented. A use case and the scope of the project is also presented. As introduced, this project seeks to build an automated email classification system that would categorize emails the Ashesi support center receives as either an IT or Operations email. To ensure the project achieves its aim, the requirements need to be validated by the intended users.

2.2 User Identification and Requirement Gathering

This section focuses on how requirements for the system was gathered as well as how the user was identified.

2.2.1 User Identification

The email classification system is going to be used by the support personnel at the Ashesi support center.

2.2.2 Requirement Gathering

Data collected for determining user requirements was done mainly through interviews. Interviews in this project was done in two ways: in-person interviews where a one-on-one conversation was held with a support personnel and through text messages with the support personnel. Guidelines for the interviews are presented in the Appendix A.

2.3 Requirement Analysis

This section details out the process used to determine what users expect from the system.

The interviews conducted shed light on how the support center categorizes problems they receive. From the interview, the following insights were discovered:

1. The support center either emails reporting an issue or an enquiry for information
2. These reports and enquires are grouped as either IT or Operations Related.
3. The support center has no way of classifying emails. Hence, a support personnel manually sifts through the emails are places them in the appropriate folder.
4. Depending on the problem, the support center either addresses the email or forwards it to an appropriate responder.
5. The support center uses Microsoft Outlook as their emailing client.

From these insights, the needs of the user and system requirements were identified. Below are some of the needs that were identified:

1. The Support Centre needs a system they could classify emails. For example, putting all emails relating to IT in an IT folder automatically.
2. The support center needs to determine an appropriate responder to an email.

2.3.1 Use Case

A use case of the user interacting with the system is presented below.

Once a user receives a new email. The system automatically captures the mail and forwards it to the classification system. There, the system determines if the mail is either IT, Operations or neither of the two and communicates the result back to the user. Figure 2.0 illustrates this procedure.

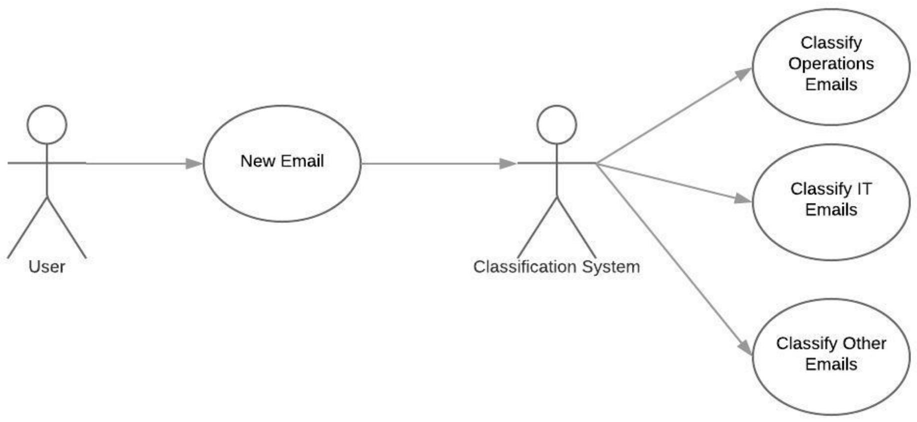


Figure 2.0 Use Case Diagram

2.5 Requirement Specification

2.5.1 Functional Requirements

1. The user must be able to view new classified IT emails in the IT folder of their email client.
2. The user must be able to view new classified Operations emails in the Operations folder of their email client.
3. The user must be able to view non-IT and Operations emails in the Other folder of their email client.
4. The classification system must be able to receive new email support email the client receives automatically.
5. The classification system must be able to assign a category to an email it receives.
6. Upon Classification the system should automatically be able to place new email in the appropriate folder

2.5.2 Non-Functional Requirements

1. Security

- Since the system works with private user emails, the system should have security features in place to ensure there is no breach of confidentiality

2. Availability

- The system should be available and work always.

3. Integrity

- The system must ensure that always, the information provided is accurate, consistent, and reflective of the user's data.

4. Integration

- The system should integrate perfectly with user's email client.

2.6 Scope of project

From the identified needs, the project would consist mainly of two parts: a client-side system that is responsible for sending a new email to the classification system and placing the new email an appropriate folder as well as a classification system that is responsible for classifying the email.

Chapter 3: Architecture and System Design

3.1 Overview

This chapter presents a detailed layout of the system architecture as well as a description of the system design and key components.

3.2 High Level Architecture

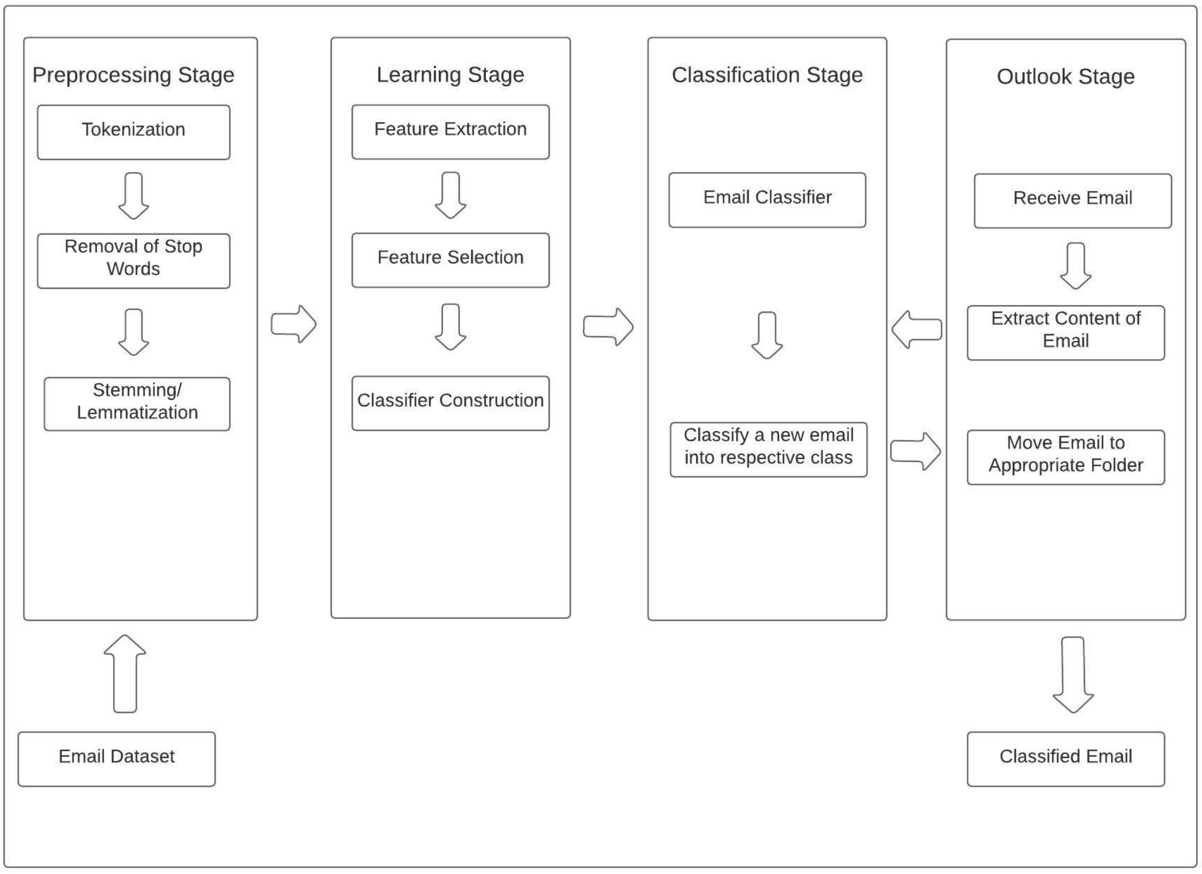


Figure 3.0 High level Architecture

The high-level architecture used can be found in figure 3.0 and is a modified version of the system architecture used in [5].

3.3 Key Components

In building an email classification system, [4] and [5] separate the system into 4 different components: the dataset used, the preprocessing component, the learning component and classification component. In addition to these components, an outlook component would be added to retrieve and forward new emails to the classification as well as place classified emails in their appropriate folder.

The significance of such an approach is that component can easily be worked on and improved without affecting the overall functionality of the system.

3.3.1 Data Selection

Every classification system requires data that is used to train the system. Hence the first step authors in [3,4,5] before is the selection of a dataset that is used to train their system. Also, the authors perform labelling of the data if it required.

3.3.2 Preprocessing Component

Emails are either sent in a plain text format or html format. They also usually come with headers like sender, receipt address, sensitivity and other headers. Therefore, it is necessary to extract the body of the message which is the actual payload of the email. In building an email classification [1, 2, 3, 4] include tokenization, lemmatization or stemming, removal of stop words and symbols as part of the email preprocessing process. Preprocessing is the first step to preparing data for our training model.

Tokenization involves the extraction of words (token) from the email body. When tokenizing, punctuation marks and other symbols are removed. The document is also converted to lower case.

Another process in preprocessing is the removal of stop words. Stop words include prepositions, conjunctions and pronouns. These words are often removed because it is assumed no information is lost by their absence. Example of stop words are (a, the, at and and).

The conversion of words to their root from is another process in preprocessing. It is achieved through either stemming or lemmatization. In stemming, the root of a word is gained through the removal of the suffix of the word. An example in converting “studying” to its root form, the suffix “ing” is removed to achieve “study”. Another example is “trouble” becomes “troubl”.

Lemmatization is similar to stemming except the lemmatization ensure the proper form of the word is obtained. Example whereas stemming produced “troubl” from “trouble”, lemmatization would ensure that we obtain “trouble”. However, it is computationally expensive and requires a dataset of words and their tenses.

After performing these operations, we are left with a stripped-down version of the data which is forwarded the learning component of the system.

This component will be built using the natural language tool kit (NLTK) library in python as it contains the necessary tools to build a preprocessing component [8]. Figure 3.1 shows the structure of the preprocessing system.

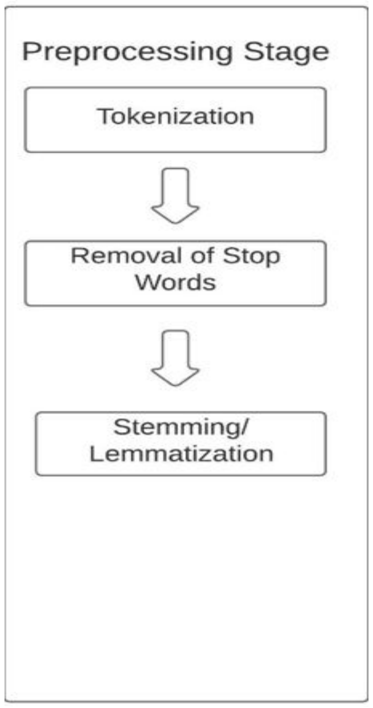


Figure 3.1 Preprocessing Stage Structure

3.3.3 Learning and Classification Component

After preprocessing the emails in the dataset, the next component in the architecture deals with extracting and selecting features that would be used to train our model.

Preprocessing leaves us with a stripped-down version of our work where stop words, punctuation marks and symbols have been removed. However, the classification algorithm cannot work with words in their plain form. Hence, there is a need to convert these words into a numerical representation which the learning model can work on.

One of the commonest ways of representing words is an attribute vector [5]. An attribute is a word in the email. In an attribute vector, words are assigned a numerical value, which could either be a count of the words, 1 or 0 indicating the presence or absence of a word or a weighted score of the word. This process is applied on every email in the dataset to obtain a set features for the classifier.

The system uses Term Frequency times Inverse Document Frequency (TF-IDF) Vectorizer a for its feature extraction process.

TF-IDF Vectorizer is used to assign weights to each word in the email. TF works by taking the counts of each word and assigning weight to each word based on the count. However, to ensure that words that appear many times in a class and hence are less informative do not mask words which appear less TF-IDF also scale down their impact.

Next, feature selection is used to ensure that the best features are selected. It also helps to shorten training times and simply the model. Feature selection is categorized into three approaches: filter, wrapper and embedded methods. In this project, the embedded approach particularly L2 regularization is used. L2 regularization reduces the value of weights by applying a penalty term. This would ensure that features without high weights would not get. In addition, L2 unlike L1 regularization does not lead to features having a weight of 0 which is essential because the dataset is not overly large.

After obtaining a set of features to build our model. The next component is the classification component which uses features to build a model to predict the class of new messages. As mentioned early in the related work section, the classification component can either be built using supervised or unsupervised learning approach. Research by [1, 4, 5] shows that the supervised learning techniques particularly Support Vector Machine and Naïve Bayes achieved great results. The learning and classification component of the system would be built using Scikit-learn library.

Figure 3.2 below shows the structure of the learning component.

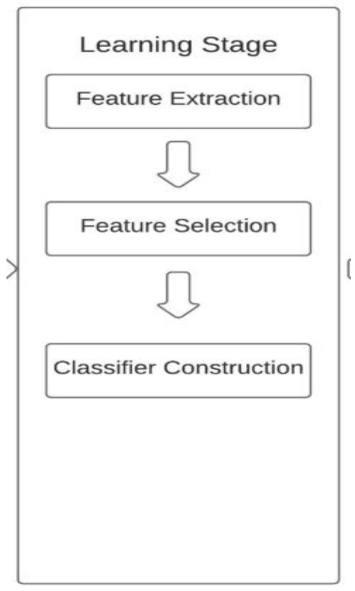


Figure 3.2 Learning Component Structure

After construction of the model, it can be used to predict the class of new emails that are fed into system. Figure 3.3 shows the structure of the classification stage.

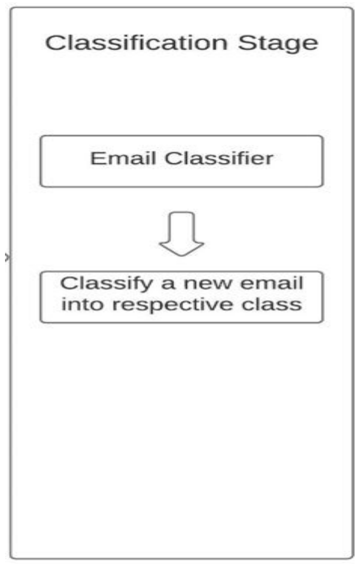


Figure 3.3 Classification Stage Structure

3.3.4 Outlook Component

To obtain new emails from the support inbox, the system would include an outlook component that would be embedded in the Microsoft outlook to obtain new mails on arrival and forward to the classification component. This component would also be responsible for placing classified emails in their appropriate folder.

This component will be built using Visual Basic for Applications (VBA) and will be embedded in the user's outlook mail client.

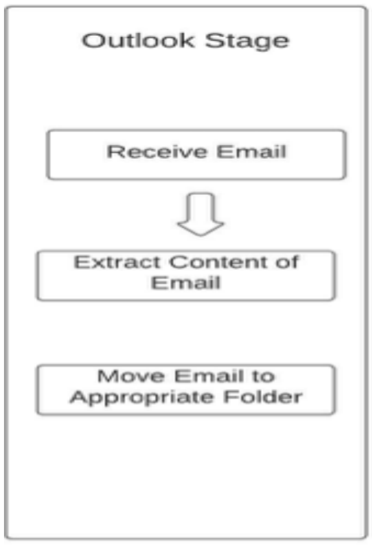


Figure 3.4 Outlook Stage Structure

3.4 Design

This section focuses on the design approach used for the development of the email classification system. Sequence and activity diagrams are also presented to show how key components interact as well as the various processes involved in classifying the user's emails. Additionally, these diagrams provide an abstraction of how the system performs the entire process of classification.

3.4.1 Sequence Diagram

Sequence diagrams provide an abstraction of how key components interact to achieve the goal of classification. In figure 3.5, the sequence diagram shows the series of interactions between the key components. When a user receives email, the email is forwarded to the classification system by the outlook system. The system calls the preprocessing system which in turn retrieves the dataset. The dataset is preprocessed and sent back to the classification which uses the data to learn. The classification system then predicts the class of the email and sends a response back to the outlook system.

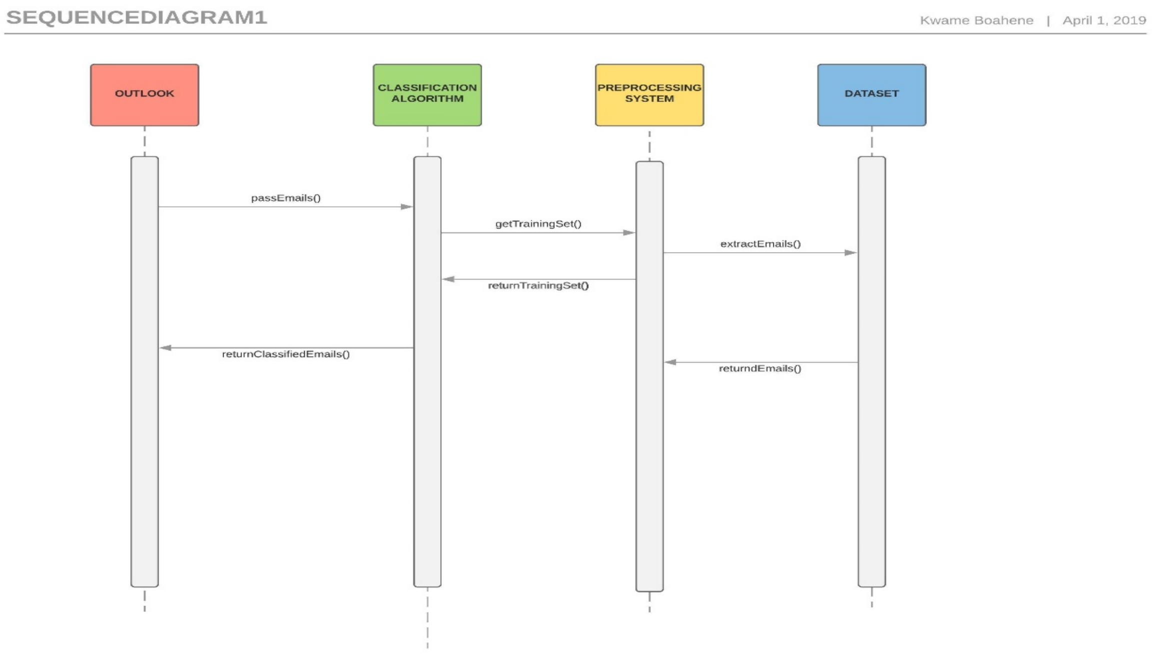


Figure 3.5 Sequence Diagram for the Classification System

Figure 3.5 depicts an illustration of how the various system components interact with each other to achieve the goal of classification.

3.4.2 Activity Diagram

In figure 3.6, a high-level activity diagram illustrates the various processes the classification system goes through to classify user emails.

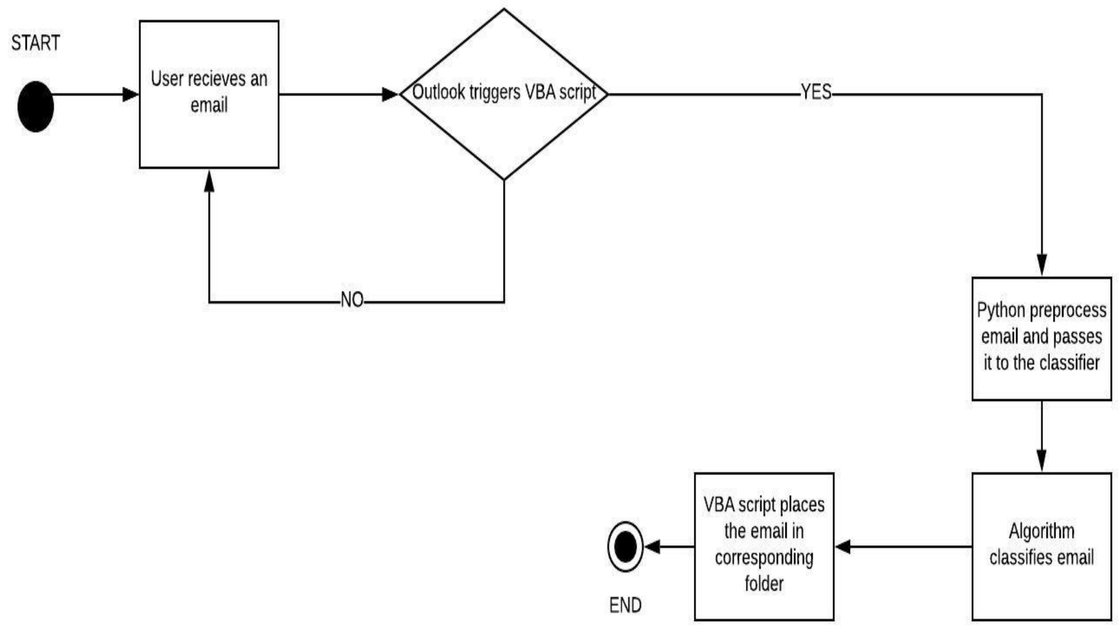

START
User recieves an email
Outlook triggers VBA script
YES
NO
Python preprocess email and passes it to the classifier
Algorithm classifies email
VBA script places the email in corresponding folder
END

Figure 3.6 Activity Diagram for the classification system

Chapter 4: Implementation

4.1 Overview

This chapter describes in detail the various processes involved in implementing the automated email classification system. The chapter is divided into 4 major sections: the building of the dataset used, the preprocessing stage, the classification system and the outlook system. In each section, any library that is used is described, essential algorithms and procedures that are used are emphasized and screenshots as well as diagrams are also used to show the overall implementation.

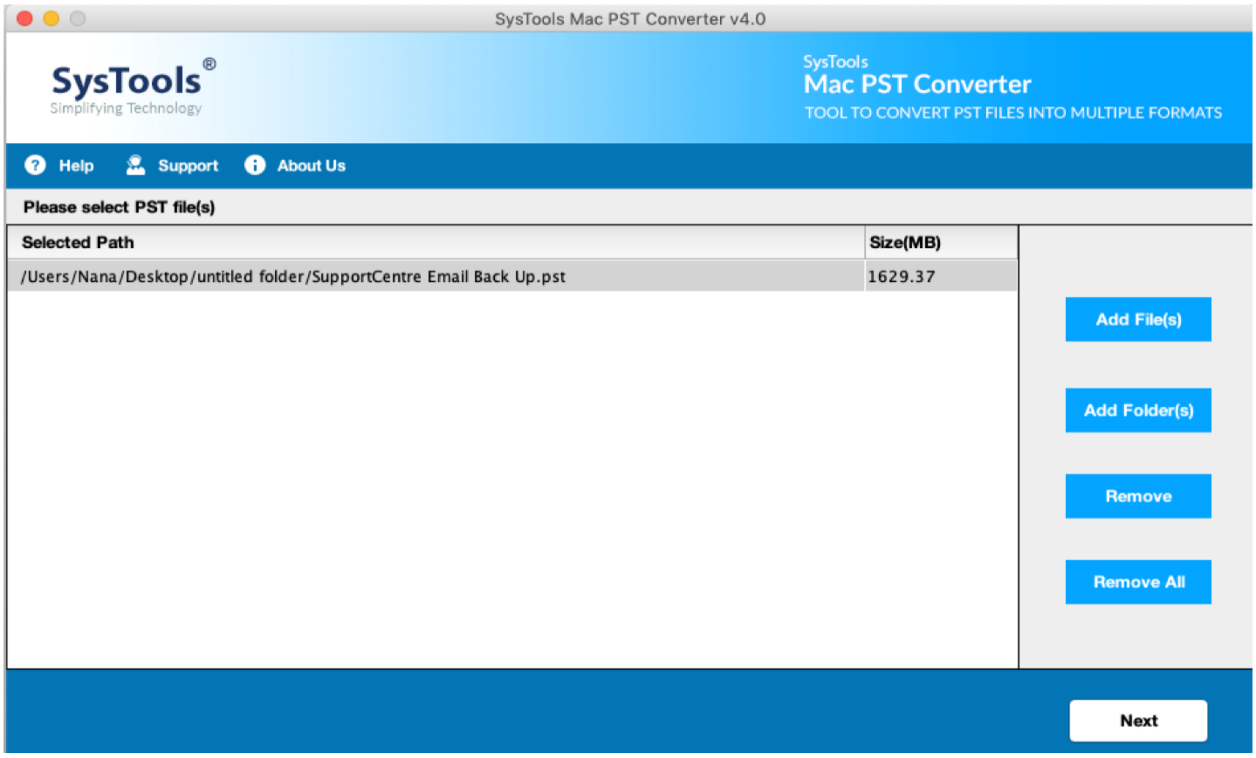
4.2 Dataset

This section focuses on how the dataset used by the classifier was built. In the section, the format, extraction process, libraries and tools used to achieve the final dataset used by the classifier is detailed out.

4.2.1 The Data

The system implemented is going to be used by the Ashesi Support Center. Hence, the email data used for this project was provided by the Ashesi Support Center. The data contained emails received by the support center in their outlook email client which is the main point the Ashesi Support Center uses to receives problems and complaints.

The data was in the format of a Personal Storage Table (.pst) which is a file format used in the Microsoft outlook system. Therefore, there was a need to convert it into a readable format that can be worked on. Hence, a PST converter tool was used to convert the PST file into a folder of pdf files containing the emails. The tool used is called SysTools PST Converter and a screenshot of the application interface is shown in Figure 4.1 below



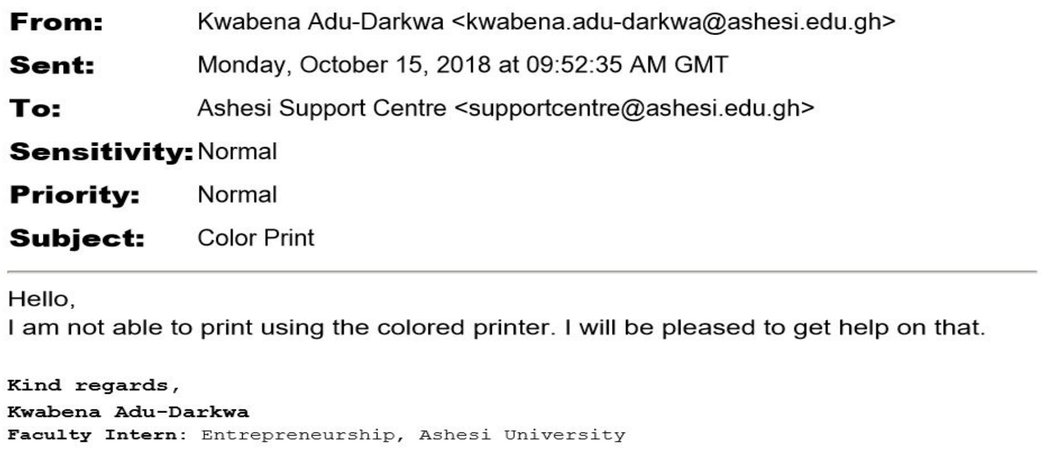
Selected Path	Size(MB)
/Users/Nana/Desktop/untitled folder/SupportCentre Email Back Up.pst	1629.37

Figure 4.1 SysTool Converter

The converted files contained over 6000 pdf files containing emails. The file structure of an email pdf is presented in Table 1 as well as an image of an email pdf is presented in Figure 4.2 below.

Row Name:	From	Sent	To	Sensitivity	Priority	Subject	Body
Description:	Sender Address	Date and Time	Recipient Address	Message Sensitivity	Message Priority	Subject of the Email	Body of the Email

Table 1


From: Kwabena Adu-Darkwa <kwabena.adu-darkwa@ashesi.edu.gh>
Sent: Monday, October 15, 2018 at 09:52:35 AM GMT
To: Ashesi Support Centre <supportcentre@ashesi.edu.gh>
Sensitivity: Normal
Priority: Normal
Subject: Color Print

Hello,
I am not able to print using the colored printer. I will be pleased to get help on that.

Kind regards,
Kwabena Adu-Darkwa
Faculty Intern: Entrepreneurship, Ashesi University

Figure 4.2 Email Structure

4.2.2 Data Extraction

Given the file structure of the pdf, there was a need to extract the needed part of the email and do away with the unnecessary headers. To achieve this, the python library PyMuPDF was used.

PyMuPDF is a python rendition of the MuPDF. MuPDF is a lightweight pdf, xps and e-book viewer [7]. It allows uses to also render graphics and can be used to edit pdf, xps and other document formats.

For this project, PyMuPDF was used to open the pdf files and extract the necessary parts of the document. The sender, subject and body of the email was extracted and stored in a json format.

4.2.2.1 Manual Labelling

The extracted pdf files were not grouped as IT, Operations or Other emails. Therefore, there was a need to manual sift through the emails and group them as either IT, Operations or

Other. This stage was necessary since the implemented system used a supervised learning approach and requires previously trained data.

After, PyMuPDF was used to extract the necessary contents of the emails. After the scripts below were used to extract the necessary parts of emails.

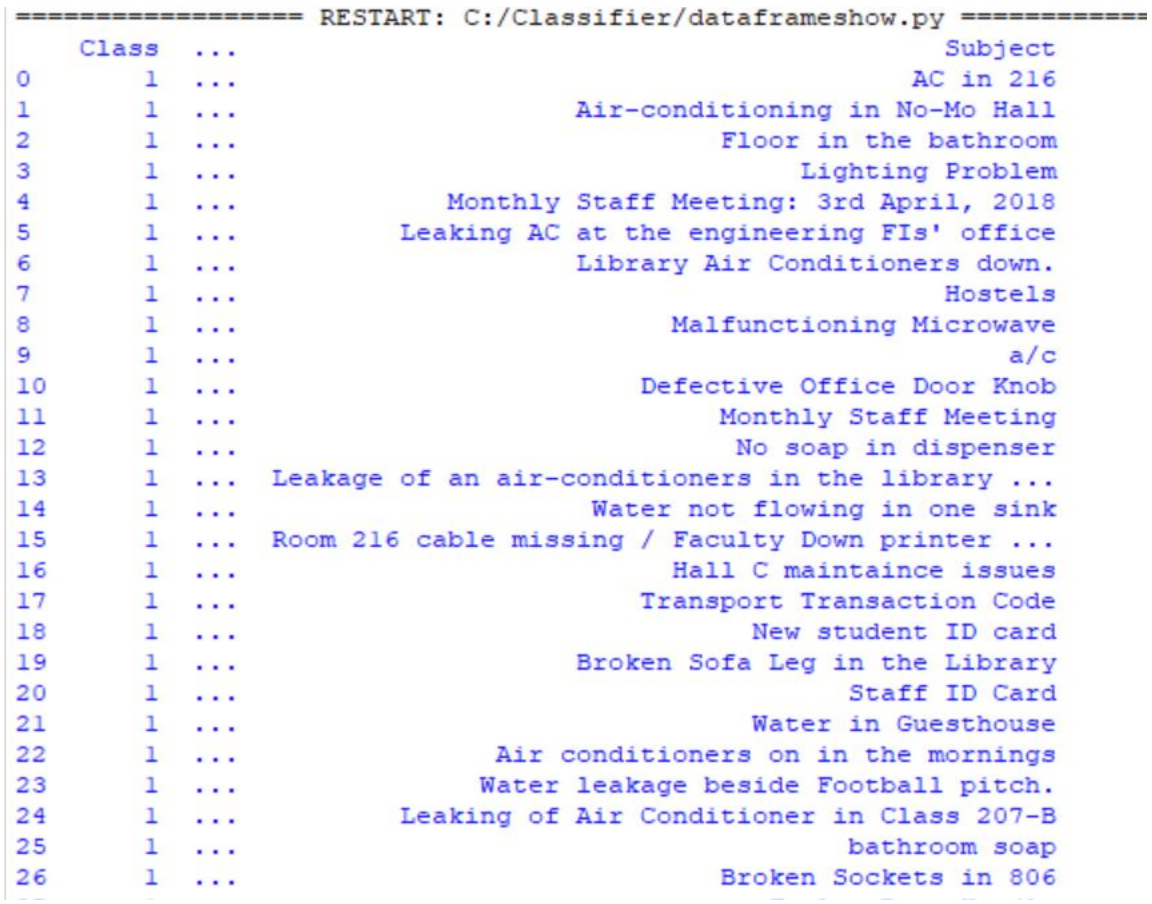
1. "**pdfreaderIT.py**" was used to extract the sender, body and subject parts of the email for the IT emails and output json file "**IT.json**" containing the extracted content for all emails in the IT folder
2. "**pdfreaderOperations.py**" was used to extract the sender, body and subject parts of the email for the Operations emails and output json file "**Operations.json**" containing the extracted content for all emails in the Operations folder.
3. "**pdfreaderOther.py**" was used to extract the sender, body and subject parts of the email for the Operations emails and output json file "**Other.json**" containing the extracted content for all emails in the Other folder.

The subject was extracted to be used as the body in case the sender left the body of the email blank.

4.3 Preprocessing Stage

As highlighted in the previous chapter, preprocessing of the dataset is a vital step in building the classification system. Therefore, to preprocess the dataset, the Natural Language Toolkit (NTLK) was used. NLTK contains the various functionalities and libraries that are suited for all the stages of preprocessing like tokenization, lemmatization, stop words removal and stemming [8].

The first step in the preprocessing stage was to extract and place the dataset into a data frame to make it easy to work on all the emails. This was achieved by using pandas. Pandas is a python library built for data analysis and manipulation [11]. Using pandas allows for easy application of functions on large datasets. Figure 4.3 shows the email data extract in a panda data frame.



```
================== RESTART: C:/Classifier/dataframeshow.py ============
    Class  ...                                             Subject
0       1  ...                                          AC in 216
1       1  ...                      Air-conditioning in No-Mo Hall
2       1  ...                                 Floor in the bathroom
3       1  ...                                      Lighting Problem
4       1  ...                 Monthly Staff Meeting: 3rd April, 2018
5       1  ...               Leaking AC at the engineering FIs' office
6       1  ...                      Library Air Conditioners down.
7       1  ...                                               Hostels
8       1  ...                             Malfunctioning Microwave
9       1  ...                                                   a/c
10      1  ...                            Defective Office Door Knob
11      1  ...                                 Monthly Staff Meeting
12      1  ...                                 No soap in dispenser
13      1  ...  Leakage of an air-conditioners in the library ...
14      1  ...                           Water not flowing in one sink
15      1  ...  Room 216 cable missing / Faculty Down printer ...
16      1  ...                                Hall C maintaince issues
17      1  ...                            Transport Transaction Code
18      1  ...                                     New student ID card
19      1  ...                          Broken Sofa Leg in the Library
20      1  ...                                          Staff ID Card
21      1  ...                                   Water in Guesthouse
22      1  ...                    Air conditioners on in the mornings
23      1  ...                     Water leakage beside Football pitch.
24      1  ...                Leaking of Air Conditioner in Class 207-B
25      1  ...                                          bathroom soap
26      1  ...                                     Broken Sockets in 806
```

Figure 4.3 Extracted Email Data

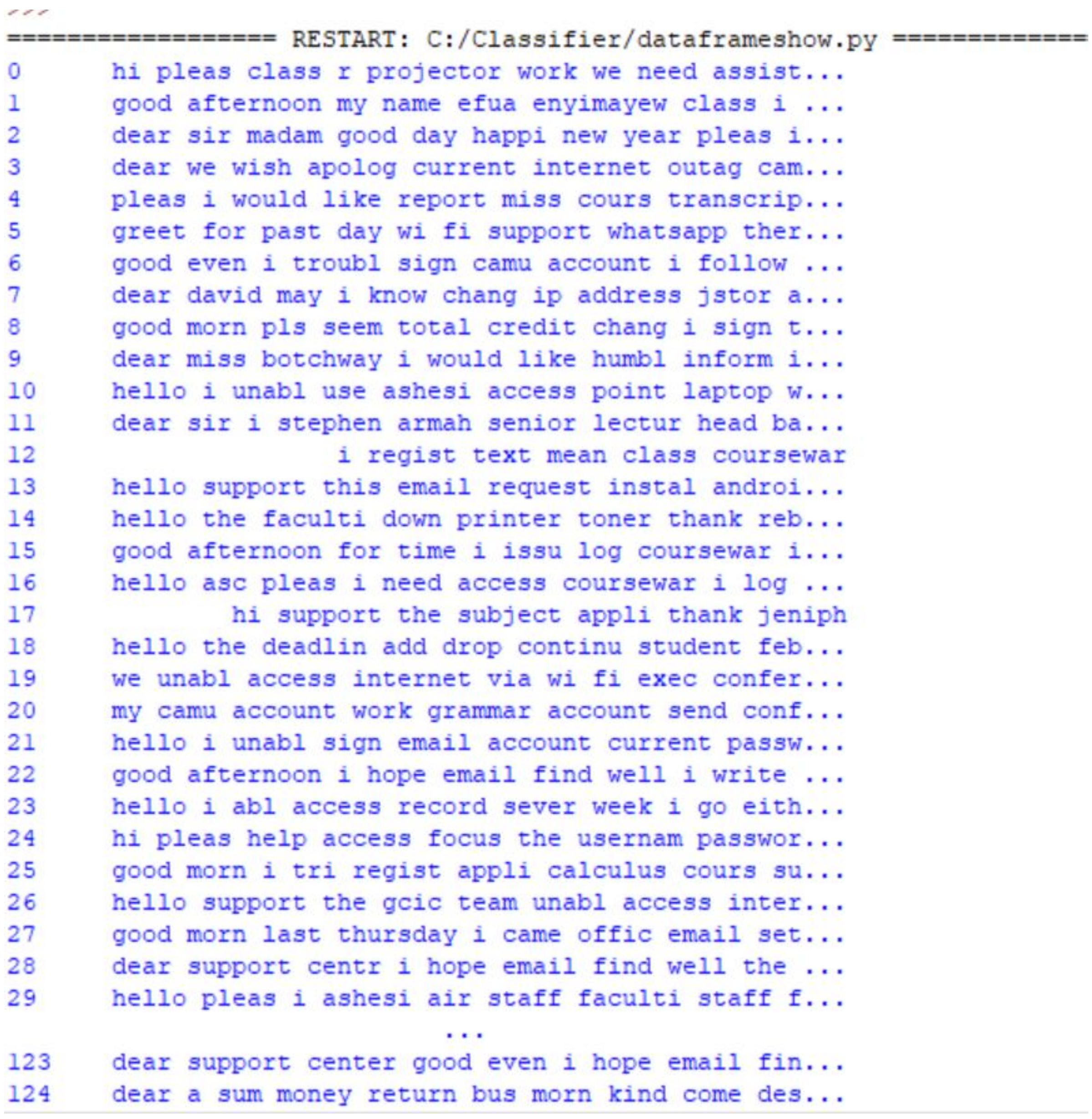
Next in the preprocessing stage is tokenization. As described in the previous chapters, tokenization involves taking a string of text and splitting it into individual substrings. This was achieved by using the split function in python.

After tokenization, the regular expression library in python “re” was used to split each token and remove all punctuations, numbers and symbols from them.

The removal of all stop words was the next step that was undertaken. NLTK libraries has a list of all stop words as one of the resources in the library. Hence, a function was created to remove the tokenized words that match words in the stop words list.

The final step in the processing stage was stemming. As described in the previous chapter. Stemming is used to convert words to their root form by removing suffixes from the words. Stemming was accomplished by using the stemming functionality in the NLTK toolkit.

After all these processes a new column was added to our data frame containing the processed text. The results are displayed in Figure 4.4 below



```
===================== RESTART: C:/Classifier/dataframeshow.py =============
0      hi pleas class r projector work we need assist...
1      good afternoon my name efua enyimayew class i ...
2      dear sir madam good day happi new year pleas i...
3      dear we wish apolog current internet outag cam...
4      pleas i would like report miss cours transcrip...
5      greet for past day wi fi support whatsapp ther...
6      good even i troubl sign camu account i follow ...
7      dear david may i know chang ip address jstor a...
8      good morn pls seem total credit chang i sign t...
9      dear miss botchway i would like humbl inform i...
10     hello i unabl use ashesi access point laptop w...
11     dear sir i stephen armah senior lectur head ba...
12                    i regist text mean class coursewar
13     hello support this email request instal androi...
14     hello the faculti down printer toner thank reb...
15     good afternoon for time i issu log coursewar i...
16     hello asc pleas i need access coursewar i log ...
17             hi support the subject appli thank jeniph
18     hello the deadlin add drop continu student feb...
19     we unabl access internet via wi fi exec confer...
20     my camu account work grammar account send conf...
21     hello i unabl sign email account current passw...
22     good afternoon i hope email find well i write ...
23     hello i abl access record sever week i go eith...
24     hi pleas help access focus the usernam passwor...
25     good morn i tri regist appli calculus cours su...
26     hello support the gcic team unabl access inter...
27     good morn last thursday i came offic email set...
28     dear support centr i hope email find well the ...
29     hello pleas i ashesi air staff faculti staff f...
                             ...
123    dear support center good even i hope email fin...
124    dear a sum money return bus morn kind come des...
```

Figure 4.4 Preprocessed Email Data

4.4 Learning and Classification Stage

As described in the architecture and systems design chapter, another component of the email classification system is the learning and classification component. The learning component uses already classified email data to train a model which is used to classify new emails. This section of the implementation focuses on how the learning and classification components are built. All aspects of these section are built using Scikit-learn library in python. Scikit-learn allows users to perform a wide range machine learning tasks such as classification in python [10].

After preprocessing the emails, the next step is the learning stage. The learning stage comprises of feature extraction, feature selection and the construction of the classifier. Even after preprocessing, the emails in its raw form cannot be fed into a learning algorithm. This is because the learning algorithms expect feature vectors of a fixed size rather than words in the emails. Hence, there is a need to extract features. A feature is a word that has been assigned a weight. This process is known as feature extraction.

4.4.1 Feature Extraction

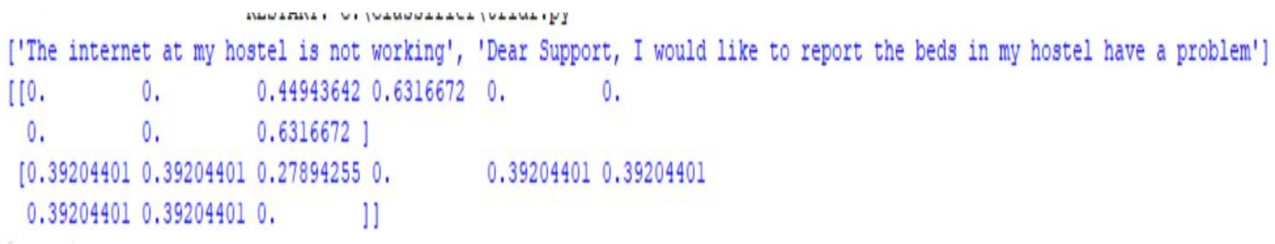
There are several models that allow users to perform feature extraction. Examples are Glove, the bag of words model and word2vec. However, TF-IDF which stands for Term Frequency times Inverse Document-Frequency will be used. As described earlier, in addition to counting the frequency of the words, TF-IDF reduces the effect of words which have little in meaning but appear a lot of times in the document. It achieves this through inverse document frequency. Below is the formula used in TF-IDF

$$TF - IDF = TF * IDF$$

$$TF = \frac{count\ of\ word(w)\ in\ document}{total\ number\ of\ words}$$

$$IDF = \log\left(\frac{total\ number\ of\ documents}{total\ number\ of\ documents\ containing\ word(w)}\right)$$

TF-IDF is implemented using the feature extraction module in Scikit-Learn library. Below in Figure 4.5 is an example of TF-IDF applied to email data.



```
['The internet at my hostel is not working', 'Dear Support, I would like to report the beds in my hostel have a problem']
[[0.         0.         0.44943642 0.6316672  0.         0.
  0.         0.         0.6316672 ]
 [0.39204401 0.39204401 0.27894255 0.         0.39204401 0.39204401
  0.39204401 0.39204401 0.        ]]
```

Figure 4.5 TF-IDF to Email Data

4.4.2 Feature Selection

After extracting features from our emails, the next step is to select the features which best represent each class in the classification system. This procedure is known as Feature Selection.

Feature Selection is essential as it helps to remove attributes which do not improve the accuracy of the predictive model in any way. It also helps to shorten training times and simplify the model. Feature selection is categorized into three approaches: filter, wrapper and embedded methods [9]. Filtering uses statistical measures to assign a score to features; these scores are ranked and features below a specified threshold are removed. The wrapper approach utilizes greedy algorithm to find the best combination of features. Wrapper approach produces good results but can be computationally expensive when large datasets are involved. Embedded approach selects the best features while the model is being built. The most common types are regularization or penalization methods.

In the classification system, L2 regularization is embedded with Support Vector Machine and Naïve Bayes as our feature selection approach. Although L2 does not remove features, it reduces their weights significantly keeping the model simple whiles reducing overfitting. L2 regularization is best suited here because our dataset is not overly large as such, using L1 regularization would led to many features being removed giving us a sparse solution.

4.4.2 Classifier Construction

As mentioned in early chapters, Naïve Bayes and Support Vector Machines are the classification algorithms going to be used in the system. These two supervised learning algorithms were selected because they performed best according in research done evaluating classification approaches for text classification. These systems are implemented using the Sci Kit library in python. Sci Kit offers arrange parameters we can specify on Naïve bayes and SVM that allows us to tune the algorithm to yield optimum results.

For SVM, the regularization algorithm was set to L2. The loss function which is responsible for measuring the penalty applied to the weights is set to “hinge” which is the standard of SVM. The error term, C, is used to specify the amount of error the classifier is willing to take when training the data. This particularly useful when there are outliers in the training data. Multi Class is used to specify the strategy used when there are more than two classes in the dataset. Here, One-vs-rest strategy is used which trains a single classifier per class in the dataset and it uses sample data of that class as positive samples and the other classes as negative samples. Max Iterations is also a parameter and it specifies the number of iterations to be run. By default, it is set to 1000 which is used.

Multinomial Naïve Bayes was used as our second classification algorithm. Naïve Bayes works well for classification involving discrete features like word counts for text classification. Multinomial Naïve Bayes in Sci Kit allows the user to specify three parameters. Alpha parameter specifies the smoothing value which by default is 1.0. However, it was changed to 0.01 since 1 is a high weight value for words not in our dataset. The parameter, fit prior, determines whether the class prior probability should be learnt. This is set to true. The last parameter allows the user to set their own class prior probability. Since, the algorithm learns it, the value is set to “None”. Additional information on the classifiers can be found in the Sci Kit Library.

4.5 Outlook System

The final component of our system is the outlook component. This system resides in the user’s email client and performs the functions of extracting the content of a new email the user receives, forwarding it to the classification system and placing the email in the appropriate folder based on the response from the classification system. This component automates the entire classification process. As mentioned earlier, this system is implemented using Visual Basic for Applications (VBA), which is an event-driven programming language used in Microsoft applications like Outlook.

The component is made up of a script which is embedded into the emailing client’s session. The first part of the script makes use of in-built functionalities to grab a new email immediately it appears in the user’s inbox. The email is presented as an object and the body can be extracted using “.body” notation on the object. After, extracting the body from the next step is to remove all new line characters to form a single string.

Next, VBA provides functionality to execute shell command. This is used to pass the email body to the classifier which predicts the class the email belongs to. The classifier prints out the respective class to the shell output which is captured and sent back to the VBA script. Finally, the script uses the output to move the email to its predicted folder.

Chapter 5: Testing and Results

5.1 Overview

This chapter describes how the system was tested to ensure all requirements have been met. It details out the testing approaches used as well as the results from these tests.

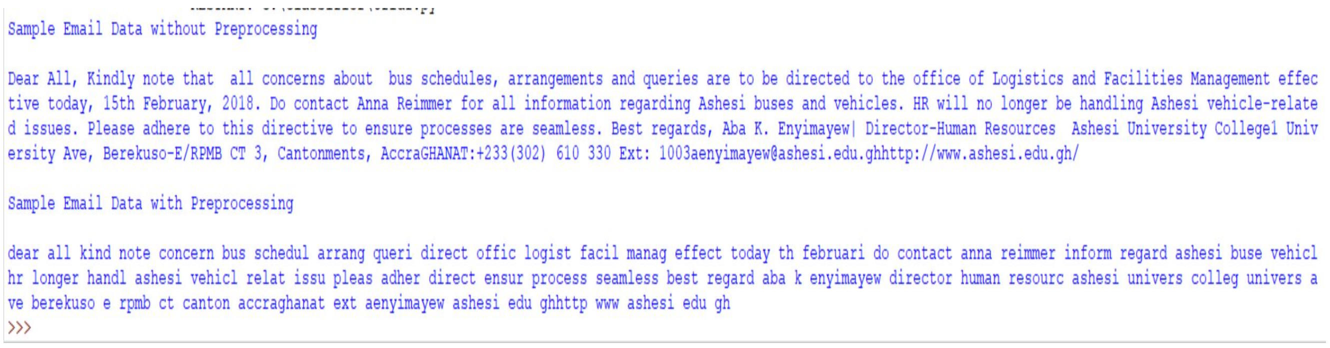
5.2 Test Description

Since this is an automated system and would run in the background without direct interact with the user, there is a need to do comprehensive testing of all system components. As such, unit testing of each system component was done as well as system testing which walks through the entire process of classifying new emails users receive.

5.3 Unit Testing

5.3.1 Preprocessing Component Test

The preprocessing system is responsible for tokenization, removing stop words, punctuation marks, symbols, converting all characters to lower case and stemming of the words in the email data. Below in Figure 5.1 preprocessing test is done to validate that the preprocessing system works.



```
Sample Email Data without Preprocessing

Dear All, Kindly note that  all concerns about  bus schedules, arrangements and queries are to be directed to the office of Logistics and Facilities Management effec
tive today, 15th February, 2018. Do contact Anna Reimmer for all information regarding Ashesi buses and vehicles. HR will no longer be handling Ashesi vehicle-relate
d issues. Please adhere to this directive to ensure processes are seamless. Best regards, Aba K. Enyimayew| Director-Human Resources  Ashesi University College1 Univ
ersity Ave, Berekuso-E/RPMB CT 3, Cantonments, AccraGHANAT:+233(302) 610 330 Ext: 1003aenyimayew@ashesi.edu.ghhttp://www.ashesi.edu.gh/

Sample Email Data with Preprocessing

dear all kind note concern bus schedul arrang queri direct offic logist facil manag effect today th februari do contact anna reimmer inform regard ashesi buse vehicl
hr longer handl ashesi vehicl relat issu pleas adher direct ensur process seamless best regard aba k enyimayew director human resourc ashesi univers colleg univers a
ve berekuso e rpmb ct canton accraghanat ext aenyimayew ashesi edu ghhttp www ashesi edu gh
>>>
```

Figure 5.1 Preprocessing Data

From Figure 5.1, we can see the number of words has reduced considerably, this is because the preprocessing component has been able to successfully remove stop words, symbols, numbers and punctuation marks. Also, the preprocessing component has been able convert all characters to lower case and it successfully applied stemming to convert words to their root forms.

5.3.2 Learning and Classification Component Testing

The learning and classification component are responsible for building a model that can classify newer emails utilizing the email dataset provided. It is made up of a feature extraction and selection models as well as a classification algorithm. Two models were built using SVM and multinomial Naïve Bayes. To test these classifiers, their accuracy, precision, recall and f1 scores would be test. Accuracy measures the ability of the model to predict correctly the class of a given test sample. Given a class, precision measures the proportion of predictions for the class that were actually correct. Recall measures for a given class, the ability of the model to predict that test samples belong to the class. F1 finds a balance between recall and precision. This necessary because as precision increases recall falls and vice versa. The formulas to compute these measurements are presented below. Scikit-learn library in python is used to compute the values.

$$TP = True\ Positive \quad FP = False\ Positive$$

$$FN = False\ Negative \quad TN = True\ Negative$$

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

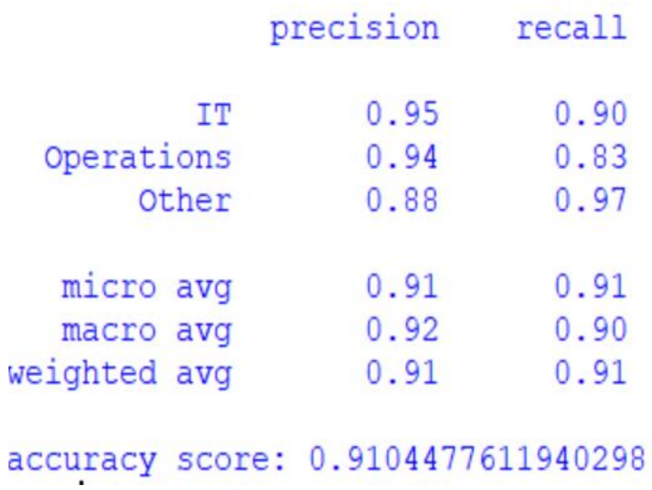
$$Precision = \frac{(TP)}{(TP + FP)}$$

$$Recall = \frac{(TP)}{(TP + FN)}$$

$$f1 = 2 * \frac{(Precision * Recall)}{(Precision + Recall)}$$

5.3.2.1 Support Vector Machine

The accuracy, precision, recall and f1 scores for SVM is presented below Figure 5.2



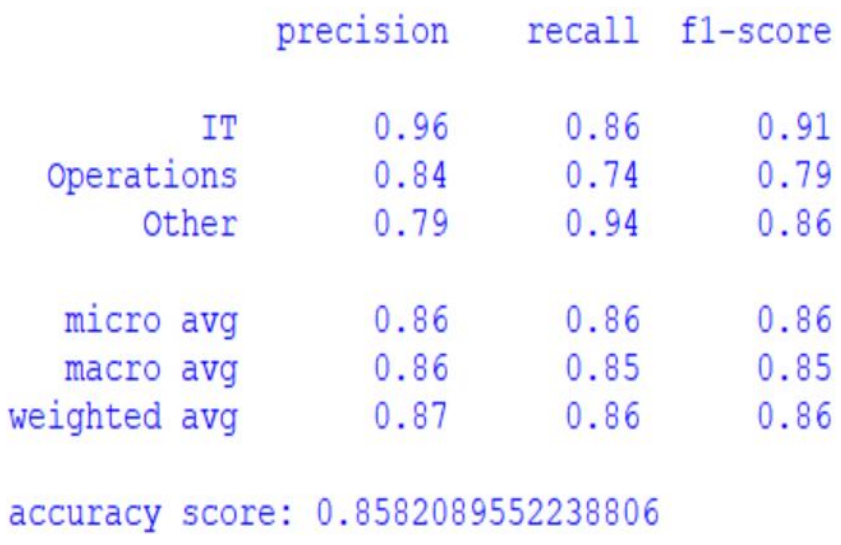
```
              precision    recall  f1-score

          IT       0.95      0.90      0.92
  Operations       0.94      0.83      0.88
       Other       0.88      0.97      0.92

   micro avg       0.91      0.91      0.91
   macro avg       0.92      0.90      0.91
weighted avg       0.91      0.91      0.91

accuracy score: 0.9104477611940298
```

Figure 5.2 Test Scores for SVM

From Figure 5.2, SVM is able to accurate predict most of the test samples passed to it. It scores high points in all testing fields similar to results from research mentioned earlier.

5.3.2.2 Multinomial Naïve Bayes

The accuracy, precision, recall and f1 scores for Naïve Bayes are presented below in Figure 5.3



```
              precision    recall  f1-score

          IT       0.96      0.86      0.91
  Operations       0.84      0.74      0.79
       Other       0.79      0.94      0.86

   micro avg       0.86      0.86      0.86
   macro avg       0.86      0.85      0.85
weighted avg       0.87      0.86      0.86

accuracy score: 0.8582089552238806
```

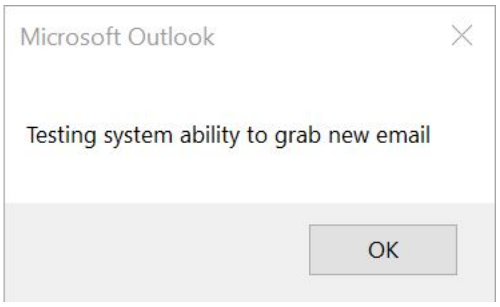
Figure 5.3 Test Scores for Naïve Bayes

From Figure 5.3, Naïve Bayes classifier also performs well however, it is outranked in almost every category by SVM except for precision for the IT class where it slightly beats SVM. Both classifiers perform well in the classifying IT and Other emails. However, more data is needed to improve upon the scores for Operations and Other.

5.3.3 Outlook System

The outlook system places a key role of forwarding new emails to the classification system as well moving the emails to the appropriate folder. The system is automated, and it triggers itself once a new email enters the user's inbox. Below, testing is done to ensure the various functionalities of the system work.

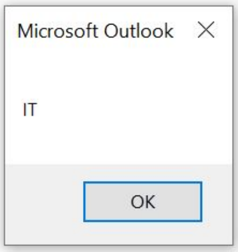
The first test of the system is to check if the system can capture new emails. To test this, the system prints out the email content once it receives it. Figure 5.4 Shows a successfully an extracted email.


Microsoft Outlook

Testing system ability to grab new email

Figure 5.4 Extracted Email

Finally, to ensure that the classification system executes the shell command and returns a class the email belongs in, the class is printed out per the response of the classification system. Figure 5.5 shows the predicted class of an email.


Microsoft Outlook

IT

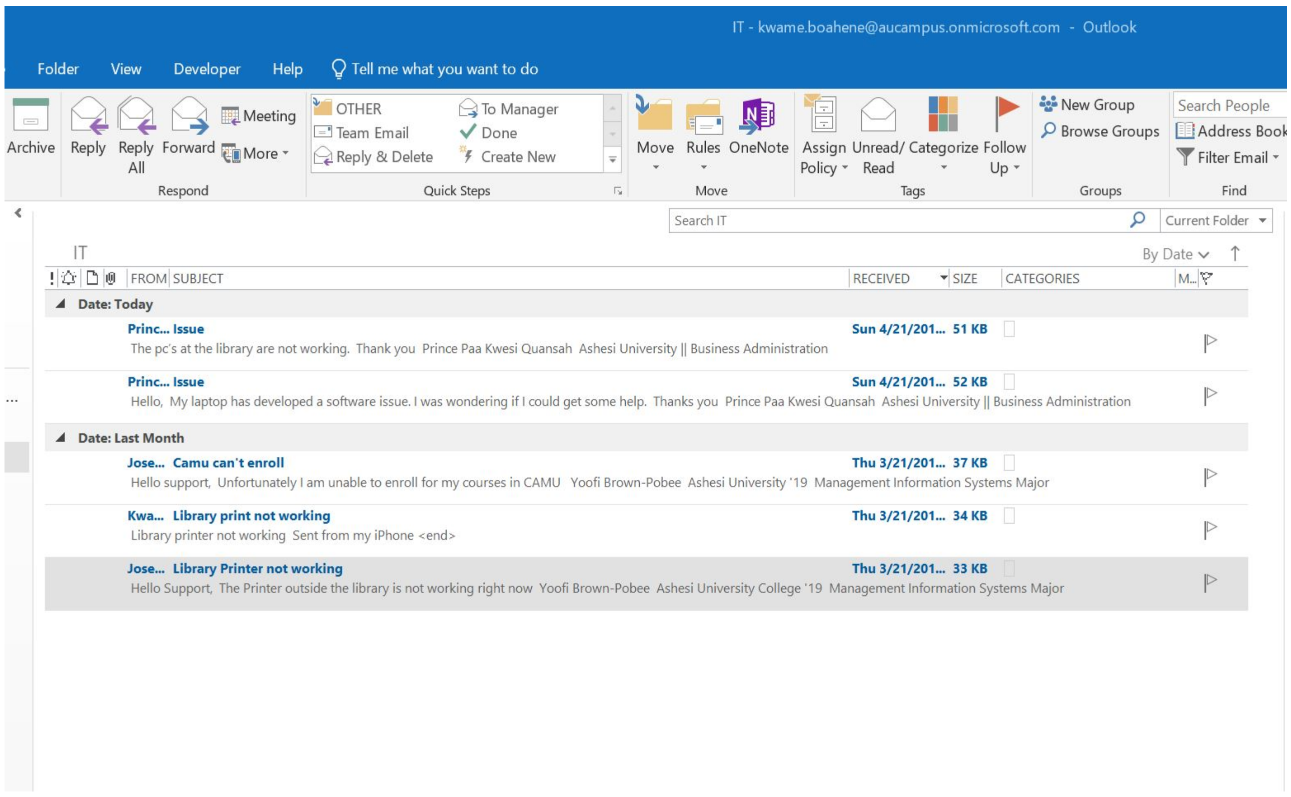
Figure 5.5 Shell Script Response

5.4 System Testing

The unit testing verified that all system components worked properly. However, to ensure the overall system met requirements a system test must be done. To achieve a complete system test, the support system was mimicked by creating IT, Operations and Other email folder in a Microsoft Outlook account and students of the Ashesi community were made to send emails complaints and problems like they normally would to the Support Center. In addition to students, emails that are usually broadcasted from to the entire Ashesi community was also classified by the system.

The SVM classifier was used in these tests as it achieved better recall, precision and accuracy scores. Below are the screenshots of classified emails:

5.4.1 IT Emails

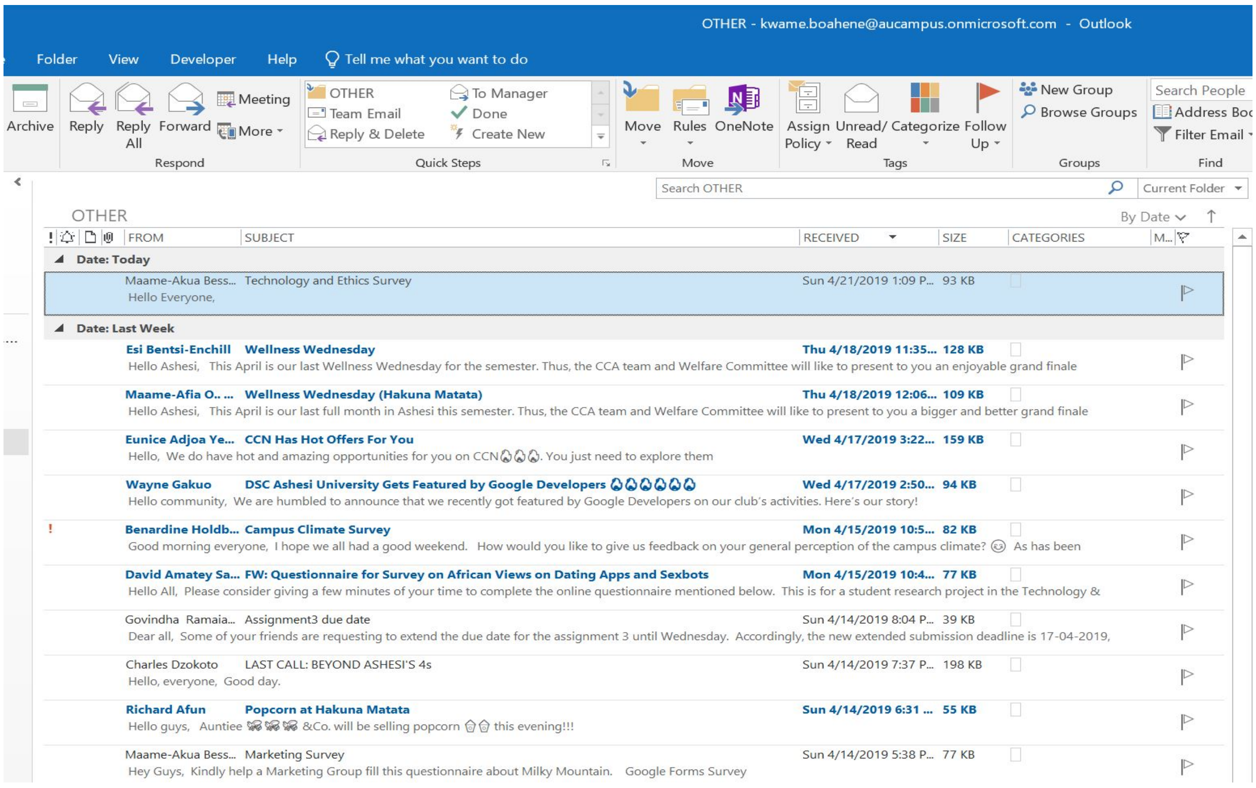


FROM	SUBJECT	RECEIVED	SIZE
Date: Today			
Princ...	Issue The pc's at the library are not working. Thank you Prince Paa Kwesi Quansah Ashesi University || Business Administration	Sun 4/21/201...	51 KB
Princ...	Issue Hello, My laptop has developed a software issue. I was wondering if I could get some help. Thanks you Prince Paa Kwesi Quansah Ashesi University || Business Administration	Sun 4/21/201...	52 KB
Date: Last Month			
Jose...	Camu can't enroll Hello support, Unfortunately I am unable to enroll for my courses in CAMU Yoofi Brown-Pobee Ashesi University '19 Management Information Systems Major	Thu 3/21/201...	37 KB
Kwa...	Library print not working Library printer not working Sent from my iPhone <end>	Thu 3/21/201...	34 KB
Jose...	Library Printer not working Hello Support, The Printer outside the library is not working right now Yoofi Brown-Pobee Ashesi University College '19 Management Information Systems Major	Thu 3/21/201...	33 KB

Figure 5.6 Classified IT Emails

Figure 5.6 shows successfully classified IT email

5.4.2 Other Emails

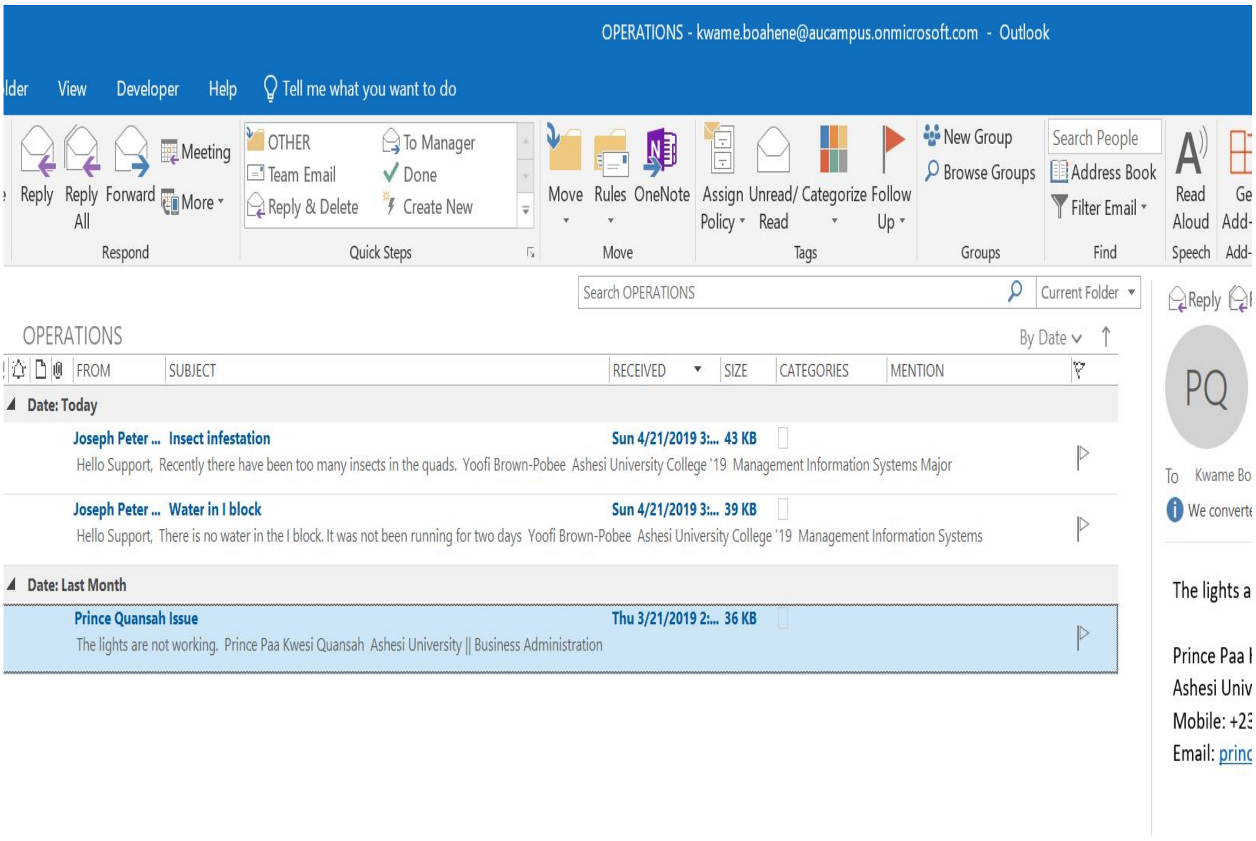


!	FROM	SUBJECT	RECEIVED	SIZE	CATEGORIES
Date: Today					
	Maame-Akua Bess...	Technology and Ethics Survey Hello Everyone,	Sun 4/21/2019 1:09 P...	93 KB	
Date: Last Week					
	Esi Bentsi-Enchill	**Wellness Wednesday** Hello Ashesi, This April is our last Wellness Wednesday for the semester. Thus, the CCA team and Welfare Committee will like to present to you an enjoyable grand finale	**Thu 4/18/2019 11:35...**	**128 KB**	
	Maame-Afia O.. ...	**Wellness Wednesday (Hakuna Matata)** Hello Ashesi, This April is our last full month in Ashesi this semester. Thus, the CCA team and Welfare Committee will like to present to you a bigger and better grand finale	**Thu 4/18/2019 12:06...**	**109 KB**	
	Eunice Adjoa Ye...	**CCN Has Hot Offers For You** Hello, We do have hot and amazing opportunities for you on CCN. You just need to explore them	**Wed 4/17/2019 3:22...**	**159 KB**	
	Wayne Gakuo	**DSC Ashesi University Gets Featured by Google Developers** Hello community, We are humbled to announce that we recently got featured by Google Developers on our club's activities. Here's our story!	**Wed 4/17/2019 2:50...**	**94 KB**	
!	**Benardine Holdb...**	**Campus Climate Survey** Good morning everyone, I hope we all had a good weekend. How would you like to give us feedback on your general perception of the campus climate? As has been	**Mon 4/15/2019 10:5...**	**82 KB**	
	David Amatey Sa...	**FW: Questionnaire for Survey on African Views on Dating Apps and Sexbots** Hello All, Please consider giving a few minutes of your time to complete the online questionnaire mentioned below. This is for a student research project in the Technology &	**Mon 4/15/2019 10:4...**	**77 KB**	
	Govindha Ramaia...	Assignment3 due date Dear all, Some of your friends are requesting to extend the due date for the assignment 3 until Wednesday. Accordingly, the new extended submission deadline is 17-04-2019,	Sun 4/14/2019 8:04 P...	39 KB	
	Charles Dzokoto	LAST CALL: BEYOND ASHESI'S 4s Hello, everyone, Good day.	Sun 4/14/2019 7:37 P...	198 KB	
	Richard Afun	**Popcorn at Hakuna Matata** Hello guys, Auntiee &Co. will be selling popcorn this evening!!!	**Sun 4/14/2019 6:31 ...**	**55 KB**	
	Maame-Akua Bess...	Marketing Survey Hey Guys, Kindly help a Marketing Group fill this questionnaire about Milky Mountain. Google Forms Survey	Sun 4/14/2019 5:38 P...	77 KB	

Figure 5.7 Classified Other Emails

Figure 5.7 shows successfully classified Other emails

5.4.2 Operations Emails


OPERATIONS - kwame.boahene@aucampus.onmicrosoft.com - Outlook
lder View Developer Help Tell me what you want to do
Reply Reply All Forward Meeting More
Respond
OTHER Team Email Reply & Delete To Manager Done Create New
Quick Steps
Move Rules OneNote
Move
Assign Policy Unread/ Read Categorize Follow Up
Tags
New Group Browse Groups
Groups
Search People Address Book Filter Email
Find
Read Aloud
Speech
Search OPERATIONS
Current Folder
OPERATIONS
By Date

FROM	SUBJECT	RECEIVED	SIZE	CATEGORIES	MENTION
Date: Today					
Joseph Peter ...	Insect infestation Hello Support, Recently there have been too many insects in the quads. Yoofi Brown-Pobee Ashesi University College '19 Management Information Systems Major	Sun 4/21/2019 3:...	43 KB		
Joseph Peter ...	Water in I block Hello Support, There is no water in the I block. It was not been running for two days Yoofi Brown-Pobee Ashesi University College '19 Management Information Systems	Sun 4/21/2019 3:...	39 KB		
Date: Last Month					
Prince Quansah	Issue The lights are not working. Prince Paa Kwesi Quansah Ashesi University || Business Administration	Thu 3/21/2019 2:...	36 KB		

Reply
PQ
To Kwame Bo
We converte
The lights a
Prince Paa I
Ashesi Univ
Mobile: +23
Email: princ

Figure 5.8 Classified Operations Emails

Figure 5.8 shows successfully classified Operations emails.

5.5 Acceptance Testing

To ensure the intended users approved of the system acceptance testing was done to ensure that requirements have been met. Below is feedback provided by a support personnel on the system:

1. When the VBA script triggers the shell script it opens a command prompt window briefly the support personnel wanted this window to run in background and not appear on the screen. Hence, this necessary revisions to the script was done to hide the pop-up window.

2. Support personnel also suggested that newly classified emails were extracted and added to the training data to improve the system's accuracy.

Chapter 6: Conclusion and Recommendations

6.1 Overview

This chapter presents a summary on the entire project. It also details out the limitations and challenges faced through the project as well as present future work that can be undertaken with this project as a basis.

6.2 Summary

This project explored the application of machine learning particularly classification algorithms to build an automated email classification system for the Ashesi Support Center. Support Vector Machine and Naïve Bayes were used to build a model that predicts whether an email is either IT or Operations related or unrelated to the two. The models achieved accuracies of 91% and 85% with support vector machine scoring the highest during training which was similar to results in [4,5,6] done using these algorithms. Further testing was done by simulating the support's centers email environment to see if the classifier would categorize user emails properly which it did.

With this system, users need not worry about manually sifting through their inbox to find important emails which was the main pain point identified during requirements gathering.

6.3 Limitations and Challenges

A major challenge faced during the project was converting of the data which was presented as a ".pst" file into text. Microsoft outlook stores backup as ".pst" therefore, there was a need to convert this into a text file since the classification and preprocessing components worked only with text. However, there was no free available tools to make this conversion and the duration of the project did not allow for building a system to convert from .pst to text format. Hence, a software tool was purchased to perform the conversion.

6.4 Future Work

The dataset as well as the classification system pave way for interesting and useful projects to improve efficiency at the Ashesi Support Center and the Ashesi community. Below are some additional features that can be implemented to make improve upon the email classification system:

1. Using the dataset, further classification can be done in the IT and Operations classes to give insights about the specific issues that are recurrent amongst complaints and problems the support center. Furthermore, classification can be done according to the roles played by the Support personnel. This would make tasks easier to identify and problems quicker to solve.
2. Automatically responding to emails. The current system is only able to classify emails however, after classification, the system can be improved to automatically send out a response to the sender. For example, if the user’s problem is classified as IT, an email containing instructions on how to deal with common IT problems senders face can be sent. A similar approach can also be done for Operations emails as well.

References

[1] Ani Nenkova and Amit Bagga. 2003. Email classification for contact centers. In Proceedings of the 2003 ACM symposium on Applied computing (SAC '03). ACM, New York, NY, USA, 789-792. DOI: https://doi.org/10.1145/952532.952689

[2] Taiwo Ayodele, Rinat Khusainov, and David Ndzi. 2007. Email classification and summarization: A machine learning approach. In *2007 IET Conference on Wireless, Mobile and Sensor Networks (CCWMSN07)*, 805–808. DOI: https://doi.org/10.1049/cp:20070271

[3] Mihajlo Grbovic, Guy Halawi, Zohar Karnin, and Yoelle Maarek. 2014. How Many Folders Do You Really Need?: Classifying Email into a Handful of Categories. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management* (CIKM '14), 869–878. DOI: https://doi.org/10.1145/2661829.2662018

[4] Xiao-Lin Wang and Cloete, "Learning to classify email: a survey," *2005 International Conference on Machine Learning and Cybernetics*, Guangzhou, China, 2005, pp. 5716-5719 Vol. 9. DOI: 10.1109/ICMLC.2005.1527956

[5] Shrawan Kumar Trivedi and Shubhamoy Dey. 2013. Effect of feature selection methods on machine learning classifiers for detecting email spams. In *Proceedings of the 2013 Research in Adaptive and Convergent Systems* (RACS '13). ACM, New York, NY, USA, 35-40. DOI: https://doi.org/10.1145/2513228.2513313

[6] R. Deepa Lakshmi and N. Radha. 2010. Spam classification using supervised learning techniques. In Proceedings of the 1st Amrita ACM-W Celebration on Women in Computing in India (A2CWiC '10). ACM, New York, NY, USA, Article 66, 4 pages. DOI: http://dx.doi.org/10.1145/1858378.1858444

[7] Ruikai Liu and McKie Jorj. 2019. *PyMuPDF: Python bindings for the PDF rendering library MuPDF*. (March 2019). Retrieved from https://github.com/pymupdf/PyMuPDF

[8] Edward Loper and Steven Bird. 2002. NLTK: the Natural Language Toolkit. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics - Volume 1* (ETMTNLP '02), Vol. 1. Association for Computational Linguistics, Stroudsburg, PA, USA, 63-70. DOI: https://doi.org/10.3115/1118108.1118117

[9] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P. Trevino, Jiliang Tang, and Huan Liu. 2017. Feature Selection: A Data Perspective. ACM Comput. Surv. 50, 6, Article 94 (December 2017), 45 pages. DOI: https://doi.org/10.1145/3136625

[10] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot,

and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. J. Mach. Learn. Res. 12 (November 2011), 2825-2830.

[11] *Pandas: a Foundational Python Library for Data Analysis and Statistics:* 2011. https://www.scribd.com/document/71048089/pandas-a-Foundational-Python-Library-for-Data-Analysis-and-Statistics. Accessed: 2019-04-22.

[13] Microsoft. 2019. Library reference VBA. Retrieved from https://docs.microsoft.com/en-us/office/vba/api/overview/library-reference

[14] Ashesi Support Center. 2019. Berekuso, ER, Ghana.

Appendices

Appendix A – Interview Guidelines

Name of interviewer:

……………………………………………………………………………

Date of interview:

………...………………………………………………………………

Place of interview:

…………………………………………………………………………….

Questions:

1. What role does email play in the problem-solving journey at the support center?
2. How many emails does the support center receive daily?
3. Does the support center classify emails?
4. Does the support center use any tool to classify emails automatically?