Email text analysis web application for emotion recognition using sentiment analysis

by

AHMAD KHAIRI BIN ROSMAN
17004391

Dissertation Report in Partial Fulfilment of The requirements for the
Bachelor of Information Technology (Hons)

SEPTEMBER 2021

Universiti Teknologi PETRONAS
32610 Seri Iskandar
Perak Darul Redzuan

CERTIFICATION OF APPROVAL

Email text analysis web application for emotion recognition using sentiment analysis

by

AHMAD KHAIRI BIN ROSMAN

17004391

A project dissertation submitted to the

Information Technology

Universiti Teknologi PETRONAS

In partial fulfilment of the requirement for the

BACHELOR OF INFORMATION TECHNOLOGY (HONS)

Approved by,

Dr Shuib bin Basri
Senior Lecturer
Department of Computer and Information Sciences
Universiti Teknologi PETRONAS
32610 Seri Iskandar,
Perak, Malaysia

DR. SHUIB BASRI

UNIVERSITI TEKNOLOGI PETRONAS

SERI ISKANDAR, PERAK SEPTEMBER

2021

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

Khairi

AHMAD KHAIRI BIN ROSMAN

ACKNOWLEDGMENT

First and foremost, I would like express my extremely grateful feeling to Allah, the Most Merciful, the Most Gracious, for providing me with the courage and compassion necessary to successfully accomplish my final year project.

Secondly, I would like to take this opportunity to express my sincere appreciation to Dr Shuib Basri. I'd want to thank him for his encouragement and assistance during this semester; I much appreciate it. Furthermore, I'd want to thank him for his overseeing technique, which simplified my work and kept me motivated during this period.

Furthermore, my heartfelt gratitude is offered to my friend Farah Batrisyia, for her support to help me during this project. I would to also say thanks to my other friends for their invaluable assistance and suggestions during the system's development. Last but not least, I want to express my heartfelt appreciation to my parents for their encouragement and support during the creation of this project. This project would not have been a success without their help, guidance, and encouragement.

ABSTRACT

While sentiment analysis for social media sites and web data has been a developing topic of research in text mining, research on email sentiment analysis has not been nearly as thorough, despite its widespread use in communication in our dayto-day tasks. The purpose of this study was to do sentiment analysis on email text in order to determine the emotional intentions represented in emails. This study presents a framework for email sentiment analysis based on a document-term matrix, and sentiment classification is carried out using the tidy text package using a lexicon-based approach. The results indicate which terms contribute to each sentiment in email data. The proposed framework aids in understanding the emotion represented by individuals in emails, which enables firms to make more informed decisions in order to meet customer expectations.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

NLP	Natural Language Processing
CRFs	Conditional Random Fields
AI	Artificial Intelligence
SVM	Support Vector Machine
VADER	Valence Aware Dictionary for Sentiment Reasoning
NLTK	Natural Language Toolkit
HTML	Hypertext Markup Language
DOM	Document Object Model

CHAPTER 1 INTRODUCTION

1.1 Project Background

Natural Language Processing (NLP) is a computer-assisted method to text analysis that is built on a foundation of concepts and technology. It is a subcategory of artificial intelligence that enables computers to comprehend, interpret, and manipulate human language. This project focuses on Sentiment Analysis, a type of natural language processing. Sentiment analysis is especially advantageous in circumstances where individuals share their opinions and feedback, such as consumer surveys, reviews, and social media comments. Sentiment analysis's simplest output is a threepoint scale with values of positive, negative, and neutral. In more complicated scenarios, the output can be a numeric score that can be categorised in any number of ways.

To narrow the scope of the study even further, it used emotion recognition for text analysis. Emotion detection is a subfield of sentiment analysis concerned with the extraction and analysis of human emotions. While much research has been conducted on speech and facial emotion recognition, text-based emotion identification systems continue to attract the attention of researchers (Sebe et al., 2005). As a result, some effort is required to achieve the best result. To improve the accuracy of emotion recognition, this project will incorporate grammar and spelling checks. This is to verify that the sentences always have the proper syntax and spelling, which will result in a good fit during the analysis.

1.2 Problem Statement

Email is widely used in the professional sector as a medium of communication between work peers. Emails appear to be used more cautiously in the business sector, as people will likely judge you by the way you write. Academic contacts, on the other hand, are more likely to be lively and carefree. Writing styles are important when writing emails as it can project emotion based on how it is written. Some people struggle in writing professional emails, as it need practice to learn the correct way of writing. This is important to be solve as these principles are intended to demonstrate professionalism and mutual respect between those exchanging emails. Sentiment analysis can be used to help prevent these types of email writing errors.

Sentiment analysis is the technique of identifying the polarity of a given text at the document or sentence level. It will decide whether the text's conveyed emotion is good, negative, or neutral. Sentiment analysis is critical because it allows businesses to quickly understand their customers' overall opinions. It is capable of processing massive amounts of data in an efficient and cost-effective manner. The problem lies on whether this data that were collected are accurate or not.

I believe that the primary issue is still the proper interpretation of the context in which certain words are used. It remains challenging for the vast majority of tools to accurately determine what constitutes a negative, neutral, or positive statement. Without a good accuracy on the sentiment analysis, the amount of data processed will be pointless. Therefore, further research needs to be conducted to get a better accuracy when detecting emotion through this method. I believe that by correcting the syntax of the text can give a better result with better accuracy. Grammar of a text can reveal emotion of what the writer wanted to tell the reader. How words are place in a sentence also matters when you are trying to convey a specific tone or emotion.

1.3 Project Objective

The objective of this project is to solve problems that have been mentioned in the previous section. The objectives are listed as the following:

i. To create a web application that will analyses email text to detect tone with sentiment analysis; and
ii. To study whether grammar syntax and spelling can increase the accuracy of sentiment analysis

1.4 Project Scope

The scope of study is listed as the following:

i. Sentiment analysis on an email; ii. Improving sentiment word identification algorithm; and iii. Developing fully automatic analysis web application.

CHAPTER 2 LITERATURE REVIEW

2.1 Email

Electronic Mail, or E-mail, is a form of communication used by a large number of people worldwide. It has been used at every level of society, from elementary school to universities, from business to consumer, and it continues to be used daily despite the introduction of more user-friendly communication services. Email has evolved from a vehicle for official documents to single-sentence greetings, and its multirecipient and attachment capabilities make it a great vehicle for informative, solicitous, and directive messages such as documents, publications, and press releases (Lan, L., 2000). Email usage in the office has established an etiquette that must be followed. Email etiquette is a set of rules that regulate how emails are written and responded to. While these standards of behaviour can be adapted to the audience and purpose of an email, they are intended to foster professionalism and mutual respect among email correspondents. Identifying the optimal sentence forms, word usage, and lexical content structure for affect detection, specifically frustration, is a difficult task in the case of text content (Munezero et al., 2014).

2.2 Natural Language Processing

Natural language processing (NLP) is a subfield of machine learning that is concerned with the processing of human language. It is the study of how computers can analyse and extract data from human language. Data gathered from conversations, emails, and even tweets will enable computers to comprehend natural language in the same way that humans do. The goal of NLP researchers is to understand how humans perceive and utilise language in order to develop tools and approaches that will aid computers in understanding and manipulating natural languages in order to perform tasks (Chowdhury, 2003). Natural language employs a variety of techniques, the most prevalent of which is sentiment analysis.

2.3 Sentiment Analysis

Sentiment analysis, alternatively referred to as Emotion Artificial Intelligence (AI), is a technology for extracting emotion or sentiment from written text. It may categorise documents, emails, reviews, and even social media posts as different emotion negative, neutral or positive. This method referred to as determining the polarity of text. It is suggested that subjective test data be used rather than objective test data. The term "objective test data" refers to statements that lack emotion or feeling. In other words, it is referred to be a neutral statement, and it will be ineffective in comparison to subjective test data. Subjective test data, on the other hand, are sentences that contain emotion-evoking phrases that can be detected using this method.

The amount of research including collecting opinion has also grown, which enable the public to access more user-generated content (Pang, Lee, et al., 2008). Consequently, several researches have been conducted, resulting in the publication of an astounding number of publications during the last two decades. With the growth of technology in parallel computing and machine learning, techniques for handling sentiment analysis problems have advanced dramatically. For the last decade, a review paper on sentiment analysis is produced, evaluating the suggested methodologies and advancements during that time period. Publications in various part were cited in such review papers and are organised in accordance with the granularities and task described in the Figure 2.1.

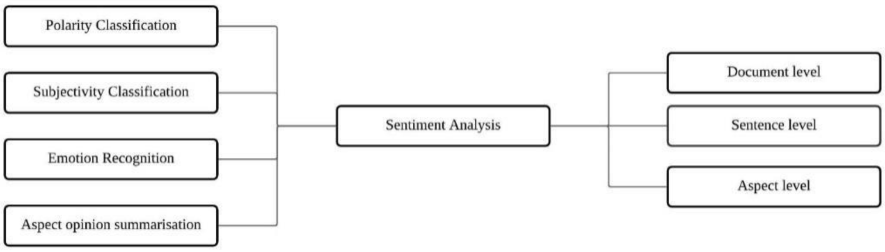


Figure 2.1: Sentiment level

2.3.1 Task in Sentiment Analysis

As mentioned on figure 2.1, the process of conducting sentiment analysis can be divided into 4 task which are as below:

- Classification of Polarity;
- Classification of Subjectivity;
- Recognition of emotion; and
- Aspect opinion summarisation.

2.3.1.1 Classification of Polarity

Sentiment analysis is concerned with polarity classification, the process of categorising texts as either positive, negative, or neutral on multiple levels such as document, phrase, and aspect (Fersini, et al., 2014). Classification of polarity is considered the key task of sentiment analysis. This task is an important task as it is what makes sentiment analysis gain interest from the machine learning community.

Polarity classification can be categorised into 2 types of method which are lexicon-centred method and machine learning method. Lexicon-centred method are used more often in the early studies of sentiment analysis. During the early studies, lexicon-centered key focus was to determine the sentiment preference of phrases. Recent research has lean to use established sentiment lexicons as characteristics rather than using sole predictor of a target's emotion polarity (Kundi et al., 2014).

The second method in polarity classification is machine learning method. Different strategies based on learning are being applied to overcome the crisis. Initially, the objective was deduced emotions from writings; however, the objective has changed to categorise the texts. Beside lexicon-centred methods, machine learning methods attempt to distinguish emotions using a pre-trained classifier. These methods employ a variety of machine developing theories, including SVM and CRFs, to decide which emotion type the input text should belong to.

2.3.1.2 Classification of Subjectivity

According to Wiebe, classification of subjectivity is concerned with determining if a text data, such as a document or phrase, has meaning or not. In contrast to classification of objective, subjective is decide if a sentence's primary goal is to be factual or not.

Classification of subjectivity, as described by Liu et al. (2010), is the process of evaluating if a statement is objective or subjective. To put it differently, classification of subjectivity is similar to a task of binary polarity classification at the phrase level. For example, Maas et al. (2011) performed subjectivity detection in sentence-level on movie data using a probabilistic model with a supervised sentiment element derived using a logistic regression predictor.

Nakagawa et al. (2010) used a probabilistic model with CRFs and unknown variables to identify phrase-level and sentence-level subjectivity, as well as to detect polarity in data from various domains. This research discovered that subjectivity categorization is frequently viewed as a before or after filtering phase for subsequent sentiment analysis, instead of a stand-alone assignment.

2.3.1.3 Emotion Recognition

Emotion recognition is a subset of classification of sentiment polarity that detect more detailed emotional conditions. In real-world, the attitudes expressed in certain types of opinionated text data may not neatly fall into binary classifications. The technologies for emotion recognition are far from perfect. While they are capable of properly detecting emotions, they nevertheless encounter and create concerns and challenges. For instance, a system may see subtle feelings and displays as more scary than those that are truly alarming.

2.3.1.4 Aspect Opinion Summarisation

The objective of aspect opinion summarisation is to summarise feelings related with particular characteristics of a collection of opinionated textual material. Aspect is described as a feature or subject that is incorporated into a document (Liu et al., 2010). This method most frequently seen in comments with numerous facets, each with a distinct sentiment polarity. Taking this into account, a summary of opinions based on traits gives a better interpretation than a classification of sentiment polarity.

According numerous research on aspect opinion summarisation, several focus on the characteristic extraction component of the project in order to increase the coverage of phrases (Yin et al., 2017), while others concentrate on the aspect-level polarity classification component of the task in order to improve the accuracy of classification in aspect-associated sentiment (Ruder et al., 2016).

2.3.2 Granularities in sentiment analysis

Aside from method, another method of conducting sentiment analysis is using granularities. Most frequently used granularities are document, sentence, and aspect level. A survey of the writing indicates that different features and methodologies for sentiment analysis may be used varying on if the main of the sentiment is using document, sentence, or aspect.

2.3.2.1 Aspect Opinion Summarisation

At document-level, sentiment analysis examines any little or lengthy whole opinionated document and considers sentiment polarity to be enough to review it. To achieve acceptable result in sentiment analysis at document-level, studies indicate that distinct aspects in little and lengthy papers should be prioritised. For brief documents, such as a message with a word limit, the emphasis is mainly on identifying phrases that contain sentiments or opinions, based on the premise that the document examines a particular topic and its associated sentiments (Kundi et al., 2014)For lengthy or variable-length publications, such as lyrics, diaries, or essays, determining the document's overall sentiment polarity is more dependent on an examination of elements and the weighted sum of the feelings associated with these variables.

2.3.2.2 Sentence level analysis

Sentence level sentiment analysis identifies sentiments contained in sentences, which are typically indicated by punctuation such as question marks, exclamation marks and full stops. Due to the inadequacy of sentence-level sentiment analysis for giving summarised data, the method more frequently undertaken at the same time with different levels of analysis. Sentiment analysis at sentence level is particularly beneficial for two types of problems: noise filtering and polarity shifts. "They played a bad game, but they win the match," is an example of a polarity turn from negative to positive. To be more precise, the first issue is handled by classifying phrases according to their subjectivity in order to exclude actual ones that are viewed as disturbance because they contain no bearing on the general feeling of the complicated material. The second issue is reffered by utilising techniques such as CRF to capture the relational and syntactical patterns that exist between the stages of a phrase (Nakagawa et al., 2010).

2.3.2.3 Analysis at Aspect Level

As some studies claim, assuming that a piece of textual data only contains a single sentiment is relatively rudimentary for document-level sentiment classification. (Ruder et al., 2016). Aspect level sentiment analysis detects and classifies the aspects conveyed in a text data. This level of study has grown in popularity over the last 10 years as approaches for studying granularity in other level have steadily improved after years of progress. A task involving sentiment analysis in aspect level typically consists of two components which are aspect extraction via probabilistic models or regression analysis, and sentiment categorization via neural system models (Ruder et al., 2016). Given that most sentiment analysis in aspect level tasks involve both sentiments and aspects, the majority of current research has concentrated on the review area, as review information typically contains numerous aspects and a score for each criterion that may be used as a reference point for assessments (Ruder et al., 2016).

2.4 Sentiment analysis model

Prabowo and Thelwall (2009) stated in a study article that with the improvement of technology, sentiment analysis could assist businesses in estimating the approval of their product. Sentiment analysis can assist businesses in analysing hundreds of product to have a more complete grasp of their consumers. This approach enables businesses to visualise client perceptions of their products and services. Sentiment analysis data can be used to help businesses address their weaknesses. Sentiment analysis provides a lot of methods will be explained in the next as per below sub-chapter .

2.4.1 Linear Regression

Linear regression is a statistical technique for predicting a Y value from a set of X features. The data sets are analysed using machine learning to determine if there is a relationship. After then, the associations are plotted along the X and Y axes with a straight line connecting them to anticipate additional relationships.

2.4.2 Naïve Bayes Algorithm

Naive Bayes is a fairly straightforward family of probabilistic procedures that designates a probability to whether a given phrase or word should be classified as either positive or negative in sentiment analysis. Essentially, Naive Bayes compares words to one another. Thus, using trained machine learning models for word polarity, we can determine if a word is positive or negative. When lemmatization, stop word elimination, and TF-IDF are used, Naive Bayes becomes increasingly predictive.

2.4.3 Support Vector Machine

A supervised machine learning method called Support Vector Machine or SVM that is capable of solving classification and regression problems. Classification is the process of predicting a label or group, whereas regression is the process of forecasting a continuous data. SVM classifies data by determining the hyperplane that separates the classes plotted in n-dimensional space.

CHAPTER 3 PROJECT METHODOLOGY

3.1 Agile Methodology

Agile technique will be utilised to manage this project. This is because agile technique results in a higher-quality product. The tasks required for this project will be split down into smaller tasks to guarantee they can be completed in a shorter amount of time. Additionally, this strategy helps ensure consistent delivery dates. In comparison to the old method, waterfall methodology have lengthy project phases, making it hard for teams to predict a release date accurately. Agile cycles occur in time-limited sprints, with each sprint resulting in a working product. As a result, the product owner can rest assured that new features will be provided at the end of each sprint.

Apart from that, this project will utilise Python to develop the sentiment model. This is because it is the machine learning language. Python includes a large number of libraries and pre-existing source code. This will benefit me in developing this project because I will have this much knowledge at my disposal. To host the sentiment model, a web application utilising ReactJs will be developed. ReactJs is a modern web application framework that is well-suited for this project.

3.2 Testing Phase

During this phase, I will employ testing techniques that are appropriate for my web application. This phase employs a white box testing technique known as unit testing to guarantee that the system is free of faults.

3.3 Vader

Valence aware Dictionary for Sentiment Reasoning or VADER is a sentiment analysis model that can be used on text which is sensitive to polarity such as positive and negative and the strength of an emotion. It can be found in the NLTK package and can be used directly to unlabelled text data without the need for further processing. The sentiment analysis performed by VADER is centred on a vocabulary that associates lexical traits with emotion strength referred to as sentiment value. The emotion score of a text may be derived by summing the intensities of each word within it. Vader polarity score will provide 4 data which are:

1. Positive or ‘pos’;
2. Neutral or ‘neu’;
3. Negative or ‘neg’; and
4. Compound or ‘compound’.

VADER calculates a compound score indicating whether a statement is positive, neutral, or negative. The compound score adds the valence ratings of all words in the lexicon, adjusts as necessary, and then normalises to a value between -1 and +1. To determine the sentence according to the category, the range for each category are as below:

1. Positive: Compound score is more or equal to 0.05;
2. Neutral: Compound score is between -0.05 and 0.05; and
3. Negative: Compound score is less or equal to -0.05

3.4 Flask

Figure 3.1: Flask

Flask is a backend framework for web application development. A framework serves as a basis for constructing applications, since it includes several pre-built modules and libraries that simplify the development process for developers. It establishes a framework for developers to follow while developing a web application. For this project, Flask is used as a back end to process the sentiment analysis.

Flask backend will take the input email text from frontend and process it on the backend. The process that will be done is , text pre-processing, text grammar and spell check and sentiment analysis. After all the process are done, the data will be sent back to the frontend to be display to the user.

3.5 ReactJS

ReactJS is a declarative, scalable, and extensible JavaScript framework for creating reusable user interface components. It is a free, component-based frontend library that is exclusively in charge for the application's view layer. A ReactJS app is composed of numerous components, each of which is accountable for a tiny piece of reusable HTML code. Every ReactJS apps are composed of components. The components could be nested to create sophisticated applications using simple building pieces. ReactJS utilises a virtual DOM-based technique to populate the HTML DOM with data. The virtual DOM is fast because it just alters individual DOM items rather than refreshing the entire DOM each time.

For the project, React is used to visualised the web application. It is used to get input from the user and send it to the Flask backend for processing. Then React will fetch the processed data and display it to the user.

CHAPTER 4 RESULT AND DISCUSSION

4.1 Flow chart

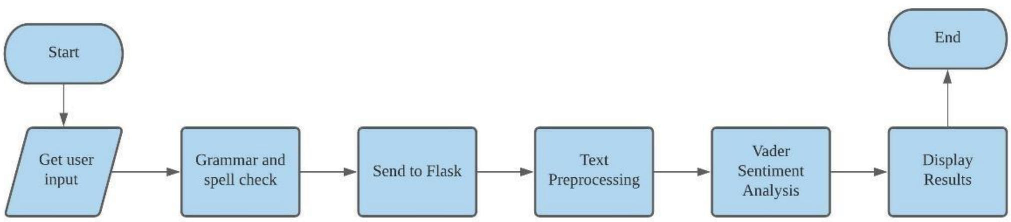


Figure 4.1: Flow Chart

4.2 Sequence Diagram

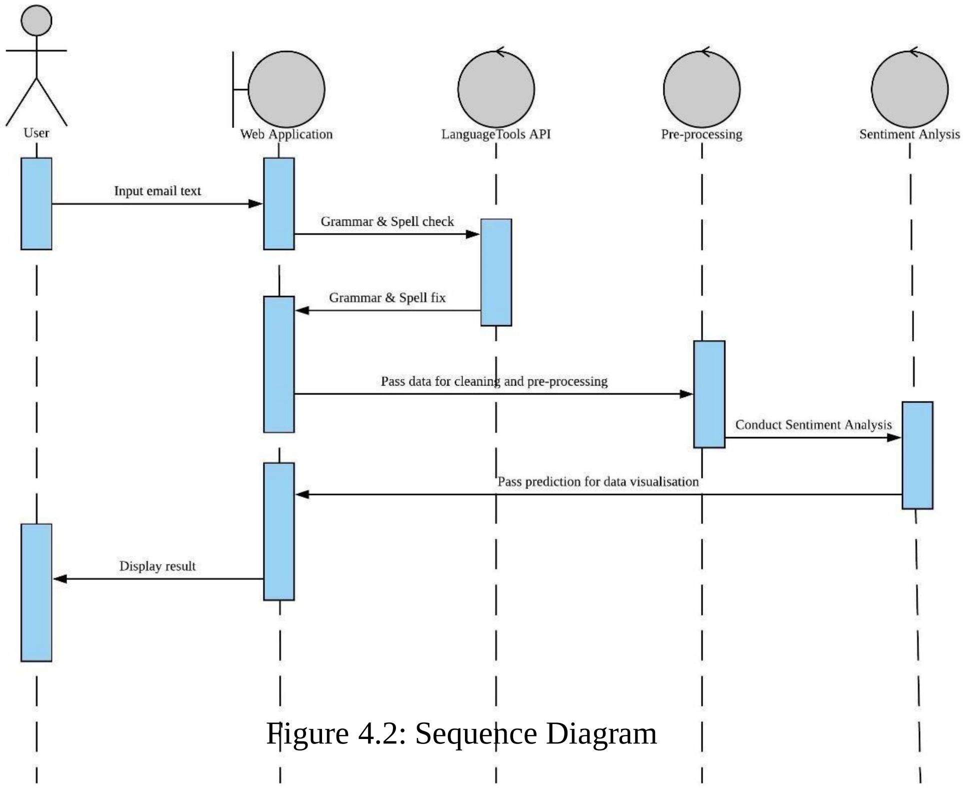


Figure 4.2: Sequence Diagram

4.3 Goal model diagram

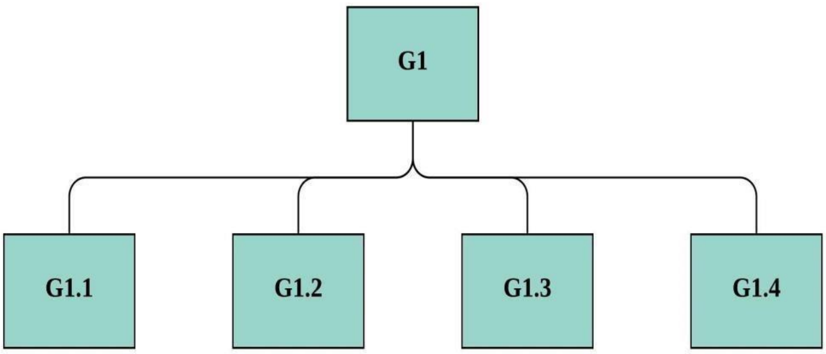


Figure 4.3: Goal Model Diagram

Goal ID	Goal

G1	Get User Input Use Case
G1.1	Grammar and spell check Use Case
G1.2	Text Pre-processing Use Case
G1.3	Sentiment Analysis Use Case
G1.4	Result Use Case

Table 4.1: Goal Model Table

4.4 Use case diagram

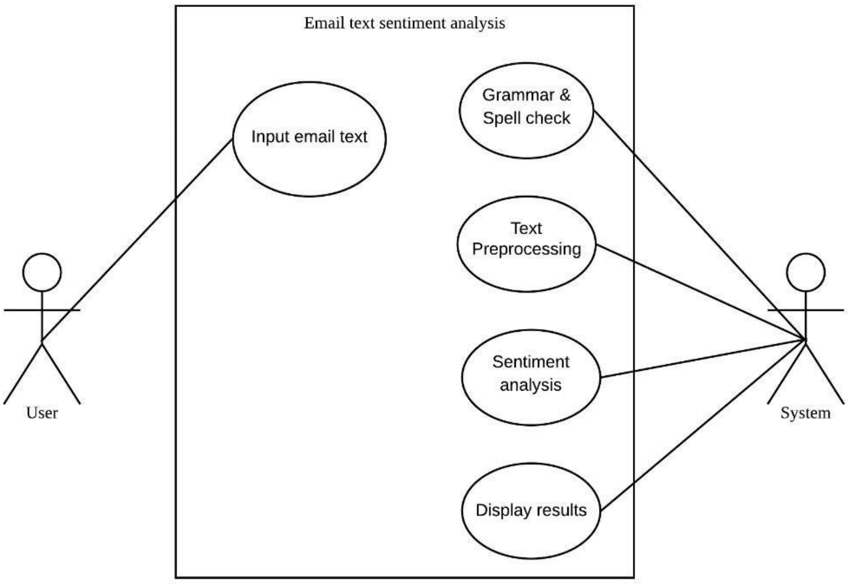


Figure 4.4: Use Case Diagram

4.5 Data flow diagram

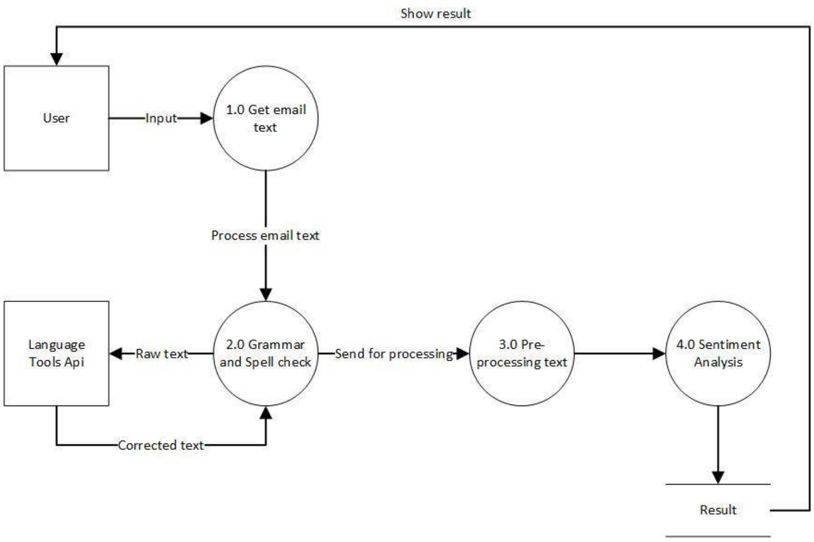


Figure 4.5: Data Flow Diagram

The data flow diagram describes the process of data flow in the sentiment analysis. The diagram starts when the user interacts with the webpage. The user will pass the email text to the textbox. The email text will be sent to the LanguageTools API for grammar and spell check. After the process is done the corrected text will be pass to the web application for pre-processing. After pre-processing, the clean text go pass through Vader. Vader will return the polarity score where it can determine whether the email text is positive, negative or neutral.

4.6 Project Interface

The email analysis web application's created system interface, features, and functionality were produced based on data collection analysis by exhibiting the application's concept. The following figures depict the produced application's UI screenshots.

4.6.1 Main Page

The figure depicts the page when the user first arrives. The page displays the project's title and a button that allows the user to interact with the project.

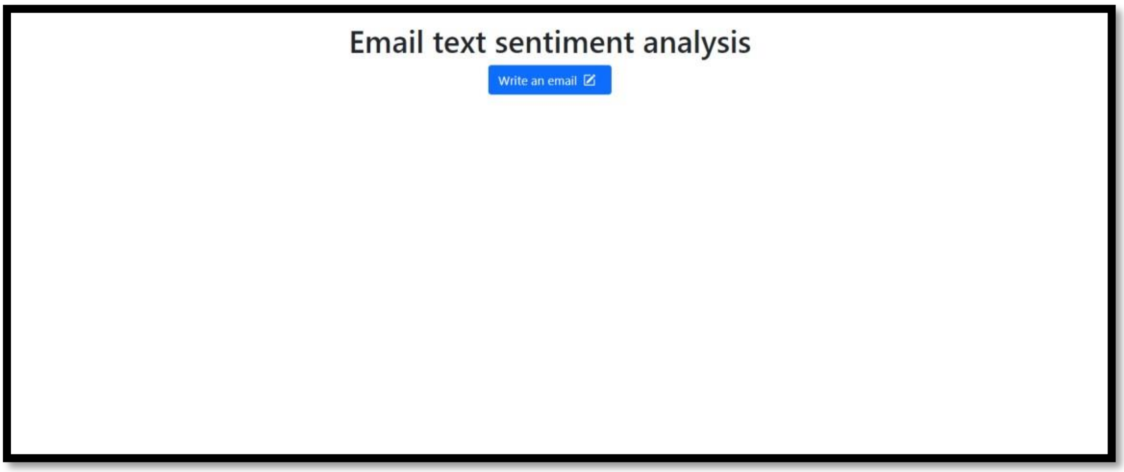


Figure 4.6: Main Page

4.6.2 “Write an email” button

In the figure below, it shows the result when the “Write an email” button was pushed. It will reveal an input box where user can put the email title and the content of the email. The sentiment analysis will be done on the text that user put in the “Body” textbox. When the user is ready to analyses their email sentiment, they can push the “Start Sentiment” button.

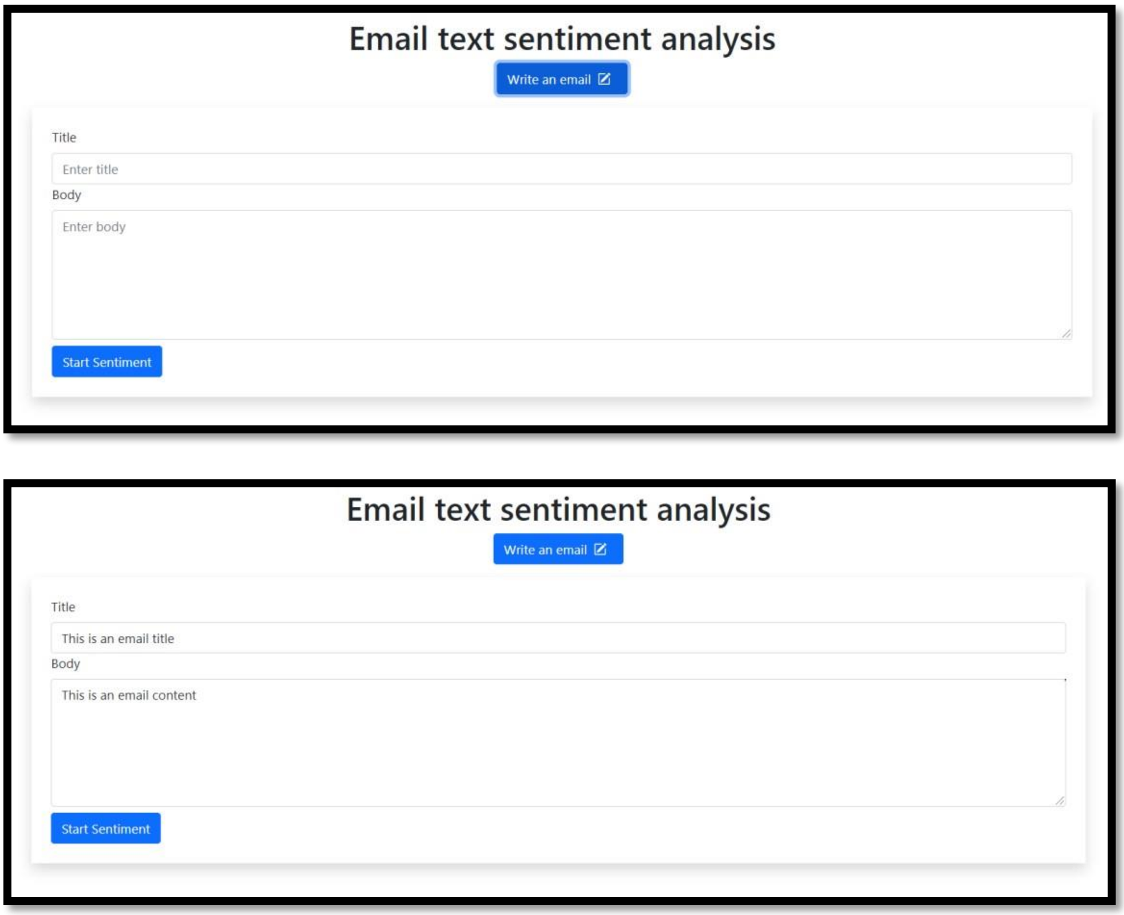


Figure 4.7: "Write an Email" Button

4.7 Sentiment result

The figure below shows the result when the “Start sentiment” button was pushed. The button will show the result of the sentiment analysis. In the figure, when the result of sentiment is shows, it will display 3 things which are:

1. Email text

In this box, it will display the original text that was put in the email body text box.

2. Sentiment Result

   In this box, it will display whether the text that was process was a positive, negative or neutral email.

3. Corrected Text

   In this box, it will display the text that was process through the grammar and spelling check. It there was no error, it will display the same text as the input email text.

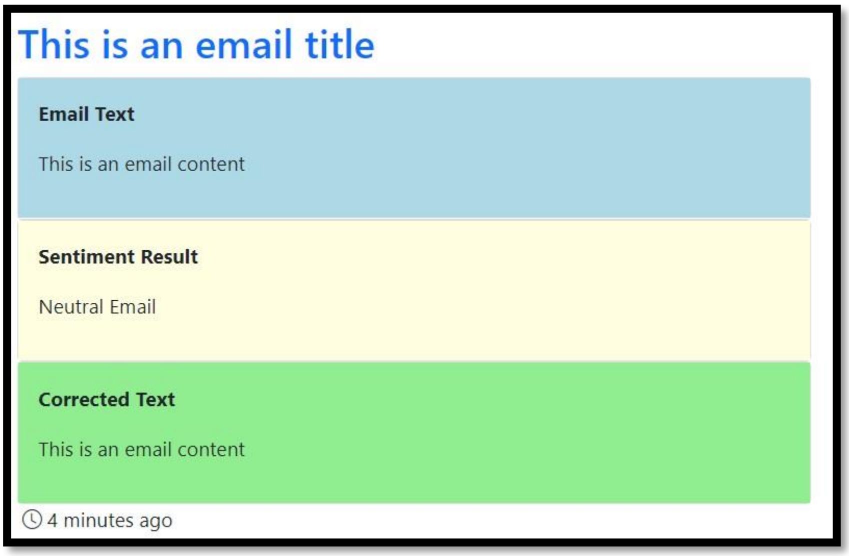


Figure 4.8: Sentiment Result

4.8 Sentiment Analysis Result

For this project, the email text sentiment analysis will show 3 different results as per below sub-chapter:

4.8.1 Positive email

Below is a result for a positive email. For this sentiment analysis the text that was provide for the session is:

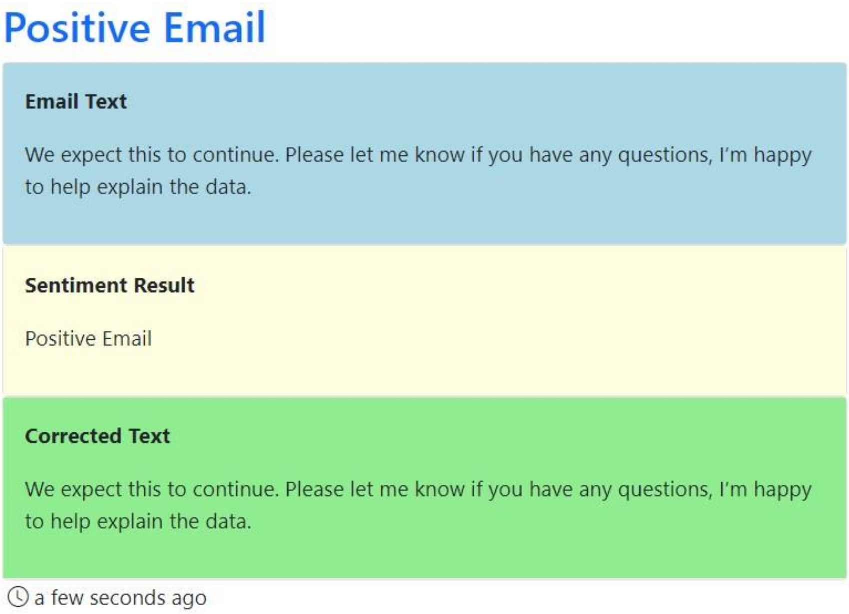


Figure 4.9: Positive Email

Email Text	We expect this to continue. Please let me know if you have any questions, I'm happy to help explain the data.
Positive percentage	**35%**
Neutral percentage	**64%**
Negative percentage	**0%**

Table 4.2: Positive Email

4.8.2 Neutral email

Below is the result for the Neutral email. For this sentiment analysis, the data for the session is as below:

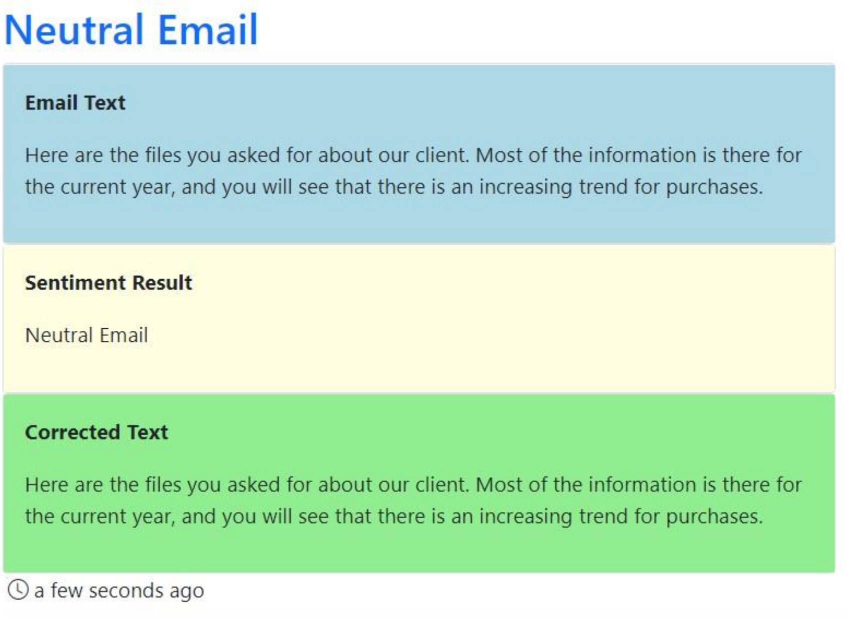


Figure 4.10: Neutral Email

Email Text	Here are the files you asked for about our client. Most of the information is there for the current year, and you will see that there is an increasing trend for purchases.
Positive percentage	**0%**
Neutral percentage	**100%**
Negative percentage	**0%**

Table 4.3: Neutral Email

4.8.3 Negative email

Below is the result for the Negative email. For this sentiment analysis, the data for the session is as below:

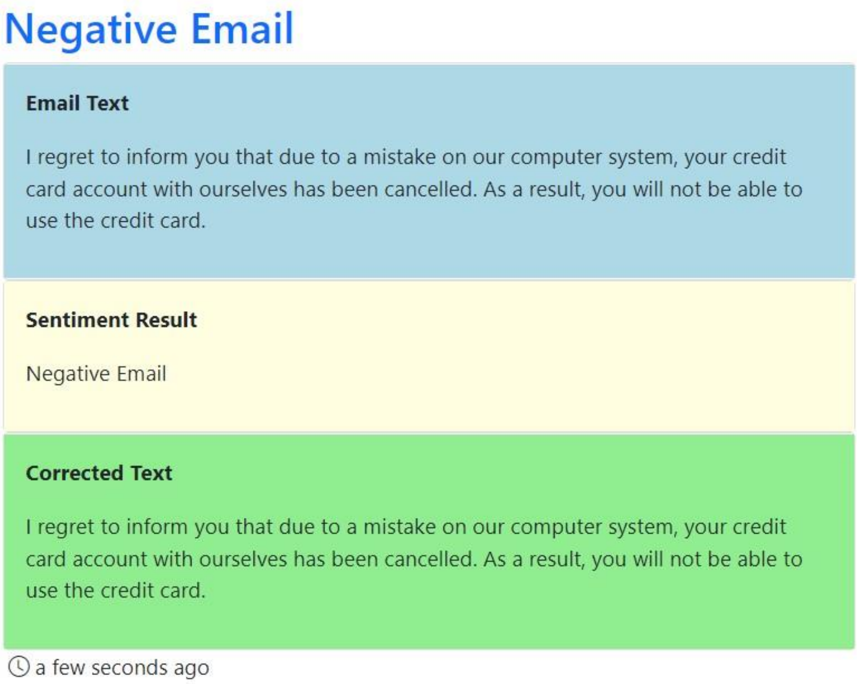


Figure 4.11:Negative Email

Email Text	**Here are the files you asked for about our client. Most of the information is there for the current year, and you will see that there is an increasing trend for purchases.**
Positive percentage	**7%**
Neutral percentage	**71%**
Negative percentage	**22%**

Table 4.4: Negative Email

4.9 Testing phase

For testing phase, I use a white box testing method called unit testing. To ensure that this project was delivered with quality, few testing has been done to ensure it is faulty free. For the testing phase, the unit testing was done on 3 different part of the project. The feature that was tested was:

1. Text cleaning and pre-processing;
2. Text spelling and grammar; and

3. Sentiment analysis.

4.9.1 Text cleaning and pre-processing

For the first feature, text cleaning and pre-processing, the unit testing result is as follow:

```
class CleanTextTest(unittest.TestCase):
    def test_cleaning(self):
        self.assertEqual(textcleaning.clean('I hope to see you all welcome him into the office and provide him with your help and feedback wherever necessary.'), "hope to see you all welcome him into the office and provide him with your help and feedback wherever necessary")

if __name__ == '__main__':
    unittest.main()
```

Figure 4.12: Test Case 1 Text Cleaning and Pre-processing

The input for **test case 1** in the text cleaning and pre-processing function was "I hope to see you all welcome him into the office and provide him with your help and feedback wherever necessary" The unit test is intended to eliminate unnecessary words that add no significance. For this test, the word "I" is removed.

Input	I hope to see you all welcome him into the office and provide him with your help and feedback wherever necessary.
Expected Output	hope to see you all welcome him into the office and provide him with your help and feedback wherever necessary
Actual Output	hope to see you all welcome him into the office and provide him with your help and feedback wherever necessary
Result	Pass

Table 4.5: Test case 1 Text Cleaning and Pre-processing

```
class CleanTextTest(unittest.TestCase):
    def test_cleaning(self):
        self.assertEqual(textcleaning.clean('Dear Mr Ali, please contact the company on the email stated below : company@gmail.com'), "Dear Mr Ali please contact the company on the email stated below company gmail com")

if __name__ == '__main__':
    unittest.main()
```

Figure 4.13: Test Case 2 Text Cleaning and Pre-processing

For **test case 2** in text cleaning and pre-processing feature, the input that was use were “Dear Mr Ali, please contact the company on the email stated below : company@gmail.com”. For this test, it removes the special character that cannot be analyses. The word that was remove was “@” , “.”, “:” and “,”.

Input	Dear Mr Ali, please contact the company on the email stated below : company@gmail.com
Expected Output	Dear Mr Ali please contact the company on the email stated below company gmail com
Actual Output	Dear Mr Ali please contact the company on the email stated below company gmail com
Result	Pass

Table 4.6: Test Case 2 Text Cleaning and Pre-processing

4.9.2 Text spelling and grammar

For the feature 2, Text Spelling and Grammar, the unit testing result is as follow:

```
class SpellGrammar(unittest.TestCase):
    def testSpell_Grammar(self):
        self.assertEqual(languageTools.spell_grammar('Dear Mr Adib, attached in this emaal is the file
for the project. Sorry for the late reply as i am currently on a pusy schedule.'),'Dear Mr Adib,
attached in this email is the file for the project. Sorry for the late reply as I am currently on a
busy schedule.')

if __name__ == '__main__':
    unittest.main()
```

Figure 4.14: Test Case 1 Text Spelling and Grammar

For **test case 1** in text spelling and grammar checking, the input that was given was “**Dear Mr Adib, attached in this emaal is the file for the project. Sorry for the late reply as i am currently on a pusy schedule.**” The test was used to correct misspell words and grammar. For this unit test, the word that was misspell were “emaal”, “i” and “pusy. The correct spelling of the word are “email” “I” and “busy”.

Input	Dear Mr Adib, attached in this emaal is the file for the project. Sorry for the late reply as i am currently on a pusy schedule.

Expected Output	Dear Mr Adib, attached in this email is the file for the project. Sorry for the late reply as I am currently on a busy schedule.
Actual Output	Dear Mr Adib, attached in this email is the file for the project. Sorry for the late reply as I am currently on a busy schedule.
Result	Pass

Table 4.7: Test Case 1 Text Spelling and Grammar

```
class SpellGrammar(unittest.TestCase):
    def testSpell_Grammar(self):
        self.assertEqual(languageTools.spell_grammar("Please note that the file in this email is
necesary for the project. This email is confidential.") ,'Please note that the file in this email is
necessary for the project. This email is confidential.')

if __name__ == '__main__':
    unittest.main()
```

Figure 4.15: Test Case 2 Text Spelling and Grammar

For **test case 2**, the input that was given was "**Please note that the file in this email is necesary for the project. This email is confidentaal.**" For this unit test, the word that was misspell were "**necesary**" and "**confidentaal**". The correct spelling of the word was "necessary" and "confidential".

Input	Please note that the file in this email is necesary for the project. This email is confidentaal.
Expected Output	Please note that the file in this email is necessary for the project. This email is confidential.
Actual Output	Please note that the file in this email is necessary for the project. This email is confidential.
Result	Pass

Table 4.8: Test Case 2 Text Spelling and Grammar

4.9.3 Sentiment analysis

For the feature 3, Sentiment Analysis result will give for data which is Positive ('pos'), Neutral ('neu'), Negative ('neg') and Compound ('compound'). The first 3 data was used to calculate the whole sentiment of the sentence which equal to the Compound. The sentiment of the email was decided as follow:

1. Text If Compound value is more or equal to 0.05, it is a Positive email; and
2. If Compound value is less or equal to -0.05, it is a Negative email.

```
class sentimentTesting(unittest.TestCase):
    def test_sentiment(self):
        self.assertEqual(sentimentAnalysis.textSentiment('We expect this to continue. Please let me
know if you have any questions, I'm happy to help explain the data.'), {'neg': 0.0, 'neu': 0.674,
'pos': 0.326, 'compound': 0.8271})

if __name__ == '__main__':
    unittest.main()
```

Figure 4.16: Test Case 1 Sentiment analysis

For **test case 1** in Sentiment analysis feature, the input that was given was “We expect this to continue. Please let me know if you have any questions, I’m happy to help explain the data.”. For this unit test, the expected result was to get a positive email. The result of the sentiment was : Negative : 0% , Neutral : 67% , Positive : 33% , Compound : 0.83. Therefore, the email is a positive email.

Input	We expect this to continue. Please let me know if you have any questions, I’m happy to help explain the data.
Expected Output	**Positive Email**
Actual Output	{'neg': 0.0, 'neu': 0.674, 'pos': 0.326, 'compound': 0.8271} **Positive Email**
Result	Pass

Table 4.9: Test Case 1 Sentiment analysis

```
class sentimentTesting(unittest.TestCase):
    def test_sentiment(self):
        self.assertEqual(sentimentAnalysis.textSentiment('Here are the files you asked for about our
client. Most of the information is there for the current year, and you will see that there is an
increasing trend for purchases.'), {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0})

if __name__ == '__main__':
    unittest.main()
```

Figure 4.17: Test Case 2 Sentiment Analysis

For **test case 2** in Sentiment analysis feature, the input that was given was "Here are the files you asked for about our client. Most of the information is there for the current year, and you will see that there is an increasing trend for purchases.". For this unit test, the expected result was to get a positive email. The result of the sentiment was : Negative : 0% , Neutral : 100% , Positive : 0% , Compound : 0.0. Therefore, the email is a neutral email.

Input	Here are the files you asked for about our client. Most of the information is there for the current year, and you will see that there is an increasing trend for purchases.
Expected Output	**Neutral Email**
Actual Output	{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0} **Neutral Email**
Result	Pass

Table 4.10: Test Case 2 Sentiment Analysis

```
class sentimentTesting(unittest.TestCase):
    def test_sentiment(self):
        self.assertEqual(sentimentAnalysis.textSentiment('I regret to inform you that due to a mistake
on our computer system, your credit card account with ourselves has been cancelled. As a result, you
will not be able to use the credit card.'), {'neg': 0.213, 'neu': 0.71, 'pos': 0.077, 'compound':
-0.5574})

if __name__ == '__main__':
    unittest.main()
```

Figure 4.18: Test Case 3 Sentiment Analysis

For **test case 3** in Sentiment analysis feature, the input that was given was "I regret to inform you that due to a mistake on our computer system, your credit card account with ourselves has been cancelled. As a result, you will not be able to use the credit card.". For this unit test, the expected result was to get a negative email. The result of the sentiment was : Negative : 22% , Neutral : 71% , Positive : 7% , Compound : -0.5574. Therefore, the email is a Negative email.

Input	I regret to inform you that due to a mistake on our computer system, your credit card account with ourselves has been cancelled. As a result, you will not be able to use the credit card.

Expected Output	**Negative Email**
Actual Output	{'neg': 0.213, 'neu': 0.71, 'pos': 0.077, 'compound': -0.5574} **Negative Email**
Result	Pass

Table 4.11: Test Case 3 Sentiment Analysis

CHAPTER 5 CONCLUSION

5.1 Project Overview

The usage of email will improve using sentiment analysis for emotion recognition. Email text can be analysed better by using Naïve Bayes Algorithm which can help professional worker to write better email when sending it to the higher ups. This feature will be host on a web application to ensure it can be access by all people.

5.2 Objective of Project

According to Chapter 1, this project has 2 objective that needed to be achieve. The objective is as follow:

i. To create a web application that will analyses email text to detect tone with sentiment analysis.
   By using ReactJs as a frontend and Flask as a backend, the project was developed to the enable user to further understand the tone of their email. As stated in the Chapter 3, we can see that the project was visualised and was able to achieve first objective of this project.

ii. To study whether grammar syntax and spelling can increase the accuracy of sentiment analysis
   Grammar syntax and spelling are important in the sentiment analysis. This is because it allow the model to read and analyse more text when it is used in the sentence correctly. Through my study and research in this project, I believe that grammar and syntax are important to achieve a higher accuracy in sentiment analysis. A misspell word may not give any sentiment but a correct word can decide whether the email contain positive, neutral or negative tone.

5.3 Recommendation

Sentiment analysis has developed into one of the most active fields of research during the previous decade. The usage of sentiment analysis has brought benefits to people from all sector. It is also an intriguing subject to investigate since it presents several novel research issues. Finally, in the age of big data technology, we now have a massive amount of opinionated data stored and freely available on the web in digital form. These factors have fuelled recent progress in this field. I believe that with more research and time, the accuracy of the sentiment analysis can be further increase. The data can be represented better in future by utilizing it feature to detect which word that bring out the tone in the email text. To conclude, an email web app which can help to detect emotion tone in the email text can help the professional sector to decide better before sending their email.

REFERENCES

Chowdhury, G. G. (2003). Natural language processing. Annual Review of Information Science and Technology, 37(1), 51–89. https://doi.org/https://doi.org/10.1002/aris.1440370103

Fersini, E., Messina, E., & Pozzi, F. (2014). Sentiment analysis: Bayesian ensemble learning. Decision Support Systems, 68, 26-38. doi:10.1016/j.dss.2014.10.004

Kundi F, Ahmad S, Khan A, Asghar (2014) Detection and Scoring of Internet Slangs for Sentiment Analysis Using SentiWordNet, Life Science Journal, 11(9):66–72.

Lan, L. (2000). Email: A challenge to Standard English? English Today, 16(4), 23–29. https://doi.org/10.1017/S0266078400000511

Liu, Bing et al. (2010). “Sentiment analysis and subjectivity.” In: Handbook of natural language processing 2.2010, pp. 627–666.

Maas, Andrew L et al. (2011). “Learning word vectors for sentiment analysis”. In: Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1. Association for Computational Linguistics, pp. 142–150.

Myriam D Munezero, Calkin Suero Montero, Erkki Sutinen, and John Pajunen. 2014. Are they different? affect, feeling, emotion, sentiment, and opinion detection in text. IEEE transactions on affective computing, 5(2):101–111.

Nakagawa, Tetsuji, Kentaro Inui, and Sadao Kurohashi (2010). “Dependency tree-based sentiment classification using CRFs with hidden variables”. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, pp. 786–794.

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. Foundations and Trends® in Information Retrieval, 2(1–2), 1-135. doi:10.1561/1500000011

Prabowo, Rudy & Thelwall, Mike. (2013). Sentiment analysis: A combined approach. Journal of Informetrics. 143-157. 10.1016/j.joi.2009.01.003.

Ruder, Sebastian, Parsa Ghaffari, and John G Breslin (2016). “A hierarchical model of reviews for aspect-based sentiment analysis”. In: arXiv preprint arXiv:1609.02745.

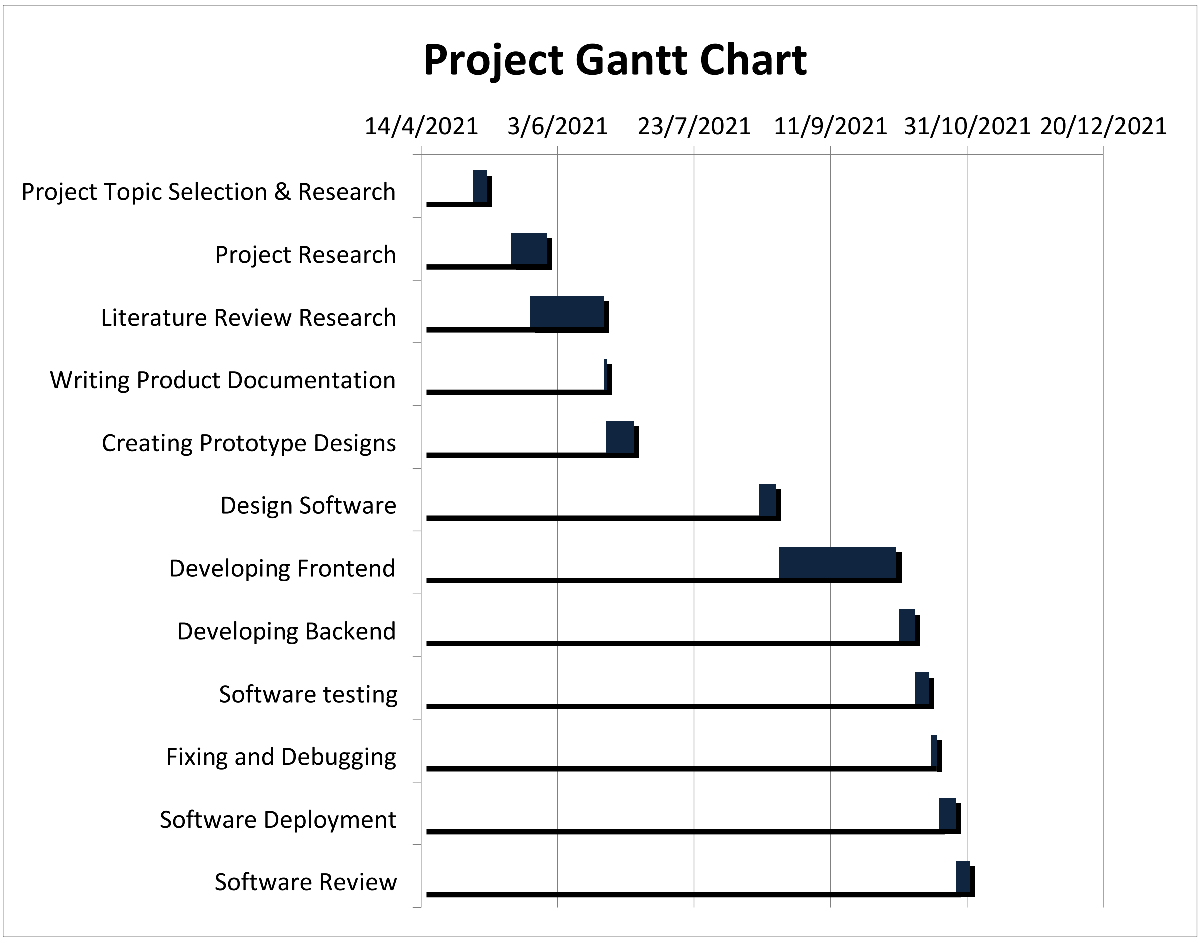
Sebe, N., Cohen, I., Gevers, T., & Huang, T. S. (2005). Multimodal approaches for emotion recognition: A survey. Proceedings of SPIE - The International Society for Optical Engineering, 5670, 56-67. [08]. https://doi.org/10.1117/12.600746

Wiebe, Janyce, Rebecca Bruce, and Thomas P O’Hara (1999). “Development and use of a gold-standard data set for subjectivity classifications”. In: Proceedings of the 37th annual meeting of the Association for Computational Linguistics, pp. 246–253.

Yin, Yichun, Yangqiu Song, and Ming Zhang (2017). “Document-level multi-aspect sentiment classification as machine comprehension”. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 2044–2054.

APPENDICES

Appendix A- 1: Gantt Chart



Appendix A- 2: Project Timeline

Task Name	Start	End	Duration (days)
Project Topic Selection & Research	3/5/2021	16/5/2021	5
Project Research	17/5/2021	30/5/2021	13
Literature Review Research	24/5/2021	20/6/2021	27
Writing Product Documentation	20/6/2021	21/6/2021	1
Creating Prototype Designs	21/6/2021	1/7/2021	10
Design software	16/8/2021	22/8/2021	6
Developing Frontend	23/8/2021	5/10/2021	43
Developing Backend	6/10/2021	12/10/2021	6
Software testing	12/10/2021	17/10/2021	5
Fixing and Debugging	18/10/2021	20/10/2021	2
Software deployment	21/10/2021	27/10/2021	6
Software Review	27/10/2021	1/11/2021	5