

GUI based Sentiment Analyzer .

A mini project report submitted by

Sai Ram.K - 20181CSE0621

Sai Tarun S.S - 20181CSE0622

Samhitha.V.Gowda - 20181CSE0629

*as part of lab-based course Programming in Python, CSE 317
of*

BACHELOR OF TECHNOLOGY

in

COMPUTER ENGINEERING

under the supervision of

Ms. Ganga V.C,Assistant Professor



DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

PRESIDENCY UNIVERSITY

Itgalpur Rajanakunte, Yelahanka, Bengaluru, Karnataka-560064



PRESIDENCY UNIVERSITY

(Established under the Presidency University Act, 2013 of the Karnataka Act 41 of 2013)

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that the project report entitled, “**GUI based Sentiment Analyzer**” is a bonafide record of Mini Project work done as part of CSE317 Programming in Python during the academic year 2020-2021 by

Sai Ram.K - 20181CSE0621
Sai Tarun S.S - 20181CSE0622
Samhitha.V.Gowda - 20181CSE0629

Submitted for the Viva Voce held on _____

Examiner

Contents

Abstract	4
Acknowledgement	4
1. Introduction	5
1.1 Domain Introduction	
1.2 Objectives	
1.3 Motivation	
1.4 Overview	
2. Design and Architecture	7
2.1 Blueprint of layout	
2.2 Architecture and geometry	
2.3 Diagrams	
3. Implementation	10
3.1 Highlight of modules used	
3.2 Flow of Code	
3.3 Code	
3.3 Tools used	
4. Conclusion and Future Scope	16
5. Output and Screenshots	17
References & Bibliography	21

Abstract

GUI based sentiment analyser is a tool that has been implemented to address the problem of Sentimental Analysis that has a vast application in today's world. The purpose of this analysis is to classify the sentences based on positive, negative or neutral. This classification helps in making clear the intentions of customers on the products or the posts that have been marketed to them in a corporate application. Domestically this tool can be used for twitter or any social media comment analysis. There are millions of active users in social media pages such as Twitter that has 250 million tweets per day. Due to this large amount of usage we hope to achieve a reflection of public sentiment by analyzing the sentiments expressed in the tweets. The main aim of this is to successfully analyse the sentences that are being given as an input to the GUI and also show the accuracy at which the built classifier is being able to handle the respective inputs.

Advantages of this tool :-

- Helps in knowing and understanding the customers.
- Helps in measuring the marketing campaign.
- Boosts customer service.
- Provides a data-based strategy that targets interested individuals.

Acknowledgement

We as a team would like to express our gratitude to our teacher Ms. Ganga V.C without whose efficient guidance this project wouldn't have shape. We also express our gratefulness to our HOD who gave us an opportunity to pursue this project. This project has made us do a lot of research and learn a lot of technical knowledge based on python. Our entire thanks also goes to the teachers of CSE department who have guided us so well and helped us in all their strength in order to complete this project.

Secondly, we would like to thank our seniors who have given us their guidance and also shared their experience and informed us about the do's and don'ts to be followed. We also would like to thank our classmates who were a constant source of positivity and motivation to help us in doing better each time.

And lastly, to everyone who have helped us indirectly and also to those who are interested in knowing more about this project we thank you very much.

Thank you.

Chapter – 1

INTRODUCTION

1.1 Domain Introduction

Let's start with the problem. After the IT boom in the year of 2010 many multi national companies began using social media as a place to market their products as it had a very wide spread reach globally. Millions of users and thousands of companies were involved in these processes. After few years it became evident there's a essential requirement to keep track of the comments that users are giving to see how well their product is being reached to the public. It was a very tedious process to keep track of the comments manually hence it gave rise to the need of sentimental analysis.

Sentiment Analysis is the process of mining the intent behind the sentence described by the user. This has a very wide application in text mining and pattern classification. In this project we have put forth a GUI that accepts the user's sentence or a paragraph and in return it classifies it into Positive, Negative or a Neutral sentence.

The features that can be used for modeling patterns and classification can be divided into two main groups: formal language based and informal blogging based. Language based features are those that deal with formal linguistics and include prior sentiment polarity of individual words and phrases, and parts of speech tagging of the sentence. Prior sentiment polarity means that some words and phrases have a natural innate tendency for expressing particular and specific sentiments in general. For example, the word "excellent" has a strong positive connotation while the word "evil" possesses a strong negative connotation. So whenever a word with positive connotation is used in a sentence, chances are that the entire sentence would be expressing a positive sentiment. Parts of Speech tagging, on the other hand, is a syntactical approach to the problem. It means to automatically identify which part of speech each individual word of a sentence belongs to: noun, pronoun, adverb, adjective, verb, interjection, etc. Patterns can be extracted from analyzing the frequency distribution of these parts of speech (ether individually or collectively with some other part of speech) in a particular class of labeled sentences.

1.2 Objectives

The objectives we intend to hit are listed below:

- To take the user-based sentence or paragraph and feed it to the analyser.
- To classify the sentence based on metrics and categorizing it into Positive, Negative or Neutral.
- To be able to exactly compute the accuracy score of the 3 categories we have classified the sentence into.
- To display the user the computed results as per the Intensity.
- Additional objectives include designing a user-friendly GUI interface along with a clear and exit button.

1.3 Motivation

We have chosen to work with this project because we strongly believe that it would provide us with a thorough understanding of the real-world applications in combination with a GUI. The real-world applications in most of the use-case scenarios appear to be challenging and this project is one of the challenging applications to be dealt with. As this would give us an idea of the modules that have been used with and also the practical challenges that have to be overcome as there is a use of a GUI based representation. Sentiment analysis of public is highly critical in macro-scale socioeconomic phenomena like predicting the stock market rate of a particular firm. This could be done by analyzing overall public sentiment towards that firm with respect to time and using economics tools for finding the correlation between public sentiment and the firm's stock market value. Firms can also estimate how well their product is responding in the market, which areas of the market is it having a favorable response and in which a negative response.

We thoroughly believe that by undertaking this project we would maximize our current capability of learning and also our ability to think and solve real world related problems and this would enhance our learning potential.

1.4 Overview

The short overview of this project is by following a heuristic approach. The procedure starts by taking a sentence or a paragraph as input from the user and classifying it into a positive, negative or a neutral sentence. Along with this we would also display the intensity score as a function of how strong has our model been able to classify the given input. In order to achieve this, we employ the use of a module called Valence Aware Dictionary for Sentiment Reasoning or Vader. In order to display the results in the form of a GUI based user friendly experience we employ the use of Tkinter module.

At the end of the procedure the GUI will display the intent, accuracy and the overall rating of the given sentence in terms of percentage and an Overall rating.

Chapter – 2

Design and Architecture

2.1 Blueprint of layout

In order to make the user friendly design of the sentiment analyser we decided to use the Tkinter module of python as it is beginner friendly and easy to implement along with a wide variety of widgets and functions. The layout we have chosen to maintain throughout the code is the dimensions of 250 x 400. This would allow us to display all the widgets, buttons and also the text area in an efficient way. The below image gives an idea of the basic window along with the dimensions that we have used in the making of the this tool.

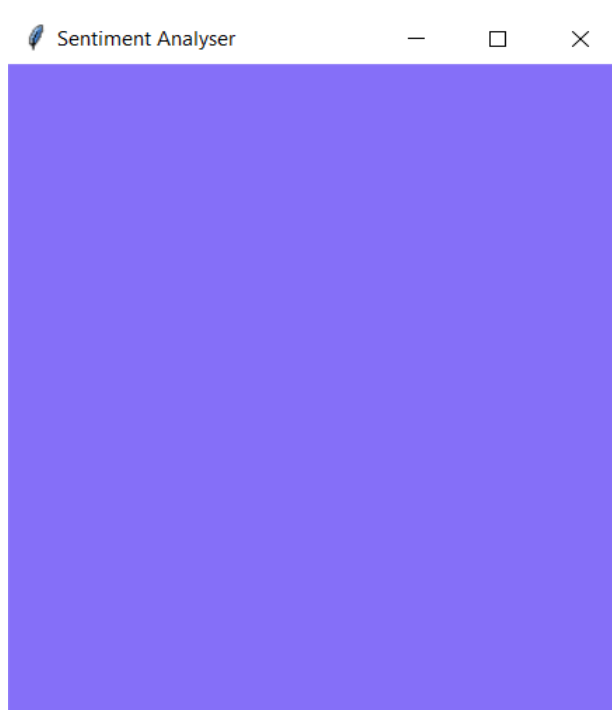


Fig 1.1 : The outline layout of the GUI

2.2 Architecture

Inclining towards the architecture of this model; the package named “vaderSentiment” is used, that contains a set or any sort of a dictionary that contains pre-defined lexical features which are labeled as positive or negative words accordingly. And moreover, the “tkinter” package is made use of, in order to implement the GUI toolkit which serves the purpose of displaying and exploring the various options of windowed display elements such as buttons, menus, display areas and many as such.

Finally, throwing light on the most important component, in essence, the **BUTTONS** of the window representing the sentiment analyzer. The “Button()” function can be made use of, that contains several parameters defining its characteristics namely, i) the ‘gui window’ that

was created is passed as 1st parameter ii) 'text' , for the required text. iii)'fg' that specifies the fore-ground iv) 'bg' gives the background colour. And another parameter i.e., command can also be written along. And also "Label()" which is pretty self-explanatory too contains the same parameters as that of the "Button()" function excluding "command". And then, grid method is used for placing; and its used with every field including the overall field and the entry area. It holds the following parameters; 'row', 'column', 'padx' and so on which are respectively with their corresponding value or weights.

2.3 Diagrams

- a. User input area :-** This is the text area generated by the Text() widget from tkinter and it is used to take the input of the sentence from the user. This is further passed to the detect_sentiment() function to compute the score.

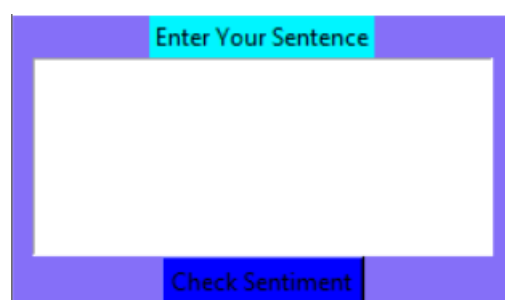


Fig 1.2 : Text area for user input

- b. Polarity score area :-** This is the section of the GUI which displays the whole scores of the sentence along with the intensity of the sentiment that has been hence detected. This section has fields which show the Positive, Negative, Neutral and Overall field in terms of percentage.



Fig 1.3 : Section that displays the sentiment

c. Clear and Exit :- This is the section of the GUI that enables the user to clear the contents of the above two areas when giving a new input in the form of a sentence. The clear button performs the clear all fields operation and the exit button allows the user to exit the GUI application window.

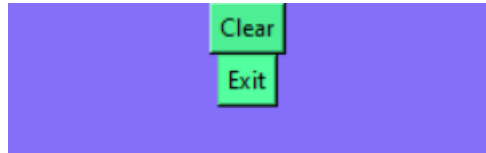


Fig 1.4 : Section depicting clear and exit button

Chapter – 3 Implementation

3.1 Modules used in the code

As there are a vast majority of modules that can be used in the detecting of sentiments like nltk, spaCY and other modules for natural language processing that would be feasible in a context where we would perform a in depth analysis. However to make the process as simple as possible we chose to work with vader sentiment module for the analysis and the tkinter package for construction of the GUI window. Hence the packages used here are :

- vaderSentiment
- Tkinter

❖ vaderSentiment:

VADER stands for Valence Aware Dictionary for Sentiment Reasoning.

Sentiment analysis is a text analysis method that detects polarity within the text, whether a whole document, paragraph, sentence, or clause.

Sentiment analysis aims to measure the attitude, sentiments, evaluations, attitudes, and emotions of a speaker/writer based on the computational treatment of subjectivity in a text.

We are using the SentimentIntensityAnalyzer .

VADER's SentimentIntensityAnalyzer() takes in a string and returns a dictionary of scores in each of four categories:

- negative
- neutral
- positive
- compound

The sentence is overall negative if the score is ≤ -0.05 and positive overall if the sentence has a score of ≥ 0.05 and the intermediate would be neutral.

❖ Tkinter :

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. In this library we use the following widgets : -

1. Entry()
2. Text()
3. Label()
4. Button()
5. Grid()
6. Get()
7. Insert()

3.2 Flow of Code

Here, moving on to the first function used which is named “clearAll()”(user-defined) does the job of clearing all the contents of the entries that have been made in the entry boxes and the text areas; it has been given few pre-defined functions in it namely `negativeField.delete(...,...)` , `neutralField.delete(...,...)` , `positiveField.delete(...,...)` , `overallField.delete(...,...)`. These function clear the contents in the positive, neutral, negative and the overall fields of the window respectively and they take 2 parameters 0 and END. While there is a function called `textArea.delete(1.0,END)` , that deletes the contents of text area.

Now, moving on to the most important part of the code i.e., the function , “detect_sentiment()” (user defined). Which consists of:

The line `sentence = textArea.get("1.0", "end")` ; gets a whole input content from text box .
`sid_obj = SentimentIntensityAnalyzer()` ; Creates a `SentimentIntensityAnalyzer` object.
`sentiment_dict = sid_obj.polarity_scores(sentence)` ;
`polarityscores` method of `SentimentIntensityAnalyzer`,object gives a sentiment dictionary.
 which contains pos, neg, neu, and compound scores.

Further we insert the required computed percentage into the positive, neutral and negative fields, using the insert methods.

3.3 Code

```
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
from tkinter import *
# 1.function to print sentiments of the sentence.
def detect_sentiment():
    # get a whole input content from text box
    sentence = textArea.get("1.0", "end")

    # Create a SentimentIntensityAnalyzer object.
    sid_obj = SentimentIntensityAnalyzer()

    # polarity_scores method of SentimentIntensityAnalyzer
    # object gives a sentiment dictionary.
    # which contains pos, neg, neu, and compound scores.
    sentiment_dict = sid_obj.polarity_scores(sentence)

    string = str(sentiment_dict['neg']*100) + "% Negative"
    negativeField.insert(10, string)

    string = str(sentiment_dict['neu']*100) + "% Neutral"
    neutralField.insert(10, string)

    string = str(sentiment_dict['pos']*100) + "% Positive"
    positiveField.insert(10, string)

    # decide sentiment as positive, negative and neutral
    if sentiment_dict['compound'] >= 0.05 :
        string = "Positive"
```

```
elif sentiment_dict['compound'] <= - 0.05 :
    string = "Negative"
```

```
else :
    string = "Neutral"
```

```
overallField.insert(10, string)
```

2.Function for clearing the contents of all entry boxes And text area.

```
def clearAll() :
```

```
    # deleting the content from the entry box
    negativeField.delete(0, END)
    neutralField.delete(0, END)
    positiveField.delete(0, END)
    overallField.delete(0, END)
```

```
    # whole content of text area is deleted
    textArea.delete(1.0, END)
```

```
def quit(root_obj):
```

```
    root_obj.destroy()
```

```
    # Driver Code
```

```
if __name__ == "__main__" :
```

```
    # Create a GUI window
    gui = Tk()
```

```
    # Set the background colour of GUI window
    #gui.config(background = "light green")
    gui.config(background = "#856ff8")
```

```
    # set the name of tkinter GUI window
    gui.title("Sentiment Detector")
```

```
    # Set the configuration of GUI window
    gui.geometry("250x400")
```

```
    # create a label : Enter Your Task
    enterText = Label(gui, text = "Enter Your Sentence",
                      bg = "turquoise1")
```

```
    # create a text area for the root with lunida 13 font text area is for writing the content
```

```
    textArea = Text(gui, height = 5, width = 25, font = "lucida 13")
```

```
    # create a Submit Button and place into the root window when user press the button,
```

#the command or function affiliated to that button is executed

```
check = Button(gui, text = "Check Sentiment", fg = "Black",  
               bg = "blue", command = detect_sentiment)
```

```
# Create a negative : label  
negative = Label(gui, text = "Negative Rating is : ",  
                 bg = "magenta")
```

```
# Create a neutral : label  
neutral = Label(gui, text = "Neutral Rating is : ",  
                bg = "cyan")
```

```
# Create a positive : label  
positive = Label(gui, text = "Positive Rating is : ",  
                 bg = "magenta")
```

```
# Create a overall : label  
overall = Label(gui, text = "Overall Rating is : ",  
                bg = "cyan")
```

```
# create a text entry box  
negativeField = Entry(gui)
```

```
# create a text entry box  
neutralField = Entry(gui)
```

```
# create a text entry box  
positiveField = Entry(gui)
```

```
# create a text entry box  
overallField = Entry(gui)
```

```
# create a Clear Button and place into the root window  
# when user press the button, the command or  
# function affiliated to that button is executed .  
clear = Button(gui, text = "Clear", fg = "Black",  
               bg = "SeaGreen2", command=lambda: clearAll(text_area=True))
```

```
# create a Exit Button and place into the root window  
# when user press the button, the command or  
# function affiliated to that button is executed .  
Exit = Button(gui, text = "Exit", fg = "Black",  
              bg = "SeaGreen1", lambda: quit(gui))
```

```
# grid method is used for placing  
# the widgets at respective positions  
# in table like structure.  
enterText.grid(row = 0, column = 2)
```

```
textArea.grid(row = 1, column = 2, padx = 10, sticky = W)

check.grid(row = 2, column = 2)

negative.grid(row = 3, column = 2)

neutral.grid(row = 5, column = 2)

positive.grid(row = 7, column = 2)

overall.grid(row = 9, column = 2)

negativeField.grid(row = 4, column = 2)

neutralField.grid(row = 6, column = 2)

positiveField.grid(row = 8, column = 2)

overallField.grid(row = 10, column = 2)

clear.grid(row = 11, column = 2)

Exit.grid(row = 12, column = 2)

# start the GUI
gui.mainloop()
```

3.3 Tools used for GUI

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. In this library we use the following widgets :-

1. Entry()
 2. Text()
 3. Label()
 4. Button()
 5. Grid()
 6. Get()
 7. Insert()
- Entry widget provides a feature to display the calculated output of the intensity score into the specified content of the GUI window .
 - Text widget provides the user to input the sentence. It further stores it in a variable and passes it on to the detect_sentiment() function.
 - Label is the name given to the text boxes in the GUI window. Using this function we can create a Label and also change color of background and foreground.

- Button is a function that allows us to perform a certain task when the user clicks on the button. The command would be specified in the arguments.
- Grid is essential for spacing of the labels and the text boxes . It divides the screen into rows and columns and allows us to space the boxes at desired place.
- Get is used in the capturing of the input from the text box and into the variable it is assigned to.
- Insert is used in displaying the output in the text box by specifying the start index and end index

Chapter – 4

Conclusion and Future Scope

CONCLUSION:

The field of sentiment analysis is an exciting new research direction due to large number of real-world applications where discovering people's opinion is important in better decision-making. The development of techniques for the document-level sentiment analysis is one of the significant components of this area. Recently, people have started expressing their opinions on the Web that increased the need of analysing the opinionated online content for various real-world applications. A lot of research is present in literature for detecting sentiment from the text. Still, there is a huge scope of improvement of these existing sentiment analysis models. Existing sentiment analysis models can be improved further with more semantic and common sense knowledge.

In this research work it has been tried to develop a system which helped in understanding the user sentiments or analyse and understand the user requirements on a product and thus infer about the same. The data is processed using the Natural Language Processing and a statistical value was associated to the opinion and sentiments of the people so that inferences can be derived on the numerical values, generally speaking numerical value is easy to derive and conclusions can be made easily from it. . The system was also able to interpret and give the numerical values to the sentence which were grammatically incorrect. The sentence was studied and interpreted closely.

There are certain limitations . Discussing about the advantages , it enabled to read and understand the user sentiments and a numerical value was given to the user sentiment values which enabled to come to the conclusive results as well produce results based on the data currently available.

Now and the upcoming years continues to drive the analytics machine forward. With more and more organizations turning to sentiment analysis to measure and predict outcomes, as well as better understand consumer behaviours , these tools are quickly building a reputation that is going to help propel it forward into the future and towards deeper and more accurate conclusions and insights.

FUTURE SCOPE:

The future of sentiment analysis is going to continue to dig deeper, far past the surface of the number of likes, comments and shares, and aim to reach, and truly understand, the significance of social media interactions and what they tell us about the consumers behind the screens. This forecast also predicts broader applications for sentiment analysis – brands will continue to leverage this tool, but so will individuals in the public eye, governments, non-profits, education centres and many other organizations.

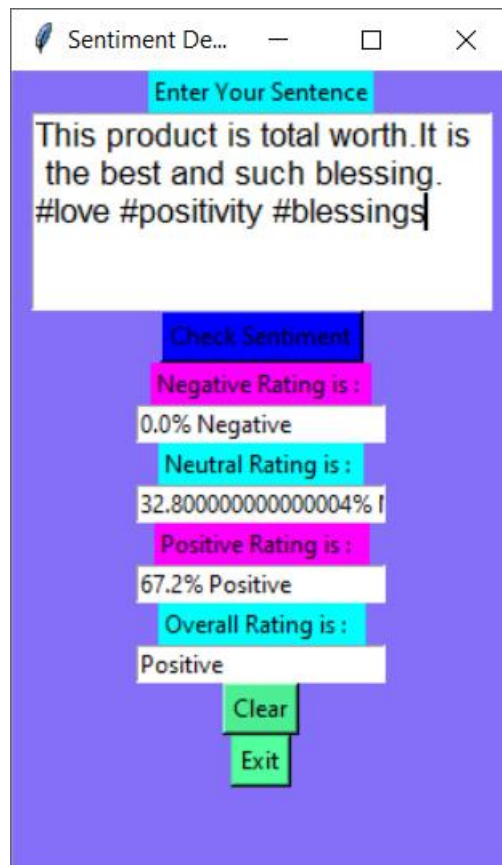
This can be extended, can be used for finding the trend analysis. To elaborate, the time of the comments can be retrieved and stored. The time of the comments can help in inferring about the opinion on the product on different periods and thus analyse or study the user comments in different periods of time. This is basically referred to as the trend analysis and is widely used by the organizations from time to time for the product study and sales forecasting.

Chapter – 5

Output and Screenshots

- **Test Case – 1 : Verifying for a Positive sentence**

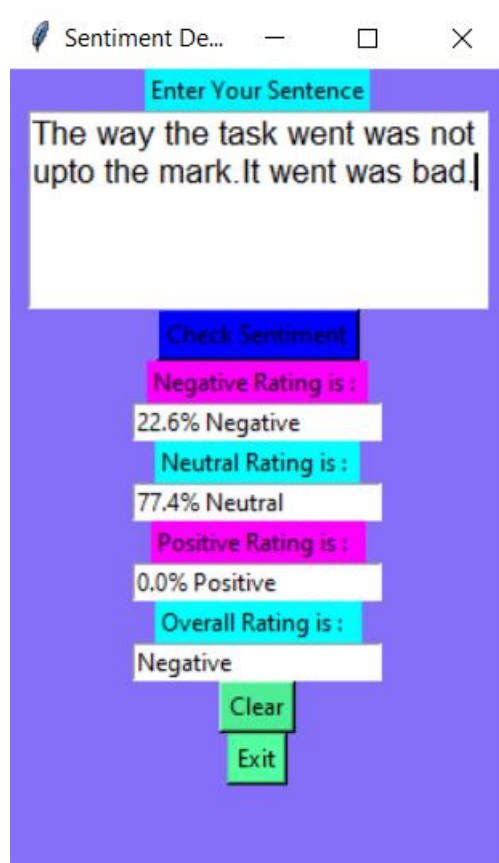
The input that is to be passed into the analyzer would be a positive sentence in order to check the underlying division between the different Sentiments.



Negative : 0.0%
Neutral : 32.8%
Positive : 67.2%
Overall : Positive

- **Test Case – 2 : Verifying for a Negative sentence**

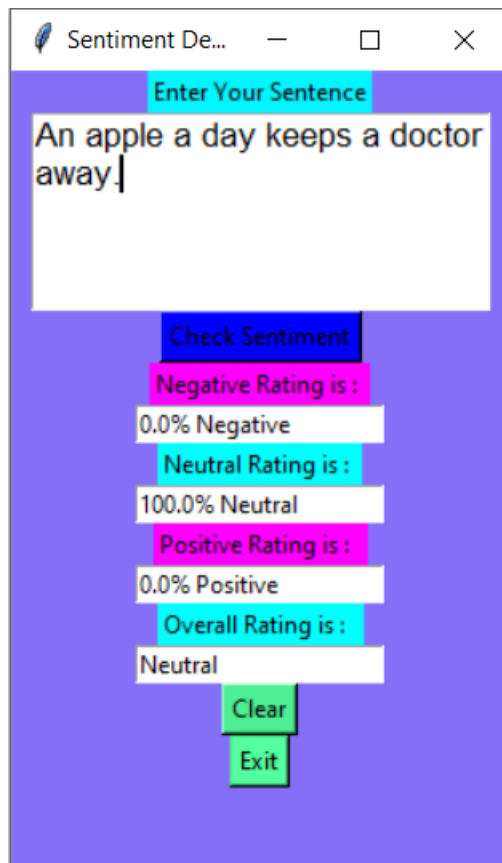
The input that is to be passed into the analyzer would be a negative sentence in order to check the underlying division between the different Sentiments.



Negative : 22.6%
Neutral : 77.4%
Positive : 0.0%
Overall : Negative

- **Test Case – 3 : Verifying for a Neutral sentence**

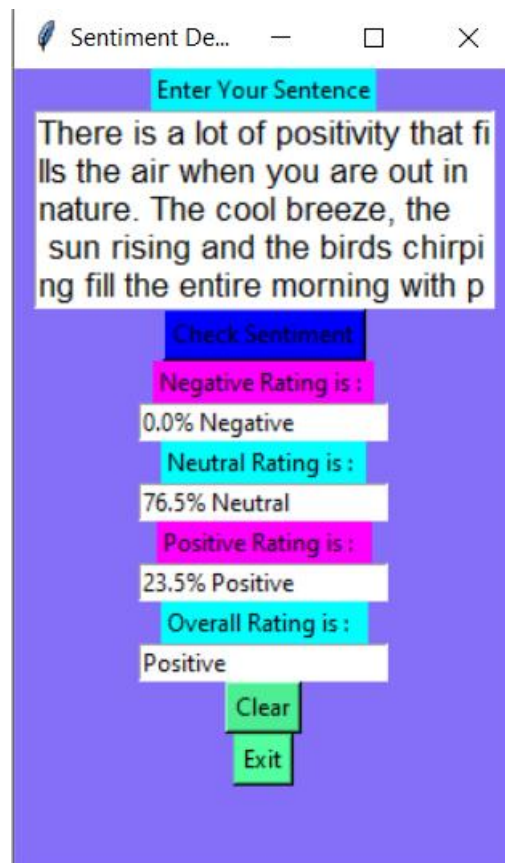
The input that is to be passed into the analyzer would be a neutral sentence in order to check the underlying division between the different Sentiments.



Negative : 0.0%
Neutral : 100.0%
Positive : 0.0%
Overall : Neutral

- **Test Case – 4 : Verifying for a Paragraph**

The input that is to be passed into the analyzer would be a negative sentence in order to check the underlying division between the different Sentiments.



Negative : 0.0%
Neutral : 76.5%
Positive : 23.5%
Overall : Positive

References and Bibliography

Blogs :

1. Lexalytics
2. SciBite , Text Mining, Semantics & Data Intelligence

Books :

1. R. Mehta, D. Mehta, D. Chheda, C. Shah, and P. M. Chawan, "Sentiment analysis and influence tracking using twitter," International Journal of Advanced Research in Computer Science and Electronics Engineering (IJARCSEE), vol. 1, no. 2, p. pp-72, 2012.
2. Akkaya, C., J. Wiebe, and R. Mihalcea. Subjectivity word sense disambiguation. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP-2009), 2009.
3. Andreevskaya, A. and S. Bergler. Mining WordNet for fuzzy sentiment: Sentiment tag extraction from WordNet glosses. In Proceedings of Conference of the European Chapter of the Association for Computational Linguistics (EACL-06), 2006.
4. Qiu, L., W. Zhang, C. Hu, and K. Zhao. Selc: a self-supervised model for sentiment classification. In Proceeding of the 18th ACM conference on Information and knowledge management (CIKM2009), 2009.
5. Tong, R. An operational system for detecting and tracking opinions in on-line discussion. In Proceedings of SIGIR Workshop on Operational Text Classification, 2001.
6. Yessenalina, A., Y. Yue, and C. Cardie. Multi-level Structured Models for Document-level Sentiment Classification. In Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2010), 2010.