

BIRLA INSTITUTE OF TECHNOLOGY MESRA, RANCHI CAMPUS



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CA545 Natural Language Processing Assignment File

on

“Sentiment Analysis on Real Phone Calls”

SUBMITTED TO:

Dr. Shruti Garg

(Asst. Prof.)

SUBMITTED BY:

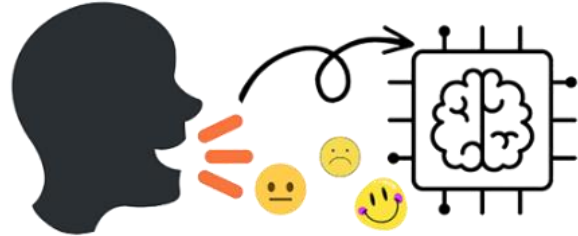
- | | |
|------------------------------|---------------------|
| 1. Lucky verma | MCA/10062/22 |
| 2. Dewesh Chopra | MCA/10032/22 |
| 3. Sameer Horo | MCA/10063/22 |
| 4. Ankit Kumar Pandey | MCA/10065/22 |

INDEX

S. No.	Date	Topics	Page No.
1.	01/11/2023	Introduction to Sentiment Analysis	3
2.	01/11/2023	Types Of Sentiment Analysis	4
3.	01/11/2023	Purpose Of Sentiment Analysis on Real Phone Calls	5
4.	01/11/2023	Tools and Libraries used in Sentiment Analysis on Real Phone Calls	5
5.	01/11/2023	Flowchart of Sentiment Analysis on Real Phone Calls	6
6.	01/11/2023	Process of Sentiment Analysis on Real Phone Calls	7-8
7.	01/11/2023	Implementation (Code)	9-13
8.	01/11/2023	Output	14-15
9.	01/11/2023	Applications	16
10.	01/11/2023	Future Scope	17

Introduction to Sentiment Analysis

Sentiment analysis can be defined as analysing the positive or negative sentiment of the customer in text. The contextual analysis of identifying information helps businesses understand their customers' social sentiment by monitoring online conversations.



As customers express their reviews and thoughts about the brand more openly than ever before, sentiment analysis has become a powerful tool to monitor and understand online conversations. Analysing customer feedback and reviews automatically through survey responses or social media discussions allows you to learn what makes your customer happy or disappointed. Further, you can use this analysis to tailor your products and services to meet your customer's needs and make your brand successful.

Recent advancements in machine learning and deep learning have increased the efficiency of sentiment analysis algorithms. You can creatively use advanced **artificial intelligence and machine learning** tools for doing research and draw out the analysis.

For example, sentiment analysis can help you to automatically analyze 5000+ reviews about your brand by discovering whether your customer is happy or not satisfied by your pricing plans and customer services. Therefore, you can say that the application of sentiment is endless.

- **Purpose:** It helps to comprehend and quantify emotions expressed in text, enabling businesses to gauge public opinion, customer satisfaction, and brand perception.
- **Inventors:**
 - **Gerald Salton:** Considered a pioneer in sentiment analysis, Salton made significant contributions to the field in the 1980s.
 - **Bo Pang and Lillian Lee:** Their influential research in the early 2000s laid the foundation for key methodologies and principles in sentiment analysis.

Types of Sentiment Analysis

The sentiment analysis process mainly focuses on **polarity**, i.e., positive, negative, or neutral. Apart from polarity, it also considers the feelings and **emotions (happy, sad, angry, etc.)**, **intentions (interested or not interested)**, or **urgency (urgent or not urgent)** of the text.

1. Fine-grained Sentiment Analysis:

If your business requires the polarity precisions, then you can classify your polarity categories into the following parts: [Very positive, Positive, Neutral, Negative, Very Negative]

For polarity analysis, you can use the 5-star ratings as a customer review where very positive refers to a five-star rating and very negative refers to a one-star rating.



Fine-grained
Sentiment Analysis

2. Emotion Detection:

It helps to detect customer **emotions like happiness, disappointment, anger, sadness**, etc. Here, you can use **sentiment lexicons** or **complex machine learning algorithms** to identify the feelings.



Emotion
Detection

DISADVANTAGES: using sentiment lexicons it may be confusing to understand human emotion.

3. Aspect-Based Sentiment Analysis:

Aspect-based sentiment analysis **extracts specific aspects or features** that customers mention in their **reviews, categorizing sentiments as positive or negative** for each identified aspect.



Aspect-based
Sentiment Analysis

4. Multilingual sentiment analysis:

Multilingual sentiment analysis is complex compared to others as it includes **many preprocessing and resources available online (i.e., sentiment lexicons)**.



Multilingual
Sentiment Analysis

Businesses value the feedback of the customer regardless of their geography or language. Therefore, It helps you **identify customer sentiment irrespective of location or language difference**.

Sentiment Analysis on Real Phone Calls

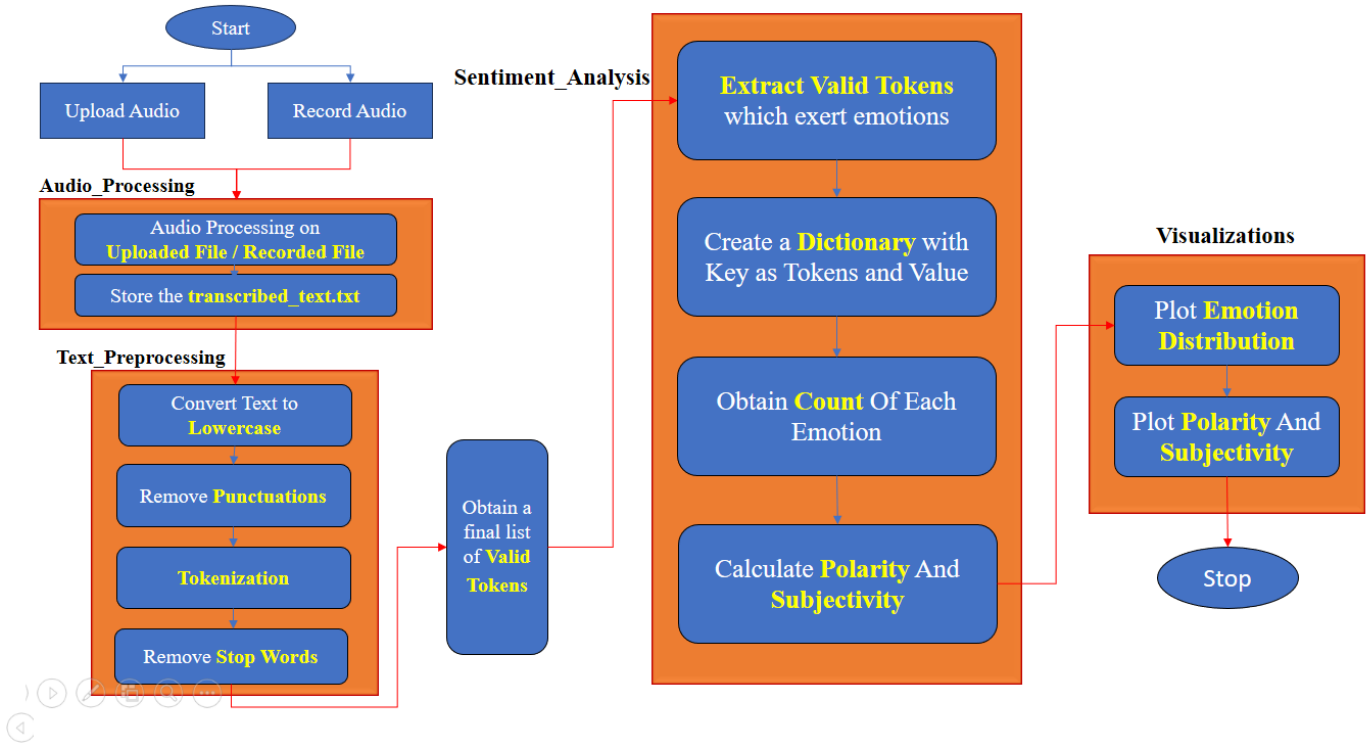
Purpose of Sentiment Analysis on Real Phone Calls

- The primary objective of this project is to develop a system capable of extracting sentiment from spoken language, specifically phone call conversations.
- By understanding the sentiment of the conversation between 2 Person on call, we aim to gain insights into person's overall sentiments trends.

Tools and Libraries used in Sentiment Analysis on Real Phone Calls

- **Tkinter:** Integrated into the project to enhance the user interface, Tkinter provided an interactive platform for visualizing sentiment data and creating a user-friendly experience.
- **Speech Recognition:** we implemented a speech recognition library to convert spoken language into text for further analysis.
- **PyAudio:** By using PyAudio library we are capturing audio data from mic.
- **Matplotlib:** Employed for generating bar graphs, Matplotlib was used to visualize the sentiment analysis results, providing an intuitive means to compare and contrast sentiment scores.

Flow-Chart of Sentiment Analysis on Real Phone Calls



Process of Sentiment Analysis on Real Phone Calls

+ Data Collection Process:

- The success of our sentiment analysis project relies heavily on the quality and relevance of the data we collect from phone calls.

+ Methods of Data Collection:

- **Call Recording:** We recorded phone calls between 2 friends or people.
- **Transcription:** The recorded calls were then transcribed into text format using our speech recognition component. This step was crucial in preparing the data for sentiment analysis.
- **Speech Recognition:** Speech recognition is a crucial component of our project. It enables us to convert spoken language from phone calls into text format, which can then be further analyzed for sentiment.

+ How Speech Recognition Works:

- **Audio Input:** The process starts with receiving audio data from phone calls. This can be in the form of recorded audio files or real-time streaming.
- **Decoding:** Using both acoustic and language models, the system decodes the audio input into a sequence of words or tokens.
- **Output Text:** The result is a text representation of the spoken language, which can be used for further analysis.

+ How Sentiment Analysis Works:

- **Text Preprocessing:** Before analysis, the text data undergoes preprocessing steps like tokenization, removing punctuation, and converting to lowercase.
- **Feature Extraction:** Relevant features, such as words or phrases, are extracted from the preprocessed text. These features are used to identify patterns associated with sentiment.
- **Sentiment Classification:** We used Lexicon Sentiment Analysis the extracted features are used to classify the sentiment into categories sad, angry, happy etc.
- **Visualization:** Data visualization is a crucial aspect of our project. It allows us to present the results of our sentiment analysis in a clear and understandable format.
 - **Emotion Distribution Plot:** Shows the distribution plot of emotions present in the phone call.
 - **Polarity and Subjectivity Plot:** Shows the value of polarity and subjectivity.

In sentiment analysis, **polarity** and **subjectivity** are two metrics used to assess the sentiment expressed in a given text.

Polarity

Polarity quantifies the sentiment orientation expressed in the text, whether it's positive, negative, or neutral. It typically ranges from -1 to 1:

- -1 denotes extremely negative sentiment.
- 0 denotes a neutral sentiment.
- 1 denotes extremely positive sentiment.

➤ **Formula for Polarity:**

The formula for polarity is often calculated by dividing the number of emotional words (positive or negative) by the total number of words in the text:

$$\text{Polarity} = \frac{\text{Total number of words}}{\text{Number of emotional words}}$$

Subjectivity

Subjectivity measures how subjective or opinionated the text is, reflecting the degree to which the text expresses personal opinions, feelings, or biases rather than factual information. The typical range for subjectivity is from 0 to 1:

- 0 denotes an objective or factual statement.
- 1 denotes highly subjective or opinionated text.

➤ **Formula for Subjectivity**

The formula for subjectivity varies across approaches. A common formula involves the count of unique emotional words divided by the total count of emotional words:

$$\text{Subjectivity} = \frac{\text{Number of unique emotional words}}{\text{Total number of emotional words}}$$

Implementation

Main.py

```
import tkinter as tk
from tkinter import filedialog

from audio_processing import process_audio
from text_preprocessing import text_cleanup
import speech_recognition as sr

def upload_file():
    file_path = filedialog.askopenfilename(initialdir="/", title="Select Recording File",
                                           filetype=((("WAV files", "*.wav"), ("all files", "*.*"))))

    if file_path:
        # Process the uploaded file
        process_audio(file_path)

        # Preprocess the text data
        text_cleanup()

        # Perform sentiment analysis
        # perform_sentiment_analysis()

        # Generate graphs
        # Via 'perform_sentiment_analysis()' a call is made

        # Update GUI with analysis results
        with open("transcribed_text.txt", "r") as file:
            transcribed_text = file.read()
            text_display.delete(1.0, tk.END) # Clear previous text
            text_display.insert(tk.END, transcribed_text)

    else:
        # Handle if no file is selected
        text_display.delete(1.0, tk.END) # Clear previous text
        text_display.insert(tk.END, "No file selected")

def listen_microphone():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        audio = r.listen(source)
    try:
        transcribed_text = r.recognize_google(audio)
        with open("transcribed_text.txt", "w") as file:
            file.write(transcribed_text)
        text_display.delete(1.0, tk.END)
        text_display.insert(tk.END, transcribed_text)
```

```

# Preprocess the text data
text_cleanup()

# # Perform sentiment analysis
# perform_sentiment_analysis()

# Generate graphs
# Via 'perform_sentiment_analysis()' a call is made

except sr.UnknownValueError:
    text_display.delete(1.0, tk.END)
    text_display.insert(tk.END, "Could not understand the audio")
except sr.RequestError:
    text_display.delete(1.0, tk.END)
    text_display.insert(tk.END, "Error. Please check your internet connection")

root = tk.Tk()

root.title("Sentiment Analysis on Real Phone Calls")
root.geometry("640x480")

# Heading
heading_label = tk.Label(root, text="Sentiment Analysis on Real Phone Calls", font=("Arial", 20))
heading_label.pack(pady=20)

# Frame for buttons
button_frame = tk.Frame(root)
button_frame.pack()

# Upload Button
upload_button = tk.Button(button_frame, text="Upload Recording", command=upload_file, bg="blue",
fg="white")
upload_button.pack(side=tk.LEFT, padx=10, pady=10)

# Listen Button for Microphone
listen_button = tk.Button(button_frame, text="Listen", command=listen_microphone, bg="green",
fg="white")
listen_button.pack(side=tk.LEFT, padx=10, pady=10)

# Text Display Area for Transcribed Audio
text_display = tk.Text(root, height=20, width=70)
text_display.pack(pady=20)

root.configure(bg='lightgray') # Change background color

root.mainloop()

```

audio_processing.py

```
import speech_recognition as sr

def process_audio(file_path):
    r = sr.Recognizer()

    with sr.AudioFile(file_path) as source:
        audio = r.record(source)
        try:
            text = r.recognize_google(audio)
            with open("transcribed_text.txt", "w") as file:
                file.write(text)
        except sr.UnknownValueError:
            print("Google Speech Recognition could not understand the audio")
        except sr.RequestError as e:
            print("Could not request results from Google Speech Recognition service; {0}".format(e))
```

Text_preprocessing.py

```
from sentiment_analysis import perform_sentiment_analysis

import string

def text_cleanup():
    # Read the transcribed text
    with open("transcribed_text.txt", "r") as file:
        transcribed_text = file.read()

    # converting to lowercase
    lower_case = transcribed_text.lower()

    # Removing punctuations
    cleaned_text = lower_case.translate(str.maketrans("", "", string.punctuation))

    # Tokenize and remove 'stop' words
    tokenized_words = cleaned_text.split()
    stop_words = ["i", "me", "my", "myself", "we", "our", "ours", "ourselves", "you", "your", "yours",
"yourself",
    "yourselves", "he", "him", "his", "himself", "she", "her", "hers", "herself", "it", "its", "itself",
    "they", "them", "their", "theirs", "themselves", "what", "which", "who", "whom", "this", "that",
    "these",
    "those", "am", "is", "are", "was", "were", "be", "been", "being", "have", "has", "had", "having",
    "do",
    "does", "did", "doing", "a", "an", "the", "and", "but", "if", "or", "because", "as", "until",
"while",
```

```

"before",
    "of", "at", "by", "for", "with", "about", "against", "between", "into", "through", "during",
    "after", "above", "below", "to", "from", "up", "down", "in", "out", "on", "off", "over", "under",
    "again",
    "further", "then", "once", "here", "there", "when", "where", "why", "how", "all", "any", "both",
    "each",
    "few", "more", "most", "other", "some", "such", "no", "nor", "not", "only", "own", "same",
"so",
    "than",
    "too", "very", "s", "t", "can", "will", "just", "don", "should", "now"]

# Removing stop words from the tokenized words list
final_words = []
for word in tokenized_words:
    if word not in stop_words:
        final_words.append(word)

print('Set of final words from the audio data:', final_words)
print() # For formatting

# Calling 'perform_sentiment_analysis()' for further processing
perform_sentiment_analysis(final_words)

```

sentiment_analysis.py

```

# Sentiment Analysis using Lexicon Based Approach

from visualizations import generate_graphs

def perform_sentiment_analysis(final_words):
    # Read emotions from emotions.txt
    emotions = {}
    with open("emotions.txt", "r") as file:
        for line in file:
            word, emotion = line.replace("\n", "").replace(",", "").replace("'", "").strip().split(':')
            emotions[word] = emotion

    # Perform sentiment analysis
    word_emotions = []
    print('Set of tokens that matched lexicon keys:')
    for word in final_words:
        if word.lower() in emotions:
            print('Word: {}, Emotion: {}'.format(word, emotions[word]))
            word_emotions.append(emotions[word.lower()])

    print() # For formatting

    # Count the occurrences of each emotion
    emotion_counts = {emotion: word_emotions.count(emotion) for emotion in set(word_emotions)}

    # Output the sentiment analysis results
    print("Emotion counts:", emotion_counts)

```

```

# Calculate polarity and sensitivity
total_words = len(final_words)
total_emotions = sum(emotion_counts.values())

polarity = total_emotions / total_words if total_words > 0 else 0

unique_emotions = len(set(word_emotions))
subjectivity = (unique_emotions / total_emotions) if total_emotions > 0 else 0

# Output the sentiment analysis results
print("Polarity:", polarity)
print("Subjectivity:", subjectivity)

# Extract the emotions and counts for plotting
emotions = list(emotion_counts.keys())
counts = list(emotion_counts.values())

# Pass required data to 'generate_graphs' for visualizations
generate_graphs(emotions, counts, polarity, subjectivity)

```

Visualizations.py

```

import matplotlib.pyplot as plt

def generate_graphs(emotions, counts, polarity, subjectivity):
    # Creating subplots for emotion distribution, polarity, and subjectivity
    fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10, 6))

    # Emotion Distribution
    ax1.bar(emotions, counts)
    ax1.set_xlabel('Emotions')
    ax1.set_ylabel('Counts')
    ax1.set_title('Emotion Distribution')

    # Rotate the x-axis labels by 45 degrees
    ax1.set_xticks(emotions)
    ax1.set_xticklabels(emotions, rotation=45, ha="right")

    # Bar labels(polarity, subjectivity) and values
    bars = ['Polarity', 'Subjectivity']
    values = [polarity, subjectivity]
    colors = ['blue', 'green']

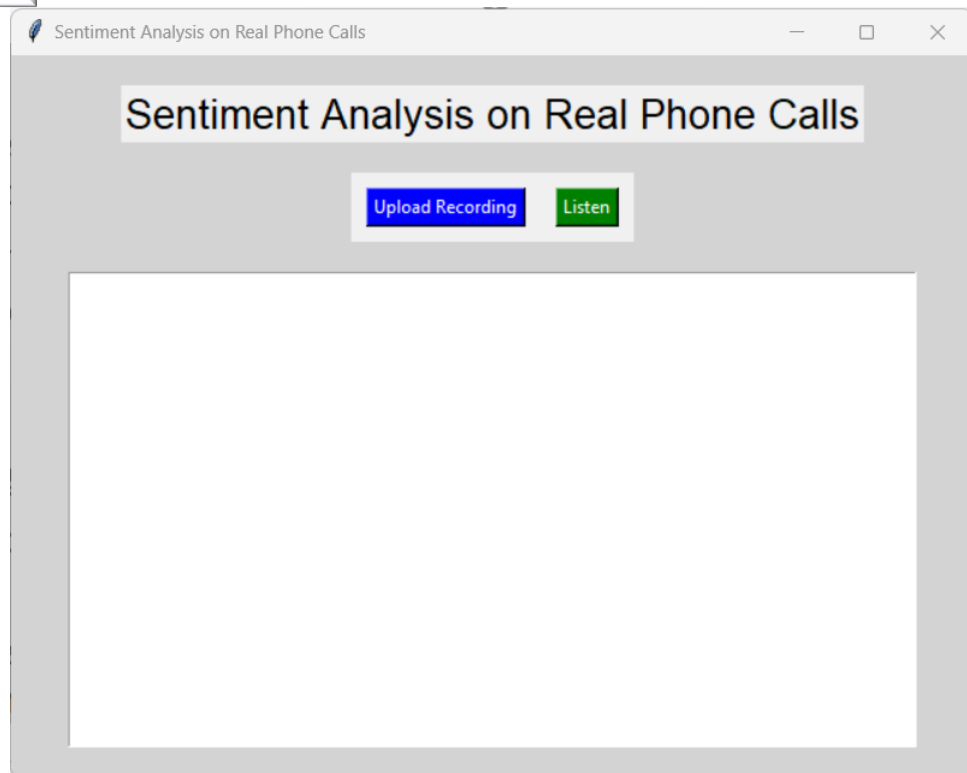
    ax2.bar(bars, values, color=colors)
    ax2.set_ylabel('Value')
    ax2.set_title('Polarity and Subjectivity')

    # Adjust layout and save the graph as an image
    plt.tight_layout()
    plt.savefig('sentiment_analysis_visualization.png', format='png')
    plt.show()

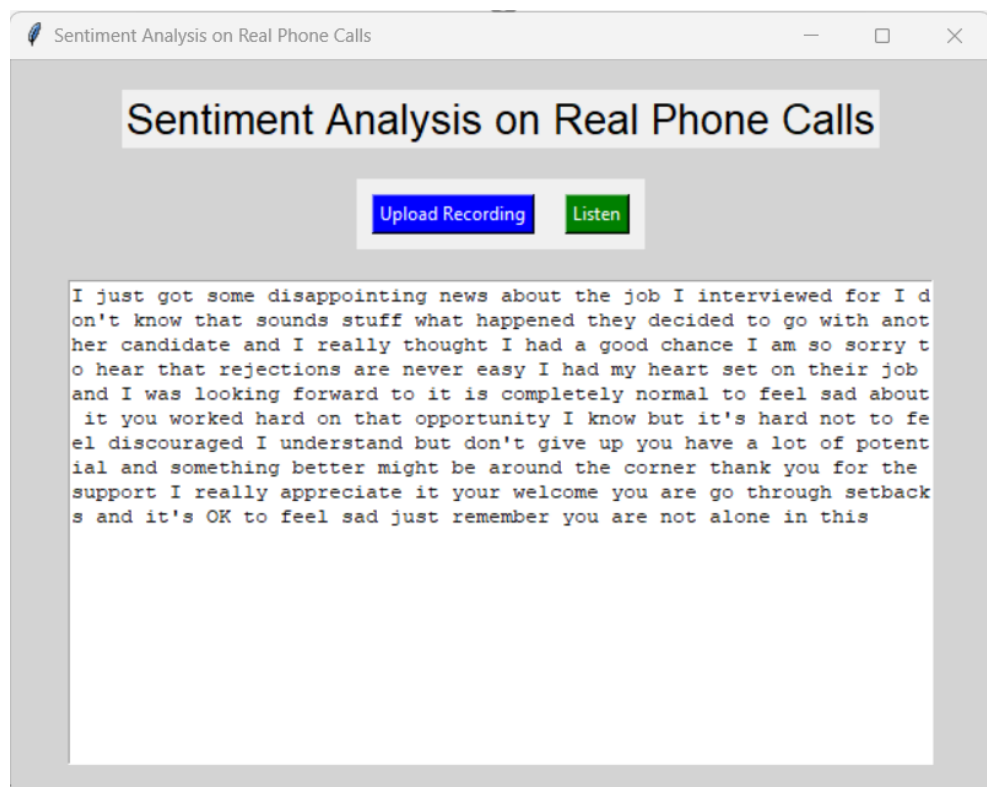
```

OUTPUT

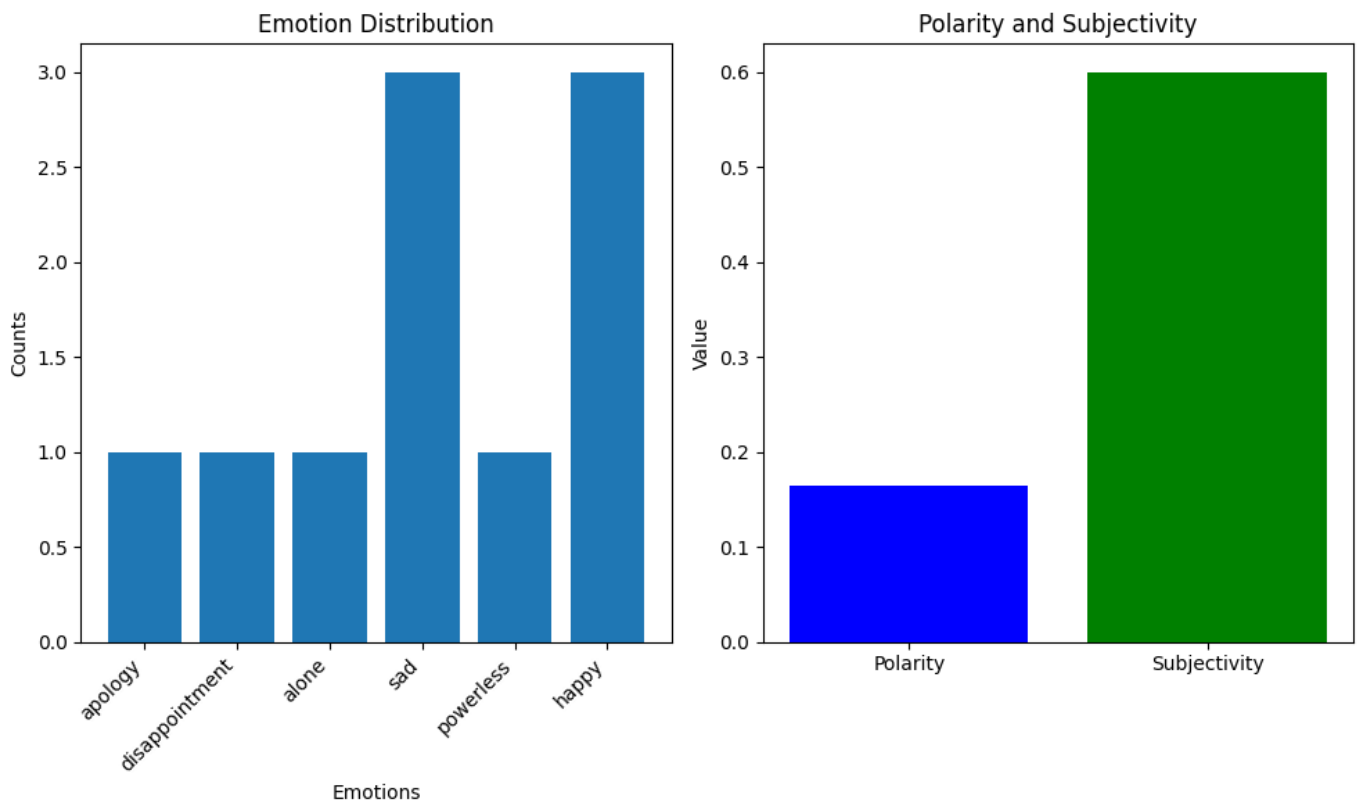
Main GUI



Transcribed Text Generated in text_display



Emotion Distribution & (Polarity & Subjectivity)



Terminal Output

```
E:\python.exe "C:\Users\user\OneDrive\Desktop\Dewesh SentimentAn\main.py"
Set of final words from the audio data: ['got', 'disappointing', 'news', 'job', 'interviewed', 'dont', 'know', 'sounds', 'stuff', 'happened', 'decided', 'go', 'another', 'cand:

Set of tokens that matched lexicon keys:
Word: disappointing, Emotion: sad
Word: good, Emotion: happy
Word: sorry, Emotion: apology
Word: rejections, Emotion: disappointment
Word: sad, Emotion: sad
Word: discouraged, Emotion: powerless
Word: thank, Emotion: happy
Word: welcome, Emotion: happy
Word: sad, Emotion: sad
Word: alone, Emotion: alone

Emotion counts: {'apology': 1, 'disappointment': 1, 'alone': 1, 'sad': 3, 'powerless': 1, 'happy': 3}
Polarity: 0.16393442622950818
Subjectivity: 0.6
```

Applications of Sentiment Analysis on Real Phone Calls



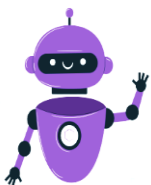
Customer Service: Analyzing sentiment in customer service calls can help assess customer satisfaction, detect dissatisfaction or frustration, and identify trends in queries or complaints. This analysis enables companies to enhance their services and improve customer experience.



Market Research & Analysis: Analyzing sentiments in phone calls provides valuable insights into market trends, customer preferences, and feedback on products or services. It assists in market research efforts, enabling companies to adapt offerings based on customer sentiments.



Fraud Detection: In financial services, sentiment analysis can help in identifying fraudulent calls by analyzing voice tones and expressions, potentially uncovering suspicious behavior or irregularities.



Emotionally Intelligent Chatbots: Sentiment analysis on calls can aid in training AI-driven chatbots to understand and respond appropriately to the emotional context of customer inquiries or issues.

Future Scope of Sentiment Analysis on Real Phone Calls

- ✚ **Contextual understanding** in sentiment analysis on phone calls is a developing area for improvement.
- ✚ **Advancements in technology**, including machine learning and NLP, strive to better grasp the context of conversations to interpret sentiments accurately.
- ✚ **Identifying the polarity** (positive, negative, neutral) in these contexts is essential for gauging customer sentiments during calls.
- ✚ **Irony and sarcasm recognition** in context can significantly impact polarity detection in sentiment analysis.