

INTERACTION ANALYSIS OVER SPEECH FOR CALL CENTRE

Submitted in partial fulfillment of the requirements
of the degree of

B. E. Computer Engineering

By

**Sneha Chitale
AaroHi Mahajan
Shivam Mehta**

Guide:

Prof. Dhananjay Raut
Assistant Professor

Department of Computer Engineering
Watumull Institute of Electronic Engineering and Computer Technology
University of Mumbai
2019-2020

Abstract

Customer Service Center is the second most important consideration just after the actual product. Also, customer service is one of the biggest contributors to the cost component for any firm. We aim to apply well-known data mining techniques to the problem of predicting the quality of interactions like those done in call centers and the problem of predicting the quality of service. The analysis of call center conversations will provide useful insights for enhancing Call Center Analytics to a level that will enable new metrics and key performance indicators (KPIs) beyond the standard approach. These metrics rely on understanding the dynamics of conversations by highlighting the way participants discuss topics. The main focus will be to reduce the average handling time, is a call center metric for the average duration of one transaction, typically measured from the customer's initiation of the call and including any hold time, talk time and related tasks that follow the transaction. Get real-time solution. The main operations will be speaker diarization, speech to text, agent analysis, emotion recognition and other measures to help with the analysis. We will use RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song) for emotion analysis, consisting of vocal emotional expressions in sentences spoken in a range of basic emotional states (happy, sad, anger, fear, disgust, surprise and calm). Emotion recognition is done by extracting features from the audio from its Mel-frequency cepstral coefficients (MFCCs) and passing it through a convolutional neural network. All of this will happen in real time as the call is taking place.

Contents

Chapter	Contents	Page No.
1	Introduction	1
	1.1 Description	1
	1.2 Problem Formulation	1
	1.3 Motivation	1
	1.4 Proposed Solution	2
	1.5 Scope of the project	2
2	REVIEW OF LITERATURE	4
3	SYSTEM ANALYSIS	6
	3.1 Functional Requirements	6
	3.2 Non-Functional Requirements	7
	3.3 Specific Requirements	7
	3.4 Use-Case Diagrams and description	8
4	ANALYSIS MODELING	9
	4.1 Activity Diagram	9
	4.2 Class Diagram	10
	4.3 Functional Modelling	11
	4.4 Timeline Chart	14
5	DESIGN	15
	5.1 Architectural Design	15
6	IMPLEMENTATION	17
7	TESTING	39
	7.1 Test cases	39
	7.2 Type of Testing used	47
8	RESULTS AND DISCUSSIONS	50
9	CONCLUSION	58

List of Figures

Fig. No.	Figure Caption	Page No.
3.4	Use case diagram for the proposed system	8
4.1	Activity diagram for the proposed system	9
4.2.1	Class diagram for the proposed system	10
4.3.1	Context diagram for the proposed system	11
4.3.2	Level 0 data flow diagram for the proposed system	12
4.3.3	Level 1 data flow diagram for the proposed system	13
4.4	Timeline Chart for the Proposed system	14
5.1	Architectural diagram for the proposed system	15
5.2	System GUI	16
7.2.2	Feedback Form for the proposed solution	48
8.1	Emotional analysis accuracy graph	50
8.2	Emotional analysis loss graph	50
8.3	CNN Model summary	51
8.4	Confusion matrix of Emotional analysis model	51
8.5	Classification report of Emotional Analysis	52
8.6	Prediction graph of Emotional Analysis Test Case 1	52
8.7	Prediction graph of Emotional Analysis Test Case 2	53
8.8	Feedback form for Test Case 1	56
8.9	Feedback form for Test Case 2	56
8.10	DynamoDB Table for Call Details	57

List of Tables

Table No.	Table Title	Page No.
4.2.2	Data dictionary table for the proposed system	11
8.1	Emotional Analysis Accuracy for all Test Cases	53
8.2	Speech-to-Text Transcription Accuracy for all Test Cases	54

List of Abbreviations

Sr. No.	Abbreviation	Expanded form
i	KPI	Key Performance Index
ii	STT	Speech to Text Transcription
iii	NLP	Natural Language Processing
iv	HMM	Hidden Markov Model
v	CNN	Convolution Neural Network

Chapter 1

Introduction

1.1 Description

In our increasingly industrialized and globalized world, a large number of companies include call centers in their structures and more than \$300 billion is spent annually on call centers around the world. For a customer, addressing the call center actually means addressing the company itself, and any negative experience on the part of the customer can lead to the rejection of company products and services. Hence, for the company, it is very important to ensure that a call centers function effectively and provides high quality service to its customers. Call centers collect a huge amount of data, and this provides a great opportunity for companies to use this information for the analysis of customer needs, desires, and intentions. Such data analysis can help improve the quality of customer service and lower the costs.

1.2 Problem Formulation

Call center optimization is an important part of customer relationship management which consists of people, processes, technology and strategies. Service quality of a call center is a result of comparison of actual service performance and customer expectations. Evaluating the service quality which is offered by customer service agent to customer is more difficult than evaluating the product quality. Reduce total call time, i.e. the average handling time, get real time emotion of the speaker, get real time solution.

1.3 Motivation

Call centers provide services for many types of sectors such as telecommunication, finance, transportation, health, automotive etc. Several studies have proposed various approaches and solutions for the problem of evaluating agent performance. Performance evaluation in call centers is generally performed through listening randomly selected calls from recorded calls, and evaluating the words one by one in the related conversation. obvious demand for automatic performance evaluation systems to reduce employee costs and to increase the time efficiency. Takeuchi has analyzed the recorded calls from a rental car reservation office with Trigger Segment Detection to find whether a customer has the intention of booking a car or not. This system is used to analyze the content of call center conversations and detect the

main issue addressed in the call. None of the previous existing methods cover all the ranges of call center analysis and give worthwhile insights on this problem.

1.4 Proposed Solution

Call Center Analytics is aimed at solving the above issue by enabling tapping into the content of conversations. We believe that just text-based content analysis approaches are highly sensitive to input quality and that conversational input is fundamentally different than text. Therefore, conversations should be treated differently. We first seek to diarize the speech as to separate the components of the customer and the agent. Thereafter, we use a mixture of emotional analysis as well as speech to text transcription while performing agent analysis through Natural Language Processing (NLP) technology which needs to be adapted to and be robust enough to deal with the conversational domain in order to achieve acceptable performance. Moreover, the level of analysis of conversation cannot be set to semantics only. It must consider the purpose of language in its context, i.e., pragmatics. Our approach to Call Center Analytics is based on Interaction Mining. Interaction Mining is a new research field aimed at extracting useful information from conversations. In contrast to Text Mining, Interaction Mining is more robust, tailored for the conversational domain, and slanted towards pragmatic and discourse analysis. All these details are stored in that will contain data about the current and previous calls to further aid the agents.

1.5 Scope of the project

1.5.1 Product Scope Description

Interaction analysis is performed to get -

1. Emotional analysis of the customer - This is shown in the form of a graph.
2. Speech to text transcription - The speech of the agent is converted into text, which is then used as the data for agent analysis.
3. Agent Analysis - The performance result of an agent will be produced in .xml format.
4. Database - Containing historical and current call data.

1.5.2 Project Deliverables

- Emotional Recognition - Gauge the emotional state of your customers by analyzing their voices for tell-tale variations in pitch or tone. In essence, the software determines the emotional tone behind a series of words, used to gain an understanding of the

attitudes, opinions and emotions expressed. Emotion analysis is done by converting the audio into spectrograms and predict the emotion from it.

- Speech to Text Transcription - Speech to text then allows us to process the speech to text which is stored as a .txt file. This file is then used to analyze the agents.
- Agent Analysis - The agent is analyzed using the speech to text transcription on the following criterion -
 1. Greeting score
 2. Number of slangs
 3. Number of repeated sentences
 4. Closing score
 5. Banned words
 6. Call length
 7. Whether the problem was solved or not

The above details will be stored in a .xml file for storage purposes and easy retrieval. Each .xml file will be then used to aid in further optimizing the agent performances.

- Hot topics with root cause analysis - Uncover and track the most frequent topics mentioned by customers, which aids in identifying trending customer satisfaction issues.

1.5.3 Project Exclusions

- Voice Activity Detection - It is a technique used in speech processing in which the presence or absence of human speech is detected. This will help in removing parts from the call that are unnecessary and improve the efficiency of the software.
- Talk over analysis - Highlight those moments when your customer and agent are talking simultaneously, an indicator of customer dissatisfaction. When neither of them is talking, though, it's time to look for agent knowledge gaps.

1.5.4 Product Acceptance Criteria

- To achieve significant cost savings.
- To increase revenue and customer satisfaction.
- To present a reliable monitoring system resulting in accurate performance measurements by analyzing all the incoming and outgoing calls.

Chapter 2

Review of Literature

Call centers provide services for many types of sectors such as telecommunication, finance, transportation, health, automotive etc. Several studies have proposed various approaches and solutions for the problem of evaluating agent performance. Performance evaluation in call centers is generally performed through listening randomly selected calls from recorded calls, and evaluating the words one by one in the related conversation. Obvious demand for automatic performance evaluation systems to reduce the employee costs and to increase the time efficiency. Takeuchi has analyzed the recorded calls from a rental car reservation office with Trigger Segment Detection to find whether a customer has the intention of booking a car or not. Mishne has proposed a call center monitoring system that uses text analytics and information retrieval methods. The system is used to analyze the content of call center conversations and detect the main issue addressed in the call. The project has presented speech analytics system adapted automatic speech recognition and text mining technologies [1]. Minnucci (2004) reports that the most required metrics by call center managers are indeed the qualitative ones topped by Call Quality (100%) and Customer Satisfaction (78%) [2].

However, these performance metrics are difficult to implement with the adequate level of accuracy. For instance, the Baird study (2004) points out that for Customer Satisfaction, accuracy can be *“negatively affected by insufficient number of administered surveys per agent resulting in not enough samples of individual agent’s work to constitute a representative sample. The result could be an unfair judgment of the agent’s performance and allocations of bonuses based more upon chance, good fortune than merit.”* Accuracy is defined as true indication and it depends on the actual level of performance attainment, especially with regard to statistical validity [3].

Current approaches to Call Centre Analytics are mostly based on Speech Analytics and Text Mining, which is essentially Search and Sentiment Analysis. Recorded speech is first indexed and searched against a set of negative terms and relevant topics. There are currently two main approaches for speech index: i) Phonetic Transcription and ii) Large Vocabulary Conversational Speech Recognition (LVCSR). Another common approach to the analysis of call center data is that of automatic call categorization through supervised machine learning (Gilman et al. 2004; Zweig et al. 2006; Takeuchi et al. 2009). These methods failed in providing satisfactory results even in very broad categories. The problem still lies on data sparseness and that huge amount of training data is necessary to achieve reasonable

discriminatory power. Getting huge training data is not an option also because training is highly influenced by domain specificity. Transferring trained models from a domain to another would be problematic [4].

Existing state-of-the-art SER methods apply deep Convolutional Neural Networks (CNNs) trained on spectral magnitude arrays of speech spectrograms. To achieve high accuracy, complex CNN structures must be trained on a very large number which is in the order of millions of labelled spectrograms. This method called “fresh training” is computationally intense, time consuming and requires large graphic processing units (GPUs). At present the availability of large datasets of emotionally labelled speech is limited. On the other hand, in many cases, close to the state-of-the-art results can be achieved using a much simpler process of transfer learning [5].

Chapter 3

System Analysis

3.1 Functional Requirements

3.1.1 Speech to text transcription

Database in use for testing and training - CMU Sphinx, which is a continuous-speech, speaker-independent recognition system making use of hidden Markov acoustic models (HMMs).

Method - Glob for selection of files and use of CMU Sphinx for speech recognition.

3.1.2 Emotional Analysis

Database in use for testing and training - RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song).

Method - Conversion to spectrogram, extraction of features and MFCC's, and use of convolution neural network.

3.1.3 Agent Analysis

Database in use for testing and training - Raw database received from STT.

Method: Using unique algorithms such as fuzzy, cosine similarity, sequence matching, etc.to provide insights on the KPI factors such as-

- Greeting
- Repeating sentences
- Closing
- If banned words are used
- Call length
- Whether the problem was solved or not

3.1.4 Speaker Diarization

Database in use for testing and training - Raw database

Method – Individually recognize the two different audio frequencies and separate the two voices.

3.1.5 Hot topic analysis

Database in use for testing and training - Raw database received from STT.

Method – Using NLP algorithms like cosine similarity, sequence matching, etc.

3.2 Non-Functional Requirements

3.2.1 Performance Requirements

The computer running the software must have a powerful CPU and GPU so that the data can be processed faster and there won't be a lag in the entire process of analyzing the agent's performance.

3.2.2 Security Requirements

As call center operations consists of entry portals for employees, knowledge management systems, and email etc. All of these constitute to be a probable vulnerability for an enterprise. The recorded calls are confidential and should not fall into the wrong hands.

We need to deal with call center security loopholes such as authentication, access control, authorization, identification and more by involving critical steps such as auditing, non-repudiation, accountability, etc.

3.2.3 Business Rules

At all times there must be confidentiality maintained regarding the information that is gathered by the software system and stored in .xml format. The data is then stored in the database containing all the personal information and information based on the call made. Only restricted or authorized personnel can view this data.

3.3 Specific Requirements

3.3.1 Hardware requirements

The hardware will require a high-performance CPU and GPU to perform the complex operations in the software model.

Minimum requirements -

CPU: Intel Core2 Quad Q6600 @ 2.4 GHz (or AMD Phenom 7950 Quad-Core, AMD Athlon II X4 620 equivalent)

RAM: 2 GB

3.3.2 Software requirements

Microsoft Windows 10/8.1/8/7

Python 3.6 or above

3.4 Use-Case Diagram and description

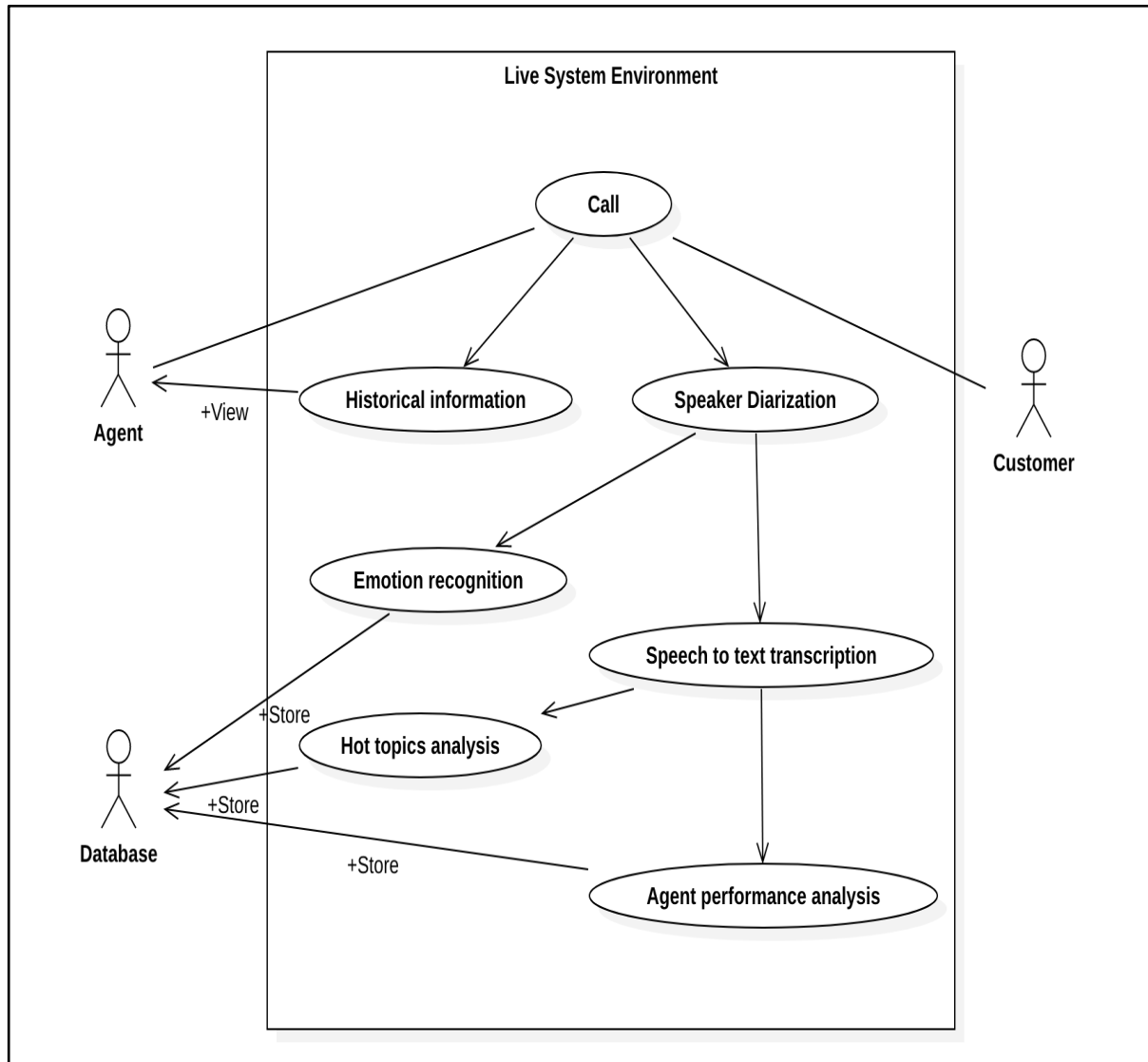


Fig 3.4 Use Case Diagram for the Proposed system

Description -

The live call audio will be given to the system. Customer Speech audio will be given for Emotional Analysis, where the detected emotions are shown with the help of a graph. Agent Speech audio will be given for Speech to Text Transcription, where the output is stored in STTOutput.txt file. The output from STT will be given as input for Hot Topic Analysis and Agent Analysis, where performance results of the agent will be generated. Historical information, if present, will be shown to the agent. The results from the modules will then be sent to a DynamoDB database for storage.

Chapter 4

Analysis Modeling

4.1 Activity Diagram

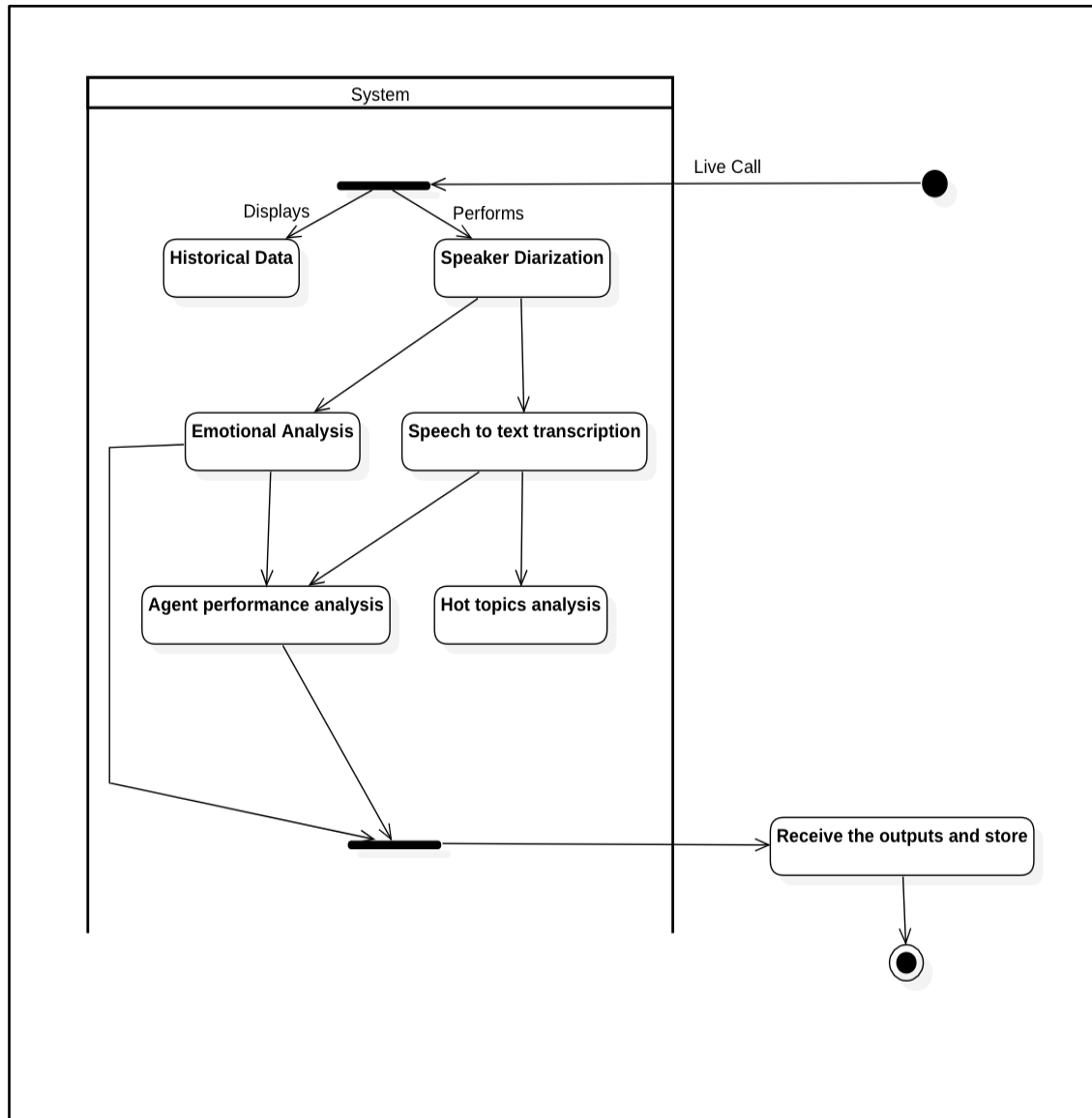


Fig 4.1 Activity diagram for the Proposed system

Description -

The main component is the system where all the processing takes place. The live call is fed directly into the system where it then displays the historical data pertaining to the caller. At the same instance, the system performs speaker diarization which is the input given to other modules such as emotional analysis, speech to text transcription, agent performance analysis and hot topic analysis. The records from all the modules are then stored in the DynamoDB database.

4.2 Class Diagram

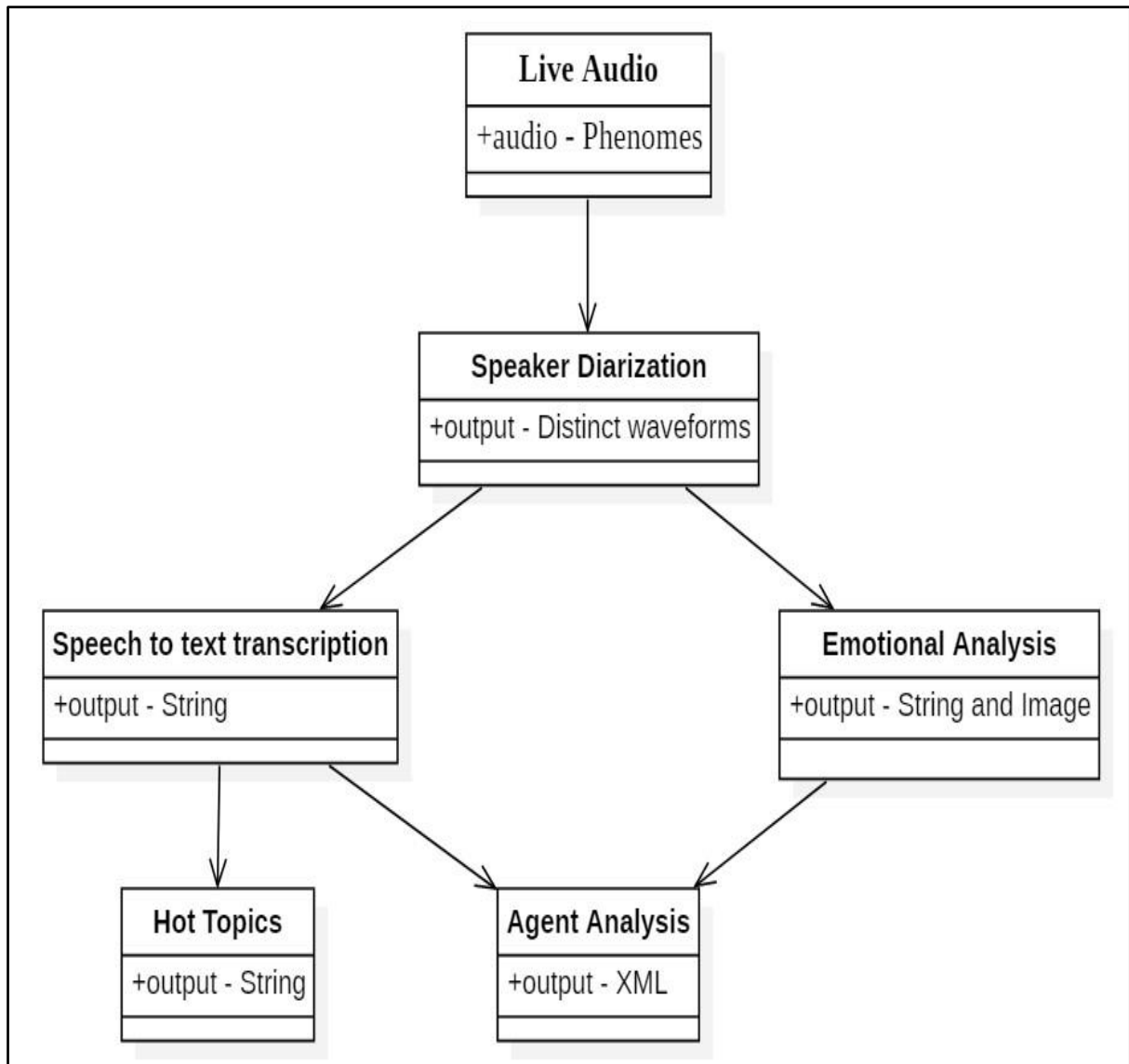


Fig 4.2.1 Class diagram for the Proposed system

Description -

The diagram above is the class diagram for the proposed system. The five classes are – Live Audio, Speaker Diarization, Speech to text transcription, Emotional analysis, Hot topics and Agent analysis. The outputs of each class are distinct to the respective class.

Table 4.2.1 Data Dictionary table for the Proposed system

Sr. No.	Name	Description	Data Type
1.	Live Audio	Input	Phonemes
2.	Speaker Diarization	Output	Distinct waveforms
3.	Emotional Analysis	Output	String and image
4.	Speech to Text	Output	String
5.	Agent Analysis	Output	XML
6.	Hot Topics	Output	String

The table above is the data dictionary for the class diagram. The 5 different modules are classified as either input or output with their data types shown in the table.

4.3 Functional Modeling

Context -

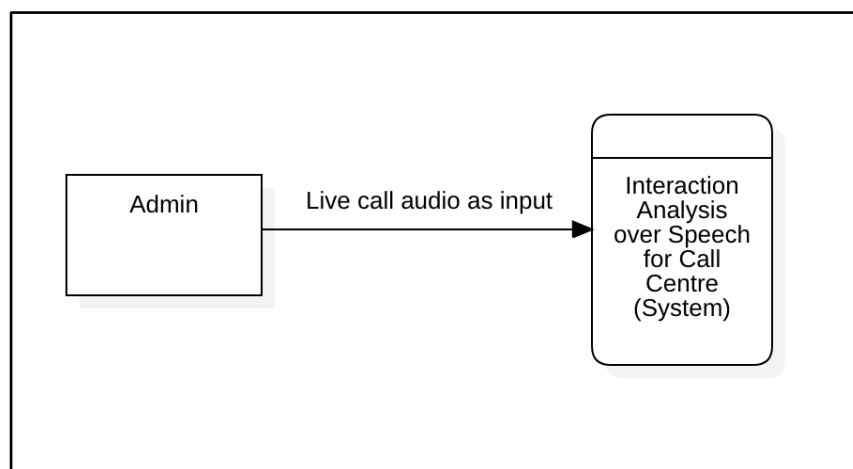


Fig 4.3.1 Context diagram for the Proposed system

Description –

The context diagram states the only relation between the two entities, the admin here refers to the employee at the call center. The admin feeds the live audio call as input to our system.

Level 0 -

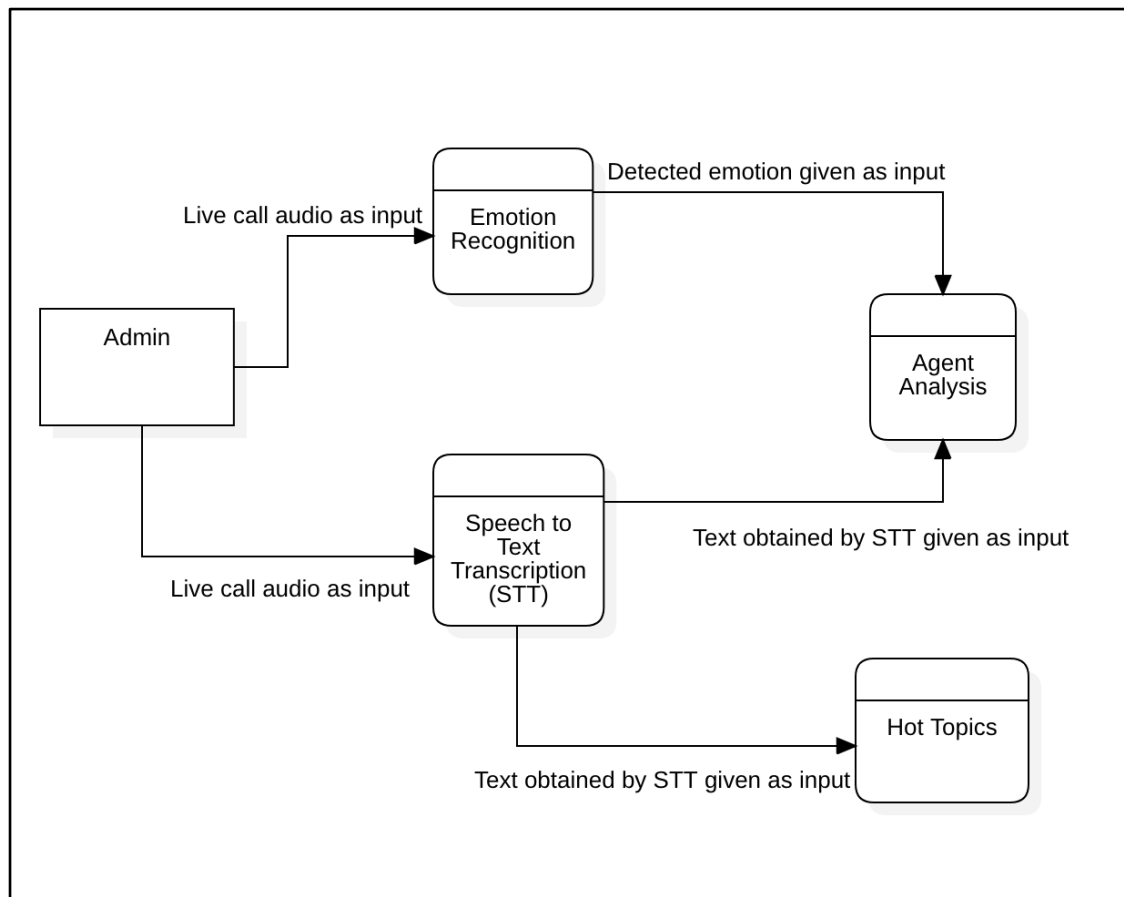


Fig 4.3.2 Level 0 Data flow diagram for the Proposed system

Description –

The data flow is initiated by the admin, the admin here refers to the employee at the call center. The admin feeds the live call as input to the emotion recognition module, where, the detected emotions are given as input to the agent analysis. The agent analysis also receives the input, in the form of text, from the speech to text transcription module. The text output from the speech to text transcription module is also given to the hot topic's module.

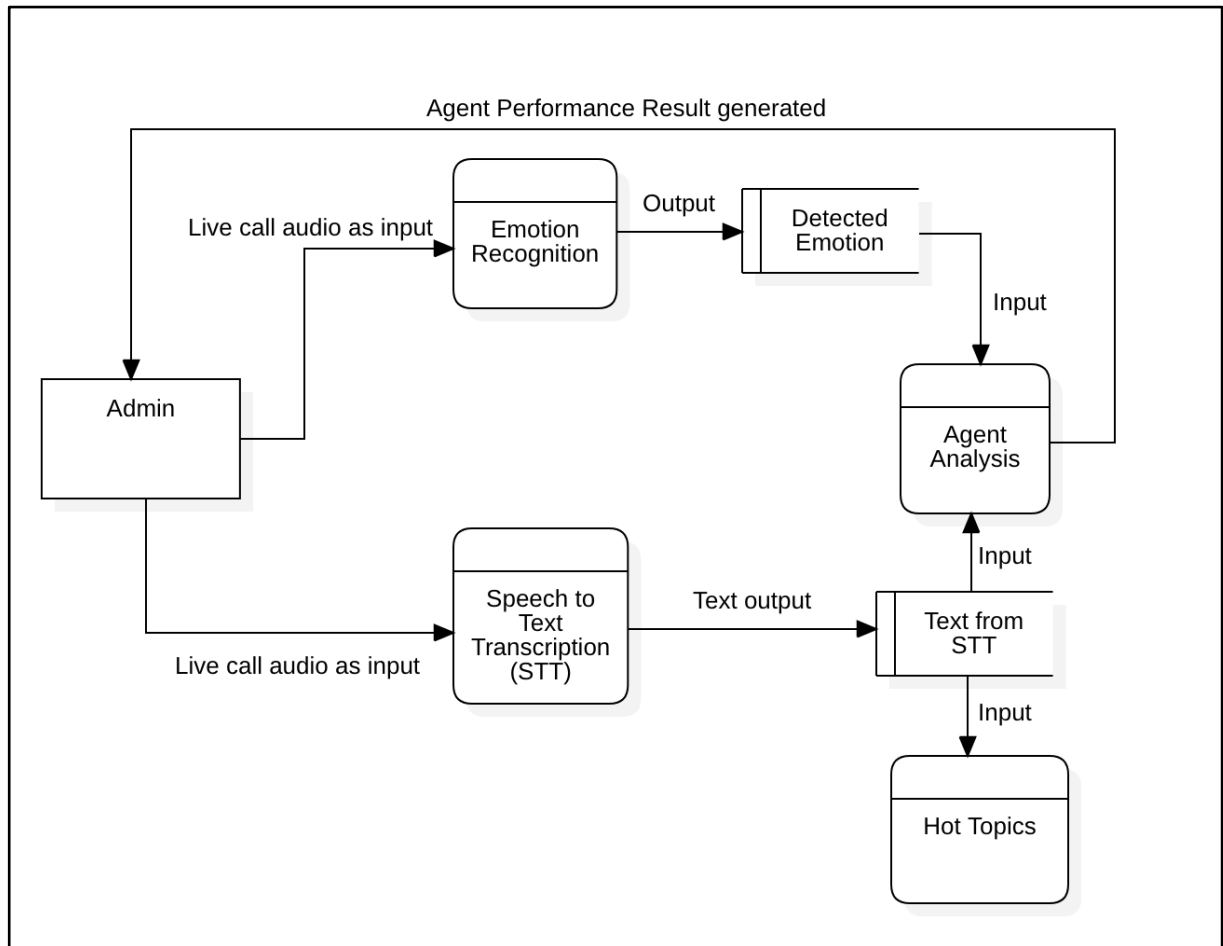


Fig 4.3.3 Level 1 Data flow diagram for the proposed system

Description –

The level 1 diagram is more detailed. The data flow is initiated by the admin, the admin here refers to the employee at the call center. The admin feeds the live call as input to the emotion recognition module, where, the detected emotions are given as input to the agent analysis. The agent analysis also receives the input, in the form of text, from the speech to text transcription module. The text output from the speech to text transcription module is also given to the hot topic's module. The agent performance result generated is given back to the admin and is stored for further analysis or record keeping.

4.4 Timeline Chart

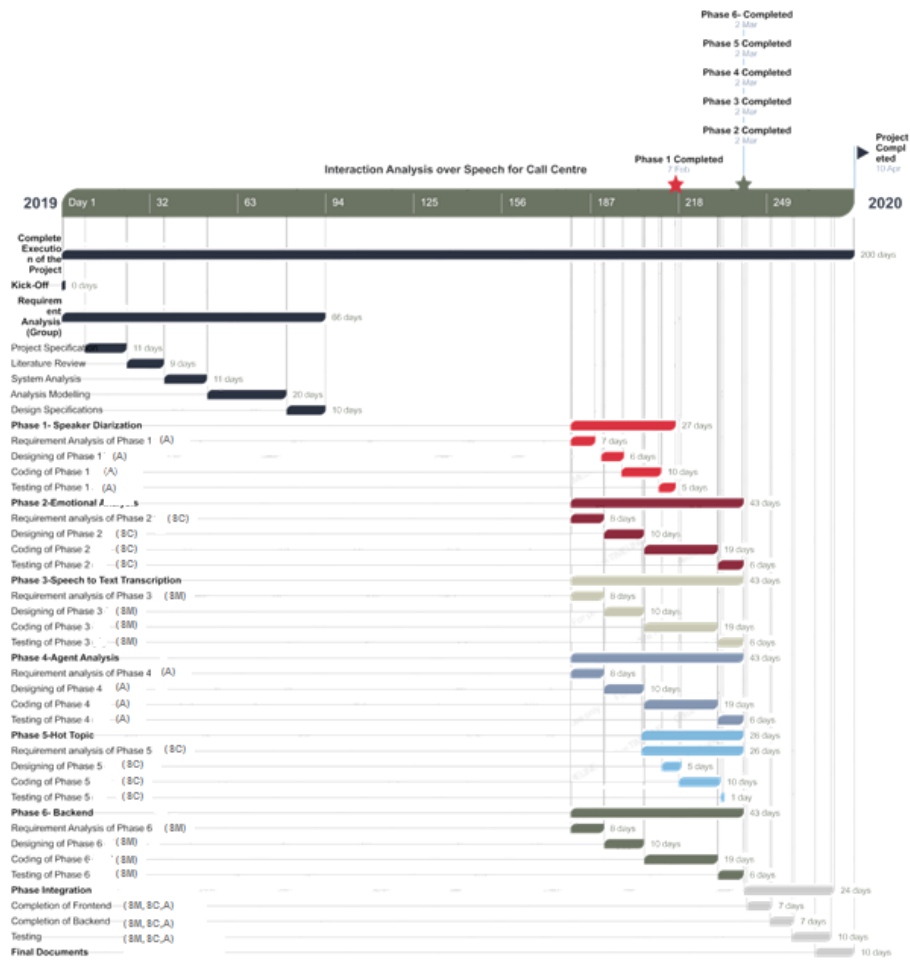


Fig 4.4 Timeline Chart

Key Legend:

A: Aarohi Mahajan

SC: Sneha Chitale

SM: Shivam Mehta

Chapter 5

Design

5.1 Architectural Design

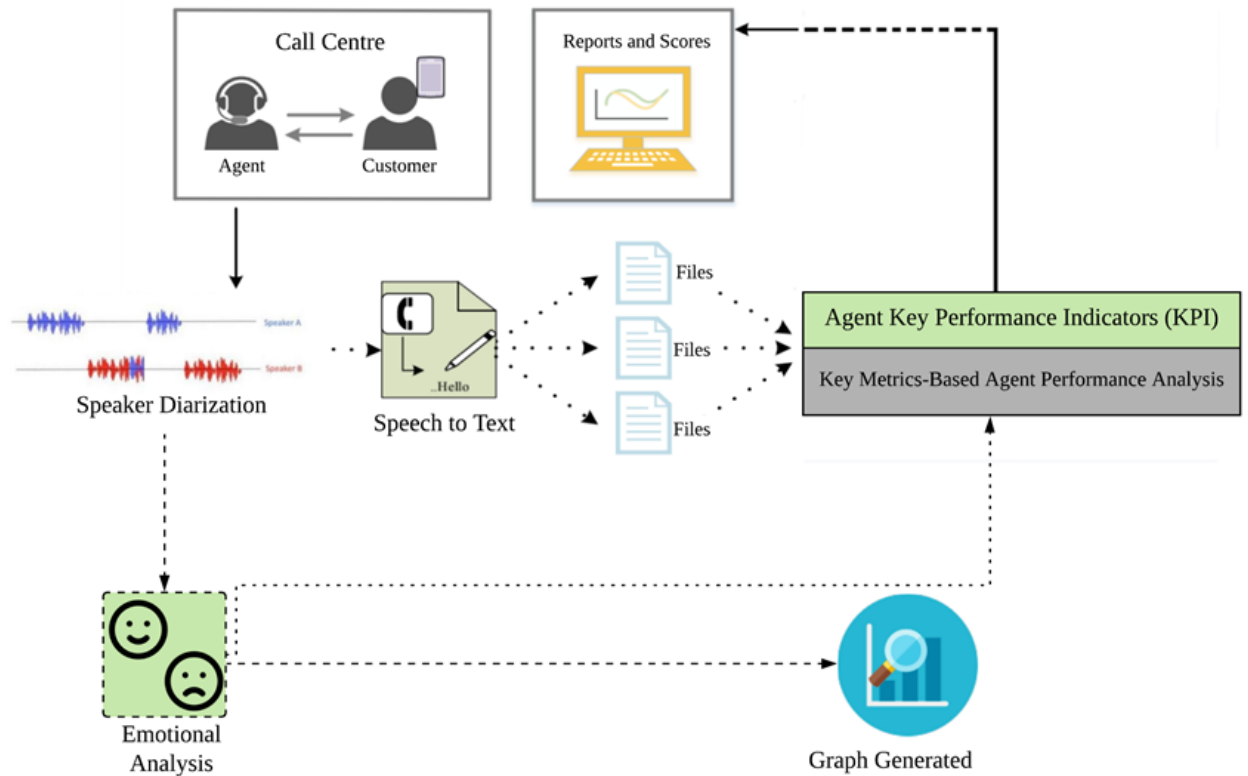


Fig 5.1 Architectural diagram for the Proposed system

The system will process the live call. Speaker diarization will be performed on the call to break down the call into two speaker components. Customer Speech audio and Agent Speech audio will be given for Emotional Analysis, where the detected emotions are shown with the help of a graph. Audios will then further be given for Speech to Text Transcription, where the output is stored in a text file. The output from the text file will be given as input for Agent Analysis and Hot Topic Analysis, where performance results of the agent will be generated and stored in the database.

5.2 User Interface Design

Main system –

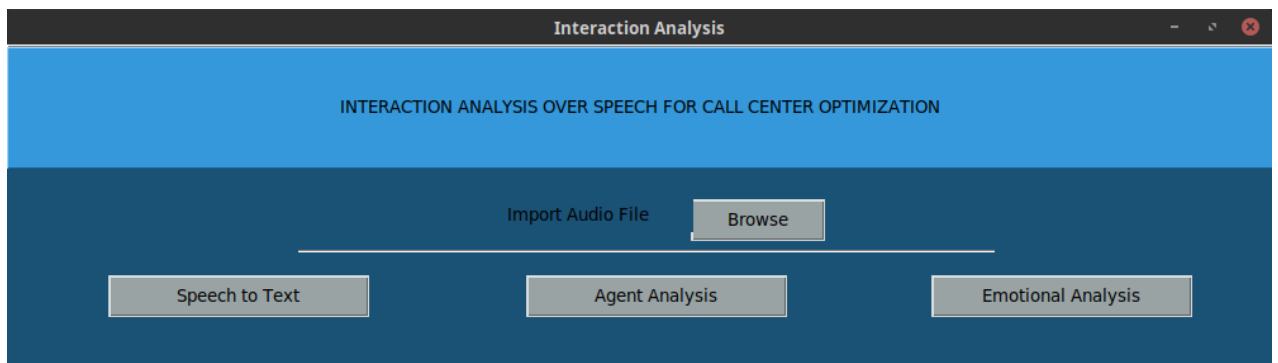


Fig 5.2 System GUI

Chapter 6

Implementation

6.1 Algorithms / Method Used

6.1.1 Convolutional Neural Network

- In neural networks, Convolutional NeuralNetwork (ConvNets or CNNs) is one of the main categories to do images recognition, images classifications. Objects detections, recognition faces etc., are some of the areas where CNNs are widely used. Computers sees an input image as array of pixels and it depends on the image resolution. Based on the image resolution, it will see $h \times w \times d$ (h = Height, w = Width, d = Dimension). E.g., An image of $6 \times 6 \times 3$ array of matrix of RGB (3 refers to RGB values) and an image of $4 \times 4 \times 1$ array of matrix of grayscale image.
- Technically, deep learning CNN models to train and test, each input image will pass it through a series of convolution layers with filters (Kernels), Pooling, fully connected layers (FC) and apply Softmax function to classify an object with probabilistic values between 0 and 1.
- CNN is used to recognise emotion from the audio recording. CNN model for emotional recognition is constructed with 7 layers - 6 Conv1D layers followed by a Dense layer.

6.1.2 Hidden Markov Model

- The Hidden Markov Model is a finite set of states, each of which is associated with a (generally multidimensional) probability distribution. Transitions among the states are governed by a set of probabilities called transition probabilities. In a particular state an outcome or observation can be generated, according to the associated probability distribution. It is only the outcome, not the state visible to an external observer and therefore states are “hidden” to the outside; hence the name Hidden Markov Model.
- It's a generic model that describes a black-box communication channel. In this model process is described as a sequence of states which change each other with a certain probability. This model is intended to describe any sequential process like speech. HMMs have been proven to be really practical for speech decoding.

- Speech to Text Transcription (STT) is done with the help of Sphinx which is a continuous-speech, speaker-independent recognition system making use of hidden Markov acoustic models (HMMs) and an n-gram statistical language model.

6.1.3 Porter Stemmer

- For grammatical reasons, documents are going to use different forms of a word, such as *organize*, *organizes*, and *organizing*. Additionally, there are families of derivationally related words with similar meanings, such as *democracy*, *democratic*, and *democratization*. In many situations, it seems as if it would be useful for a search for one of these words to return documents that contain another word in the set. The goal of both stemming is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form.
- Porter Stemmer focuses on removing suffixes by automatic means is an operation which is especially useful in the field of information retrieval. In a typical IR environment, one has a collection of documents, each described by the words in the document title and possibly by words in the document abstract. Ignoring the issue of precisely where the words originate, we can say that a document is represented by a vector of words, or *terms*. Terms with a common stem will usually have similar meanings, for example:

CONNECT
CONNECTED
CONNECTING
CONNECTION
CONNECTIONS

- Frequently, the performance of an IR system will be improved if term groups such as this are conflated into a single term. This may be done by removal of the various suffixes -ED, -ING, -ION, IONS to leave the single term CONNECT. In addition, the suffix stripping process will reduce the total number of terms in the IR system, and hence reduce the size and complexity of the data in the system, which is always advantageous.

6.1.4 Cosine Similarity

- Cosine similarity is the cosine of the angle between two n-dimensional vectors in an n-dimensional space. It is the dot product of the two vectors divided by the product of the two vectors' lengths (or magnitudes).

- Cosine similarity is computed using the following formula:

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n A_i^2} \times \sqrt{\sum_{i=1}^n B_i^2}}$$

- Values range between -1 and 1, where -1 is perfectly dissimilar and 1 is perfectly similar. The library contains both procedures and functions to calculate similarity between sets of data. The function is best used when calculating the similarity between small numbers of sets. The procedures parallelize the computation and are therefore more appropriate for computing similarities on bigger datasets.
- Cosine similarity is used in Agent Analysis in order to find if any sentences are repeated by the agent during a call.

6.1.5 Tokenization

- Given a character sequence and a defined document unit, tokenization is the task of chopping it up into pieces, called tokens, perhaps at the same time throwing away certain characters, such as punctuation. Here is an example of tokenization:

Input: Friends, Romans, Countrymen, lend me your ears;

Output:

Friends	Romans	Countrymen	lend	me	your	ears
---------	--------	------------	------	----	------	------

- A *token* is an instance of a sequence of characters in some particular document that are grouped together as a useful semantic unit for processing.
- A *type* is the class of all tokens containing the same character sequence.
- A *term* is a (perhaps normalized) type that is included in the IR system's dictionary.
- Tokenization is used in STT to tokenize the transcribed output which helps in calculating the accuracy of Speech to Text Transcription and it is also used in Hot Topic Analysis.

6.1.6 Removal of Stop Words

- A stop word is a commonly used word (such as “the”, “a”, “an”, “in”) that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query.
- In computing, stop words are words which are filtered out before processing of natural language data (text) Stop words are generally the most common words in a language; there is no single universal list of stop words used by all natural language processing tools, and indeed not all tools even use such a list. Some tools avoid removing stop words to support phrase search.
- Removal of stop words is therefore used in Hot Topic Analysis which is a way to hide particular words or phrases from appearing in the analysis.

6.2 Working of the project

```
import keras
import os
import numpy as np
import librosa
import matplotlib.pyplot as plt
import glob
import os.path
import speech_recognition as sr
import pydub
from pydub import AudioSegment
from pydub.utils import make_chunks
import nltk
import datetime
from difflib import SequenceMatcher
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize, sent_tokenize
from collections import Counter
from tkinter import filedialog
from tkinter.filedialog import askopenfilename
import sys
import tkinter as tk
import tkinter.ttk as ttk
from scipy.io import wavfile
import wave
```

```

from dicttoxml import dicttoxml
import re
import subprocess, sys
import spacey

try:
    import Tkinter as tk
except ImportError:
    import tkinter as tk

try:
    import ttk
    py3 = False
except ImportError:
    import tkinter.ttk as ttk
    py3 = True

import Login_support

def vp_start_gui():
    """Starting point when module is the main routine."""
    global val, w, root
    root = tk.Tk()
    top = Toplevel (root)
    Login_support.init(root, top)
    root.mainloop()

w = None

def create_Toplevel1(root, *args, **kwargs):
    """Starting point when module is imported by another program."""
    global w, w_win, rt
    rt = root
    w = tk.Toplevel (root)
    top = Toplevel1 (w)
    Login_support.init(w, top, *args, **kwargs)

```

```

    return (w, top)

def destroy_Toplevel1():
    global w
    w.destroy()
    w = None

class Toplevel1:
    def __init__(self, top=None):
        """This class configures and populates the toplevel window.
            top is the toplevel containing window."""
        _bgcolor = '#d9d9d9' # X11 color: 'gray85'
        _fgcolor = '#000000' # X11 color: 'black'
        _compcolor = '#d9d9d9' # X11 color: 'gray85'
        _ana1color = '#d9d9d9' # X11 color: 'gray85'
        _ana2color = '#ecec' # Closest X11 color: 'gray92'

top.geometry("413x674+757+150")
top.title("IAOSFCC")
top.configure(background="#A0CFEC")
top.configure(highlightbackground="#d9d9d9")
top.configure(highlightcolor="black")

self.AgentID = tk.Entry(top)
self.AgentID.place(relx=0.121, rely=0.445,height=34, relwidth=0.76)
self.AgentID.configure(background="white")
self.AgentID.configure(disabledforeground="#a3a3a3")
self.AgentID.configure(font="TkFixedFont")
self.AgentID.configure(foreground="#000000")
self.AgentID.configure(highlightbackground="#d9d9d9")
self.AgentID.configure(highlightcolor="black")
self.AgentID.configure(insertbackground="black")
self.AgentID.configure(selectbackground="#c4c4c4")
self.AgentID.configure(selectforeground="black")

```

```

self.AgentPassword = tk.Entry(top)
self.AgentPassword.place(relx=0.121, rely=0.579, height=34
                        , relwidth=0.76)
self.AgentPassword.configure(background="white")
self.AgentPassword.configure(disabledforeground="#a3a3a3")
self.AgentPassword.configure(font="TkFixedFont")
self.AgentPassword.configure(foreground="#000000")
self.AgentPassword.configure(highlightbackground="#d9d9d9")
self.AgentPassword.configure(highlightcolor="black")
self.AgentPassword.configure(insertbackground="black")
self.AgentPassword.configure(selectbackground="#c4c4c4")
self.AgentPassword.configure(selectforeground="black")

self.Login = tk.Button(top)
self.Login.place(relx=0.278, rely=0.757, height=43, width=166)
self.Login.configure(activebackground="#ececce")
self.Login.configure(activeforeground="#000000")
self.Login.configure(background="#d9d9d9")
self.Login.configure(disabledforeground="#a3a3a3")
self.Login.configure(font="-family {Segoe UI} -size 12")
self.Login.configure(foreground="#000000")
self.Login.configure(highlightbackground="#d9d9d9")
self.Login.configure(highlightcolor="black")
self.Login.configure(pady="0")
self.Login.configure(relief="ridge")
self.Login.configure(text="Log In")

self.Label1 = tk.Label(top)
self.Label1.place(relx=0.121, rely=0.371, height=46, width=75)
self.Label1.configure(activebackground="#f9f9f9")
self.Label1.configure(activeforeground="black")
self.Label1.configure(background="#A0CFEC")
self.Label1.configure(disabledforeground="#a3a3a3")
self.Label1.configure(font="-family {Segoe UI} -size 12")
self.Label1.configure(foreground="#000000")

```

```

self.Label1.configure(highlightbackground="#d9d9d9")
self.Label1.configure(highlightcolor="black")
self.Label1.configure(text="Agent ID")

self.Label1_3 = tk.Label(top)
self.Label1_3.place(relx=0.109, rely=0.519, height=36, width=95)
self.Label1_3.configure(activebackground="#f9f9f9")
self.Label1_3.configure(activeforeground="black")
self.Label1_3.configure(background="#A0CFEC")
self.Label1_3.configure(disabledforeground="#a3a3a3")
self.Label1_3.configure(font="-family {Segoe UI} -size 13")
self.Label1_3.configure(foreground="#000000")
self.Label1_3.configure(highlightbackground="#d9d9d9")
self.Label1_3.configure(highlightcolor="black")
self.Label1_3.configure(text="Password")

self.Label2 = tk.Label(top)
self.Label2.place(relx=0.254, rely=0.645, height=26, width=190)
self.Label2.configure(activebackground="#f9f9f9")
self.Label2.configure(activeforeground="black")
self.Label2.configure(background="#A0CFEC")
self.Label2.configure(disabledforeground="#a3a3a3")
self.Label2.configure(foreground="#000000")
self.Label2.configure(highlightbackground="#d9d9d9")
self.Label2.configure(highlightcolor="black")
self.Label2.configure(text="Having trouble logging in?")

import main_support

def vp_start_gui():
    """Starting point when module is the main routine."""
    global val, w, root
    root = tk.Tk()
    top = Toplevel1 (root)

```

```

main_support.init(root, top)
root.mainloop()

w = None

def create_Toplevel1(root, *args, **kwargs):
    """Starting point when module is imported by another program."""
    global w, w_win, rt
    rt = root
    w = tk.Toplevel (root)
    top = Toplevel1 (w)
main_support.init(w, top, *args, **kwargs)
    return (w, top)

def destroy_Toplevel1():
    global w
w.destroy()
    w = None

class Toplevel1:
    def __init__(self, top=None):
        """This class configures and populates the toplevel window.
        top is the toplevel containing window."""
        _bgcolor = '#d9d9d9' # X11 color: 'gray85'
        _fgcolor = '#000000' # X11 color: 'black'
        _compcolor = '#d9d9d9' # X11 color: 'gray85'
        _ana1color = '#d9d9d9' # X11 color: 'gray85'
        _ana2color = '#ecec' # Closest X11 color: 'gray92'
self.style = ttk.Style()
        if sys.platform == "win32":
self.style.theme_use('winnative')
self.style.configure('.',background=_bgcolor)
self.style.configure('.',foreground=_fgcolor)
self.style.configure('.',font="TkDefaultFont")
self.style.map('.',background=
        [('selected', _compcolor), ('active',_ana2color)])

```

```

top.geometry("770x635+412+185")
top.title("IAOSFCC")
top.configure(background="#A0CFEC")
top.configure(highlightbackground="#d9d9d9")
top.configure(highlightcolor="black")

self.Label1 = tk.Label(top)
self.Label1.place(relx=0.065, rely=0.189, height=36, width=243)
self.Label1.configure(activebackground="#f9f9f9")
self.Label1.configure(activeforeground="black")
self.Label1.configure(background="#A0CFEC")
self.Label1.configure(disabledforeground="#a3a3a3")
self.Label1.configure(font="-family {Segoe UI} -size 13")
self.Label1.configure(foreground="#000000")
self.Label1.configure(highlightbackground="#d9d9d9")
self.Label1.configure(highlightcolor="black")
self.Label1.configure(text="Enter the phone number")

self.Entry1 = tk.Entry(top)
self.Entry1.place(relx=0.065, rely=0.283, height=34, relwidth=0.317)
self.Entry1.configure(background="white")
self.Entry1.configure(disabledforeground="#a3a3a3")
self.Entry1.configure(font="TkFixedFont")
self.Entry1.configure(foreground="#000000")
self.Entry1.configure(highlightbackground="#d9d9d9")
self.Entry1.configure(highlightcolor="black")
self.Entry1.configure(insertbackground="black")
self.Entry1.configure(selectbackground="#c4c4c4")
self.Entry1.configure(selectforeground="black")

self.Button1 = tk.Button(top)
self.Button1.place(relx=0.117, rely=0.394, height=33, width=166)
self.Button1.configure(activebackground="#ecec"

```



```

self.Button1.configure(background="#d9d9d9")
self.Button1.configure(disabledforeground="#a3a3a3")
self.Button1.configure(foreground="#000000")
self.Button1.configure(highlightbackground="#d9d9d9")
self.Button1.configure(highlightcolor="black")
self.Button1.configure(pady="0")
self.Button1.configure(text="Retrieve")

self.TSeparator1 = ttk.Separator(top)
self.TSeparator1.place(relx=0.455, rely=0.142, relheight=0.457)
self.TSeparator1.configure(orient="vertical")

self.TSeparator2 = ttk.Separator(top)
self.TSeparator2.place(relx=0.052, rely=0.63, relwidth=0.896)

self.menubar = tk.Menu(top,font="TkMenuFont",bg=_bgcolor,fg=_fgcolor)
top.configure(menu = self.menubar)

self.Frame1 = tk.Frame(top)
self.Frame1.place(relx=0.0, rely=0.0, relheight=0.134, relwidth=1.006)
self.Frame1.configure(relief='groove')
self.Frame1.configure(borderwidth="2")
self.Frame1.configure(relief="groove")
self.Frame1.configure(background="#d9d9d9")
self.Frame1.configure(highlightbackground="#d9d9d9")
self.Frame1.configure(highlightcolor="black")
self.Frame1.configure(width=775)

self.Label2 = tk.Label(self.Frame1)
self.Label2.place(relx=0.142, rely=0.353, height=26, width=571)
self.Label2.configure(activebackground="#f9f9f9")
self.Label2.configure(activeforeground="black")
self.Label2.configure(background="#d9d9d9")
self.Label2.configure(disabledforeground="#a3a3a3")
self.Label2.configure(foreground="#000000")

```

```
self.Label2.configure(highlightbackground="#d9d9d9")
self.Label2.configure(highlightcolor="black")
self.Label2.configure(text="INTERACTION ANALYSIS OVER SPEECH FOR CALL
CENTRE")
```

```
self.Frame2 = tk.Frame(top)
self.Frame2.place(relx=0.0, rely=0.63, relheight=0.37, relwidth=1.006)
self.Frame2.configure(relief='groove')
self.Frame2.configure(borderwidth="2")
self.Frame2.configure(relief="groove")
self.Frame2.configure(background="#d9d9d9")
self.Frame2.configure(highlightbackground="#d9d9d9")
self.Frame2.configure(highlightcolor="black")
self.Frame2.configure(width=775)
```

```
self.Button3 = tk.Button(self.Frame2)
self.Button3.place(relx=0.697, rely=0.596, height=33, width=196)
self.Button3.configure(activebackground="#ecec")
self.Button3.configure(activeforeground="#000000")
self.Button3.configure(background="#A0CFEC")
self.Button3.configure(disabledforeground="#a3a3a3")
self.Button3.configure(font="-family {Segoe UI} -size 11")
self.Button3.configure(foreground="#000000")
self.Button3.configure(highlightbackground="#d9d9d9")
self.Button3.configure(highlightcolor="black")
self.Button3.configure(pady="0")
self.Button3.configure(text="Agent Analysis")
```

```
self.Button4 = tk.Button(self.Frame2)
self.Button4.place(relx=0.374, rely=0.596, height=33, width=196)
self.Button4.configure(activebackground="#ecec")
self.Button4.configure(activeforeground="#000000")
self.Button4.configure(background="#A0CFEC")
self.Button4.configure(disabledforeground="#a3a3a3")
self.Button4.configure(font="-family {Segoe UI} -size 11")
```

```

self.Button4.configure(foreground="#000000")
self.Button4.configure(highlightbackground="#d9d9d9")
self.Button4.configure(highlightcolor="black")
self.Button4.configure(pady="0")
self.Button4.configure(text="Speech to text")

self.Button5 = tk.Button(self.Frame2)
self.Button5.place(relx=0.039, rely=0.596, height=33, width=196)
self.Button5.configure(activebackground="#ececce")
self.Button5.configure(activeforeground="#000000")
self.Button5.configure(background="#A0CFEC")
self.Button5.configure(disabledforeground="#a3a3a3")
self.Button5.configure(font="-family {Segoe UI} -size 11")
self.Button5.configure(foreground="#000000")
self.Button5.configure(highlightbackground="#d9d9d9")
self.Button5.configure(highlightcolor="black")
self.Button5.configure(pady="0")
self.Button5.configure(text="Emotional Analysis")

self.Button2 = tk.Button(top)
self.Button2.place(relx=0.351, rely=0.693, height=33, width=226)
self.Button2.configure(activebackground="#ececce")
self.Button2.configure(activeforeground="#000000")
self.Button2.configure(background="#A0CFEC")
self.Button2.configure(disabledforeground="#a3a3a3")
self.Button2.configure(font="-family {Segoe UI} -size 11")
self.Button2.configure(foreground="#000000")
self.Button2.configure(highlightbackground="#d9d9d9")
self.Button2.configure(highlightcolor="black")
self.Button2.configure(pady="0")
self.Button2.configure(text="Select call")

self.Scrolledtext1 = ScrolledText(top)
self.Scrolledtext1.place(relx=0.481, rely=0.22, relheight=0.384
    , relwidth=0.487)

```

```

self.Scrolledtext1.configure(background="white")
self.Scrolledtext1.configure(font="TkTextFont")
self.Scrolledtext1.configure(foreground="black")
self.Scrolledtext1.configure(highlightbackground="#d9d9d9")
self.Scrolledtext1.configure(highlightcolor="black")
self.Scrolledtext1.configure(insertbackground="black")
self.Scrolledtext1.configure(insertborderwidth="3")
self.Scrolledtext1.configure(selectbackground="#c4c4c4")
self.Scrolledtext1.configure(selectforeground="black")
self.Scrolledtext1.configure(width=10)
self.Scrolledtext1.configure(wrap="none")

self.Label3 = tk.Label(top)
self.Label3.place(relx=0.481, rely=0.157, height=29, width=151)
self.Label3.configure(activebackground="#f9f9f9")
self.Label3.configure(activeforeground="black")
self.Label3.configure(background="#A0CFEC")
self.Label3.configure(disabledforeground="#a3a3a3")
self.Label3.configure(font="-family {Segoe UI} -size 10")
self.Label3.configure(foreground="#000000")
self.Label3.configure(highlightbackground="#d9d9d9")
self.Label3.configure(highlightcolor="black")
self.Label3.configure(text="Customer details -")

```

```

class AutoScroll(object):
    "Configure the scrollbars for a widget."

    def __init__(self, master):
        try:
            vsb = ttk.Scrollbar(master, orient='vertical', command=self.yview)
            except:
                pass
            hsb = ttk.Scrollbar(master, orient='horizontal', command=self.xview)

```

```

        #self.configure(yscrollcommand=_autoscroll(vsb),
        # xscrollcommand=_autoscroll(hsb))
    try:
self.configure(yscrollcommand=self._autoscroll(vsb))
        except:
            pass
self.configure(xscrollcommand=self._autoscroll(hsb))

self.grid(column=0, row=0, sticky='nsew')
    try:
vsb.grid(column=1, row=0, sticky='ns')
        except:
            pass
hsb.grid(column=0, row=1, sticky='ew')

master.grid_columnconfigure(0, weight=1)
master.grid_rowconfigure(0, weight=1)

if py3:
    methods = tk.Pack.__dict__.keys() | tk.Grid.__dict__.keys() \
        | tk.Place.__dict__.keys()
else:
    methods = tk.Pack.__dict__.keys() + tk.Grid.__dict__.keys() \
        + tk.Place.__dict__.keys()

    for meth in methods:
        if meth[0] != '_' and meth not in ('config', 'configure'):
setattr(self, meth, getattr(master, meth))

    @staticmethod
    def _autoscroll(sbar):
        """Hide and show scrollbar as needed."""
        def wrapped(first, last):
            first, last = float(first), float(last)
            if first <= 0 and last >= 1:

```

```

sbar.grid_remove()
    else:
sbar.grid()
sbar.set(first, last)
    return wrapped

def __str__(self):
    return str(self.master)

def _create_container(func):
    """Creates a tkk Frame with a given master, and use this new frame to
    place the scrollbars and the widget."""
    def wrapped(cls, master, **kw):
        container = tkk.Frame(master)
        container.bind('<Enter>', lambda e: _bound_to_mousewheel(e, container))
        container.bind('<Leave>', lambda e: _unbound_to_mousewheel(e, container))
        return func(cls, container, **kw)
    return wrapped

class ScrolledText(AutoScroll, tk.Text):
    """A standard Tkinter Text widget with scrollbars that will
    automatically show/hide as needed."""
    @_create_container
    def __init__(self, master, **kw):
        tk.Text.__init__(self, master, **kw)
        AutoScroll.__init__(self, master)

import platform
def _bound_to_mousewheel(event, widget):
    child = widget.winfo_children()[0]
    if platform.system() == 'Windows' or platform.system() == 'Darwin':
        child.bind_all('<MouseWheel>', lambda e: _on_mousewheel(e, child))
        child.bind_all('<Shift-MouseWheel>', lambda e: _on_shiftmouse(e, child))
    else:
        child.bind_all('<Button-4>', lambda e: _on_mousewheel(e, child))

```

```

child.bind_all('<Button-5>', lambda e: _on_mousewheel(e, child))
child.bind_all('<Shift-Button-4>', lambda e: _on_shiftmouse(e, child))
child.bind_all('<Shift-Button-5>', lambda e: _on_shiftmouse(e, child))

def _unbound_to_mousewheel(event, widget):
    if platform.system() == 'Windows' or platform.system() == 'Darwin':
        widget.unbind_all('<MouseWheel>')
        widget.unbind_all('<Shift-MouseWheel>')
    else:
        widget.unbind_all('<Button-4>')
        widget.unbind_all('<Button-5>')
        widget.unbind_all('<Shift-Button-4>')
        widget.unbind_all('<Shift-Button-5>')

def _on_mousewheel(event, widget):
    if platform.system() == 'Windows':
        widget.yview_scroll(-1*int(event.delta/120), 'units')
    elif platform.system() == 'Darwin':
        widget.yview_scroll(-1*int(event.delta), 'units')
    else:
        if event.num == 4:
            widget.yview_scroll(-1, 'units')
        elif event.num == 5:
            widget.yview_scroll(1, 'units')

def _on_shiftmouse(event, widget):
    if platform.system() == 'Windows':
        widget.xview_scroll(-1*int(event.delta/120), 'units')
    elif platform.system() == 'Darwin':
        widget.xview_scroll(-1*int(event.delta), 'units')
    else:
        if event.num == 4:
            widget.xview_scroll(-1, 'units')
        elif event.num == 5:
            widget.xview_scroll(1, 'units')

```

```

def STT(self):
    global r
    print (r)
    audioloc=os.path.dirname(r)
    print(audioloc)
    filename = (os.path.splitext(os.path.basename(r))[0])
    filenames=[]
    filenames = sorted(glob.glob(audioloc+"/"+filename+"*.wav"))
    os.chdir(audioloc)
    open('STToutput.txt', 'w').close()
    l=len(filenames)
    print(filenames)
    for i in range(0,l):
    wave_file=filenames[i]
    print("Processing { }...".format(wave_file))
    frequency_sampling, audio_signal = wavfile.read(wave_file)
    print('Signal shape:', audio_signal.shape)
    print('Signal Datatype:', audio_signal.dtype)
    print('Signal duration:', round(audio_signal.shape[0] /
        float(frequency_sampling), 2), 'seconds')

    audio_signal = audio_signal / np.power(2, 15)
    audio_signal = audio_signal [:100]

    time_axis = 1000 * np.arange(0, len(audio_signal), 1) / float(frequency_sampling)

    plt.plot(time_axis, audio_signal, color='blue')
    plt.xlabel('Time (milliseconds)')
    plt.ylabel('Amplitude')
    plt.title('Input audio signal')
    plt.show()

    AUDIO_FILE = (wave_file)
    t = sr.Recognizer()

```



```

with sr.AudioFile(AUDIO_FILE) as source:
    audio = t.record(source)
    value = t.recognize_sphinx(audio)
    try:
print("The audio file contains: " +value)
        except sr.UnknownValueError:
print("Speech Recognition could not understand audio")
            except sr.RequestError as e:
print("Could not request results from Speech Recognition service; {0}".format(e))
os.chdir(r"C:\Users\vee25\OneDrive\Desktop\Project\Agentspeech")
Outfile=open("STToutput.txt", 'a+')
Outfile.write(t.recognize_sphinx(audio))
Outfile.write('.')
Outfile.close()

def agentAnalysis(self):
    global r
    a = 1
    audioloc=os.path.dirname(r)
    text "Idiot","Stupid","Bitch","Asshole","Just go to our Website","Thats the
Manufacturers Responsibility","Idont Know","Thats not my Department","You will have to
speak with", "Can I transfer your call?","Idont set the Company Policy","You did that
Wrong","Ill have call you back in 5 minutes"]
    greet=["Hello","Morning","Welcome"]
    close=["Thank you","Goognight","Welcome","Niceday"]
ps = PorterStemmer()
    f =
open(r"C:\Users\vee25\OneDrive\Desktop\Project\Agentspeech\STToutput.txt").read()
    words = word_tokenize(f)
    f1=f
    l=sent_tokenize(f1)
    WORD = re.compile(r'\w+')
    ###Repeating Sentences
    c=0
    for i in range(len(l)-2):

```

```

        for j in range(i):
            r=SequenceMatcher(None,l[j],l[i]).ratio()
            if(r>0.65):
                c=c+1

###Banned words
d=0
def get_cosine(vec1, vec2):
    intersection = set(vec1.keys()) & set(vec2.keys())
    numerator = sum([vec1[x] * vec2[x] for x in intersection])

    sum1 = sum([vec1[x]**2 for x in vec1.keys()])
    sum2 = sum([vec2[x]**2 for x in vec2.keys()])
    denominator = math.sqrt(sum1) * math.sqrt(sum2)

    if not denominator:
        return 0.0
    else:
        return float(numerator) / denominator

D1={"Call ID": "1", "Banned Words":d, "Call Attempt": "Successful!", "Agent ID":1,
"Repeating Sentences":c, "Date":date , "Greeting Score":greetscore, "Closing
Score":closescore}
print("Agent analysis done")

opener = "open" if sys.platform == "darwin" else "xdg-open"
subprocess.call([opener, "AgentAnalysis.xml"])

def emotionAnalyse(self):
    global r
    print (r)
audioloc="A"
audioloc=os.path.dirname(r)
print(audioloc)
filename = (os.path.splitext(os.path.basename(r))[0])

```

```

    print(filename)
loaded_model =
keras.models.load_model(r'C:\Users\vee25\OneDrive\Desktop\Project\Emotion_Voice_Detec
tion_Model.h5')
    filenames=[]
plotemotion = []
plotx = []
plotemotion.append("None")
plotx.append(0)
    #making chunks
myaudio = AudioSegment.from_file(r, "wav")
chunk_length_ms = 3000 # pydub calculates in millisec
    chunks = make_chunks(myaudio, chunk_length_ms) #Make chunks of three sec
os.chdir(audioloc+"/")
    for i, chunk in enumerate(chunks):
chunk_name = "chunk{0}.wav".format(i)
    print ("exporting", chunk_name)
chunk.export(chunk_name, format="wav")
    for i in range(0,1):
wave_file=filenames[i]
    predictions = loaded_model.predict_classes(x)
    #making predictions
    if predictions == 0:
pred = "neutral"
    elif predictions == 1:
pred = "calm"
    elif predictions == 2:
pred = "happy"
    elif predictions == 3:
pred = "sad"
    elif predictions == 4:
pred = "angry"
    elif predictions == 5:
pred = "fearful"
    elif predictions == 6:

```

```
pred = "disgust"
elif predictions == 7:
pred = "surprised"
    print(pred)
plotemotion.append(pred)
    print(plotemotion)
plotx.append(plott)
    print(plotx)
    c=c+1

plt.ylabel('Emotions')
plt.xlabel('Customer speech time')
plt.bar(plotx,plotemotion)
plt.savefig('emotion.png')
plt.show()
```

Chapter 7

Testing

7.1 Test cases

We have recorded 20 calls to act as test cases for the system. The scripts for each individual test case are shown below –

7.1.1 Test case 1

Agent - Hello, Good morning, This is Veerus from Waze call center, How may I help you?

Customer - Hi, I am calling regarding no network on my sim card.

Agent - Alright, Could I know the registered sim card number?

Customer - Yes, it's 654321

Agent - This number is registered under the name Rhea, correct?

Customer - Yes, that's me.

Agent - No network is due to suspension of service as you have not paid your bill since 3 months.

Customer - Oh really? I thought I had kept it on automatic payment.

Agent - For resuming your service, you can pay online or visit your nearest waze store

Customer - Thank you for that.

Agent - Will that be all?

Customer - Yes, that will be all.

Agent - Thank you for calling, hope you have a nice day.

Customer - You too.

7.1.2 Test case 2

Agent - Hello, How may I help you?

Customer - Hi, I am calling regarding no network on my sim card.

Agent - Did you try removing the sim card and putting it back again?

Customer - Of course, I did it twice.

Agent - Are you sure you know how to do it properly?

Customer - Why are you being so rude?

Agent - Alright, Could I know the registered sim card number?

Customer - Yes, it's 654321

Agent - This number is registered under the name Rhea, correct?

Customer - Yes, that's me.

Agent - You have not paid your bill and expect network.

Customer - Wait what? I thought I had kept it on automatic payment

Agent - What what? You idiots tell this reason only when you don't pay. Now go and pay the bill.

Customer - Is this the way you talk to your customers.

Agent - Yes, bye.

7.1.3 Test case 3

Agent - Hello, this is Vaishali from Waze Call Centre. How may I help you?

Customer - Hi, my sim card is not getting detected.

Agent - Since when are you having this problem?

Customer - It's been 2 days.

Agent - So how did you survive without your sim for the last 2 days?

Customer - I don't think that concerns you.

Agent- I'm sorry. I was just trying to understand your problem.

Customer - I think you were straying away from the problem. Just stick to the topic and give me a solution pertaining to my problem.

Agent- Since your sim is not getting detected ,it might be out of place. Have you dropped your phone somewhere recently?

Customer - Yes. I think I dropped it in the train a couple of days back.

Agent - Just power off your phone. Remove your sim card and insert it back again.

Customer - I will give that a try. Thank you for your help.

Agent - Thank you for calling. Have a nice day.

7.1.4 Test case 4

Agent- Hello, this is Mukti from Waze Call Centre. How may I help you?

Customer- Hi, this is Anil and my sim card is not getting detected.

Agent- Have you tried removing the sim card and inserting it again?

Customer- Do you think I am a fool?

Agent- I haven't met you so I cannot comment on that.

Customer- Is this how you talk to your customers?

Agent - I am sorry, sir. I was just kidding.

Customer - It's alright. Now give me a solution to my problem and yes, I have tried re-inserting my sim card.

Agent- Sometimes dust can get between the SIM and your phone causing communication issues.

Customer - So what should I do?

Agent - Turn off your phone and remove the SIM card. Clean the gold connectors on the SIM with a clean lint-free cloth. Replace the battery and turn your phone on without the SIM. Turn your phone off, replace the SIM and restart the phone

Customer - I will do this. Thank you so much for your help.

Agent - Will that be all?

Customer - Yes.

Agent - Thank you for calling us. Have a nice day.

Customer - You too.

7.1.5 Test case 5

Agent - Hello, Good morning, How may I help you?

Customer - I recently got a new sim card and I lost all my contacts. How can I retrieve them?

Agent - Your contacts might be saved on your phone. Go to your contacts app. Select the import option. You'll find your contacts there hopefully

Customer - Wow! Thank you so much for your help.

Agent - Glad I could help. Have a nice day.

Customer - You too

7.1.6 Test case 6

Agent - Hello, this is Ramola from Waze Call Centre. How can I help you?

Customer - Hi, I am calling with regard to not being able to send messages.

Agent - Are you able to receive messages?

Customer - Yes. That's what I can't understand.

Agent - The sim card that you are trying to use might not be enabled.

Customer - So what should I do?

Agent - Go to settings and then select the sim card option and enable the sim card that you want.

Customer- I will do this asap. Thank you for your help.

Agent- Thank you for calling us. Have a nice day.

7.1.7 Test case 7

Agent -Hi, good morning. This is Suzan from ABC Call Centre.

Customer- Hello, I am having a slight problem during my calls.

Agent- What problem are you facing?

Customer-The voice of the person who I am talking to is distorted during every call.

Agent- Is there any other problem that you are facing?

Customer - Why don't you solve my first problem before jumping into other problems?

Agent - Why don't you answer my question instead of acting like a prick?

Customer - Mind your language please! Is this how you talk to your customers?

Agent - Yes, this is exactly how I talk to my customers who do not cooperate.

Customer - I am going to complain to the higher authorities and I might change my network provider.

Agent - Do what you want to do.

Customer - I will.

7.1.8 Test case 8

Agent - Hi, this is Amy from YZ Call Centre. How may I help you?

Customer - Hi, I am having a problem pertaining to listening to calls.

Agent - You should visit an ENT specialist.

Customer - Now is not the time for jokes.

Agent -I am sorry. Can you elaborate on what problem you are facing?

Customer -So whenever i am on a call with someone. Their voice sounds distorted throughout the call.

Agent-So this problem is only when you call one specific person, right?

Customer - No, it's the same with everyone.

Agent - I am sorry I cannot help you with this problem.

Customer - Why not?

Agent - I don't know the solution to the problem.

Customer- So do you get paid for not giving solutions to customers facing problems?

Agent - I am sorry to disappoint you.

Customer - This was a waste of time. These Call Centres are useless.

7.1.9 Test case 9

Agent - Hello, this is Ann from ABC Call Centre. How may I help you?

Customer- Hi, I recently bought a Jio sim to avail its benefits. But the sim card is not getting detected somehow.

Agent - Have you tried removing the sim card and inserting it again?

Customer-Yes, i tried that a couple of times. Still not getting detected.

Agent- Sometimes dust can get between the SIM and your phone causing communication issues.

Customer- I live in a really clean area. So I'm pretty sure that dust is not the problem.

Agent- I think compatibility might be the problem. Which phone do you use?

Customer- Samsung Galaxy J2.

Agent- I don't think Jio 4G is compatible with that phone.

Customer- Oh No!

Agent- You can avail it's 3g network by making some minor changes in the settings.

Customer- I don't think I have any other option for now.

Agent-There's not much difference between 3g and 4g anyway. Won't take a lot of time in changing the settings.

Customer- Yes. Thank you so much for your help.

Agent- Thank you for calling us. Have a nice day.

Customer- You too.

7.1.10 Test case 10

Agent- Hi, this is Neeta from Waze Call Centre. How may I help you?

Customer- Hi, I am Rhea and I am calling regarding no network on my sim card.

Agent- Have you tried restarting your phone?

Customer- Yes. There are still no signal bars.

Agent- Head to the settings menu and click on the sim settings and select the LTE option.

Customer-I don't think you understand my problem.

Agent- I do. That's why I am giving you the solution.

Customer- I said that I am not getting network on my sim card but you somehow you interpreted that as me not getting internet.

Agent- Fair point. But sometimes making basic changes in the network settings helps in getting the network signal.

Customer- I don't think that this is going to work. Do you have any other solution?

Agent- No. I am sorry I cannot help you.

Customer- I think it's time for me to change my service provider.

Agent- And switch to what? Airtel? Vodafone? I would rather not have a sim.

Customer - At Least they have better customer service.

Agent - I am sorry I couldn't be of much help. There might be a problem with your phone.
You should go and check it in your nearest Call Centre.
Customer - I will do that.

7.1.11 Test case 11

Agent - Hi, this is Sam from Waze Call Centre. How may I help you?
Customer- Hello. So, whenever i dial a number, it shows that the call is not getting connected.
Agent- Is the problem with just one number?
Customer- No, the same problem happens with many numbers.
Agent- Try checking that the phone number and the area code that you are dialling are correct.
Customer- Yes, I always double check the number before pressing the button to call the person.
Agent- Network problem might be the problem. Try hanging up the call and then try again from another area.
Customer- I live in a pretty open area. I get all the signal bars.
Agent- You should try removing the sim card and inserting it back again. Try sending a message with another sim card so that we can find out whether the problem is with the phone or the sim card.
Customer- I will do that. Thank you for your help.
Agent- Thank you for calling us. Have a nice day.

7.1.12 Test case 12

Agent - Hello, how may I help you?
Customer-Hi there, when people try calling my phone, they get a busy signal.
Agent- People will get a busy signal when you are on the phone with someone else.
Customer- People say that my phone is busy even though i am not talking to anyone.
Agent- Have you tried removing your sim card and inserting it again?
Customer-Yes, I have even waited for a couple of minutes before switching the phone back on.
Agent- You could also toggle the airplane mode on and then wait for a few minutes before switching it back on again.
Customer- I'll check if this works and then I'll get back to you.
Agent- I can give you few more solutions to try out.
Customer- Okay. Go on.
Agent- You can reboot your phone.
Customer-That seems like a tedious process.
Agent- It is. You can also reset the network. I'm sure that both might work.
Customer - I'll try them out. Thank you so much for your help.
Agent- Thank you for calling us. Have a nice day.

7.1.13 Test case 13

Agent- Hello, this is Parth from Waze call center. How may I help you?
Customer- I can't receive any calls on my phone. I want to know why.
Agent- Have you tried restarting your phone?

Customer- Yes, I have already done that.

Agent- Make sure your phone is not in airplane mode or offline mode. If airplane mode or offline mode is activated on your phone, all connections such as calls, mobile data, Wi-Fi, Bluetooth function, etc. will not work.

Customer- I haven't used the airplane mode toggle yet. I'll still check.

Agent-Make sure call blocking is not activated on your mobile device.

Customer- I did know about such a feature. How do I deactivate it?

Agent- Open the Call app. Click on the 3 dots on the corner of the screen. Then click on Call settings and then select - Reject Calls and then deactivate it.

Customer-Okay.

Agent - If possible, try the same sim in another phone. So that we can find out whether the problem is with the phone or with the sim.

Customer - Okay. Thanks.

Agent - Thank you for calling us. Have a nice day.

Customer - Okay.

7.1.14 Test case 14

Agent - Hi, this is Ritika. How may I help you?

Customer-I cannot make outgoing calls from my sim card.

Agent-So this might happen because sometimes the 4G network gets switched automatically from 4G network to 3G during the call when the network environment is not stable.

Customer- That shouldn't affect the calling, right?

Agent-It does actually. If your location is not covered by 3G network by Airtel, your phone will get "No service" and thus lead to failure to making the outgoing calls and incoming calls.

Customer- So what should I do about this?

Agent- First confirm whether your location is covered by a 3g network.

Customer-How should i do that?

Agent- Go to Settings then select Dual SIM and Mobile Network then click on Preferred network type, select 3G only and check if the status bar shows No service.

Customer- Alright. Then?

Agent- If your call fails with 3G network while it is back to normal with the 2G network, that means the location didn't get the 3G network covered by the Airtel operator.

Customer- So should i just change my preferred network type to 2g then?

Agent- Yes.

Customer. Ok. Thank you so much for your help.

Agent. Thank you for calling us. Have a nice day.

7.1.15 Test case 15

Agent- Hi this is Jenifer from Waze Call Center. How may I help you?

Customer- Hi, whenever someone tries to call my phone, it says that it is switched off even when it's not.

Agent- Have you tried restarting your phone?

Customer- Yes, I have done that and i even tried removing the sim card and inserting it back again.

Agent- Did reinserting the sim card help in any way?

Customer- If it would've helped, I wouldn't be calling you for help.
Agent- Try putting in a different sim card in your phone to check if the problem is with the sim card or with the phone.
Customer- I don't think the problem is with the phone. I just got a new iPhone 11 last week.
Agent- So why don't you use that phone?
Customer- I am using that phone only.
Agent- It is highly unlikely that the problem is with the phone.
Customer- Yes. I agree.
Agent- Check whether call blocking is activated in your device. If it is, deactivate it.
Customer- OK. Thanks.
Agent- Have a nice day.

7.1.16 Test case 16

Agent- Hi, this is Ryan from Waze Call center, how may i help you?
Customer- Hi, this is Diksha, I am not able to make calls.
Agent- Is this problem occurring when you call one person only?
Customer- Yes.
Agent- Try calling a different person. If the call connects, there may be a issue with the other person who you were trying to call.
Customer- Oh no wait! I misunderstood your question earlier. I have tried calling different numbers. I still cannot connect to them.
Agent- Okay. Can I know the number which is having this problem?
Customer- Yes. It is 98877867.
Agent- This is registered under the name Diksha, right?
Customer- Yes.
Agent- You haven't paid this month's bill.
Customer- Oh. I am sorry I may have forgotten to do so.
Agent- It's alright. Pay your bills to continue using our services.
Customer- Okay. Thank you.

7.1.17 Test case 17

Agent- How may I help you?
Customer- I am having trouble sending picture messages.
Agent- Try keeping your mobile data switched on while sending picture messages.
Customer- My mobile data is always on.
Agent- Make sure that the picture message settings are correctly set up on your device.
Customer- Alright.
Agent- Which device are you using currently that is facing this problem?
Customer- I am using an iPhone X.
Agent- Check if the MMS messaging is switched on.
Customer- Okay. I am having trouble receiving picture messages as well.
Agent- Make sure your inbox and sent messages folders aren't full. You can free up space on your device by deleting old messages.
Customer- Okay thanks.

7.1.18 Test case 18

Agent-Hi, this is Jim from Waze Call Center, how may i help you?

Customer- My internet is not working properly.

Agent- Is it slow or something?

Customer- Yes.

Agent- Can I have the number which is facing this problem?

Customer- Yes. 98394893.

Agent- The number is registered under the name Pam, right?

Customer-Yes.

Agent-So you have chosen a pre-paid plan. Have you got a message stating that you have exhausted 90% of your data pack?

Customer-Yes.

Agent- So once you have finished using 90% of the data pack, the services are limited and the speed decreases.

Customer-Oh alright. I am in need of internet right now and I am in no mood to go to the shop and recharge.

Agent- Well, if you have balance on your phone, you can dial *111# and select the data pack that you want. It will be cut from your balance.

Customer- I will do that. Thank you for your help.

Agent- Have a nice day.

7.1.19 Test case 19

Agent- Yes?

Customer-Is this Waze Call Center?

Agent- Yes

Customer- My calls are getting disconnected in the middle of the call.

Agent- Ok.

Customer- It happens every time. What should I do?

Agent- Umm I don't know.

Customer- Isn't this the customer help line?

Agent- Ya.

Customer-So, aren't you supposed to *help* me?

Agent- I am sorry I cannot help you since I don't know what to do how to give you the solution to this problem.

Customer- Fine. What a waste of time!

Agent- Ya. Likewise.

7.1.20 Test case 20

Agent- Hi, this is Vikram from Waze Call Center. How can i help you?

Customer- Why does my phone say "emergency calls only" ?

Agent- Have you tried restarting your phone?

Customer- Yes. I have tried restarting it more than once.

Agent- Can you tell me the number that is facing this problem?

Customer- Yes. It is 897889889.

Agent- It shows here that you have paid your pending bills.

Customer- Yes, I have.

Agent- Try removing the sim card and inserting it back again. There might be a possibility that the sim is not placed properly.

Customer- Okay I will do that.

Agent- Are you receiving proper signal? In some rare cases, your phone may display the “Emergency calls only” message when it can’t connect to a wireless tower on your network.

Customer- There is no signal.

Agent - You could do a factory reset or you could take your phone to the closest service center.

Customer - I'll get my phone checked at a service center. Thank you for your help.

Agent- You're welcome. Have a nice day.

7.2 Type of testing used

7.2.1 Unit testing

Unit Testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use.

Unit tests are short code fragments. Each test case is independent from the others. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended. It is also known as component testing.

For each module, a different technique is used to ensure testing.

Emotional analysis was tested against the testing set of the RAVDESS dataset.

Speech to text transcription used gensim similarity. Gensim is used to compute similarities across a collection of documents in the Vector Space Model.

The main class is Similarity, which builds an index for a given set of documents. Once the index is built, you can perform efficient queries like “Tell me how similar is this query document to each document in the index?”.

We found how similar the transcribed output of the recorded calls and the expected output (script of the calls) were with the help of the following steps:

1. Tokenization of words.
2. Tokenization of sentences.
3. Create a dictionary - Gensim allows to create a Dictionary object that maps each word to a unique id. A dictionary maps every word to a number.

4. Create a bag of words - Corpus (a Bag of Words) is basically an object that contains the word id and its frequency in each document, where frequency just lists the number of times each word occurs in the sentence.
5. TFIDF - Term Frequency – Inverse Document Frequency (TF-IDF) is also a bag-of-words model but unlike the regular corpus, TFIDF down weights tokens (words) that appears frequently across documents.
6. Creating similarity measure object
7. Create query document - Here our query document will be the script of the calls.
8. Find similarity between the query document and the initial document (transcribed output).
9. Calculate average similarity.

Agent analysis and hot topic analysis validation was done by third party analysis.

The 20 recorded calls were assessed by third party, this was done with the help of a feedback form that was designed by us. The feedback form would help us in validating the output of our system. The third party would listen to the recording of each call and fill the form respectively. A blank feedback form is shown below –

Agent Analysis

Call-ID - _____ Agent ID - _____

Topic of the call - _____

On the basis of the call you have heard, please select the most appropriate answer -

1. Banned words usage -	Yes <input type="checkbox"/> No <input type="checkbox"/>
2. Repeating Sentences -	Yes <input type="checkbox"/> No <input type="checkbox"/>
3. Greeting -	Yes <input type="checkbox"/> No <input type="checkbox"/>
4. Closing -	Yes <input type="checkbox"/> No <input type="checkbox"/>
5. Call attempt -	Successful <input type="checkbox"/> Unsuccessful <input type="checkbox"/>

Notes (Optional) - _____

Call examined by : _____

Date : _____ Sign : _____

Fig 7.2.2 Feedback form

7.2.2 Integration testing

The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items i.e. assemblages (or groups of units). Test cases are constructed to test whether all the components within assemblages interact correctly. Subordinate stubs are replaced one at time actual module. Tests were conducted at each module was integrated.

Chapter 8

Results and Discussion

Emotional analysis trained model performance -

Accuracy –

The accuracy of the model is 91.86%

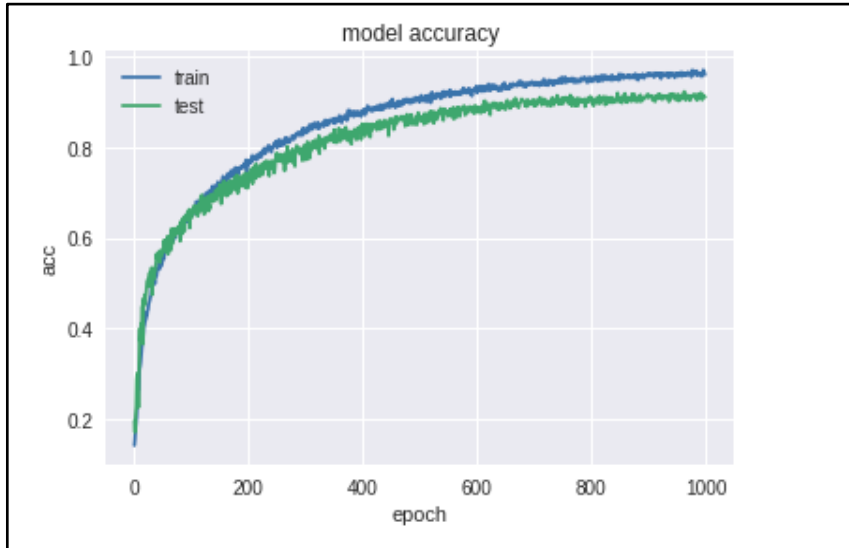


Fig 8.1 Emotional analysis accuracy graph

Loss –

The loss incurred by the trained model is 8.78%

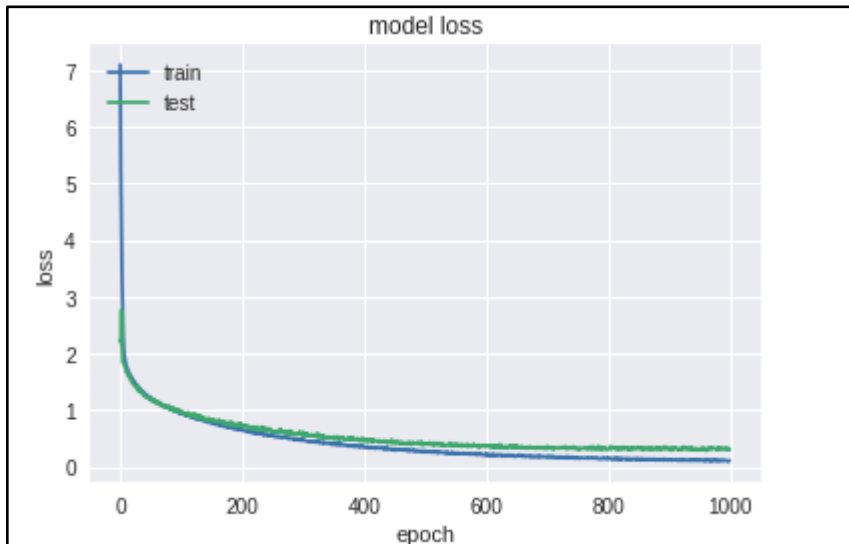


Fig 8.2 Emotional analysis loss graph

Model Summary -

Layer (type)	Output Shape	Param #
conv1d_3 (Conv1D)	(None, 40, 128)	768
activation_4 (Activation)	(None, 40, 128)	0
dropout_3 (Dropout)	(None, 40, 128)	0
max_pooling1d_2 (MaxPooling1D)	(None, 5, 128)	0
conv1d_4 (Conv1D)	(None, 5, 128)	82048
activation_5 (Activation)	(None, 5, 128)	0
dropout_4 (Dropout)	(None, 5, 128)	0
flatten_2 (Flatten)	(None, 640)	0
dense_2 (Dense)	(None, 8)	5128
activation_6 (Activation)	(None, 8)	0
Total params: 87,944		
Trainable params: 87,944		
Non-trainable params: 0		

Fig 8.3 CNN Model summary

Confusion matrix of the model -

```
from sklearn.metrics import confusion_matrix
matrix = confusion_matrix(new_Ytest, predictions)
print (matrix)

# 0 = neutral, 1 = calm, 2 = happy, 3 = sad, 4 = angry, 5 = fearful, 6 = disgust, 7 = surprised

[[128  2  0  3  0  0  1  0]
 [ 2 226  7  8  0  0  8  0]
 [ 4  1 220  4  5  6  0  2]
 [ 2  2  2 241  3  5  7  9]
 [ 2  0  2  2 244  0  1  2]
 [ 1  0  2 19  0 214  2  1]
 [ 0  0  2  2  2  0 121  0]
 [ 0  2  4  0  0  2  2 106]]
```

Fig 8.4 Confusion matrix of Emotional analysis model

Classification Report -

	precision	recall	f1-score	support
0	0.92	0.96	0.94	134
1	0.97	0.90	0.93	251
2	0.92	0.91	0.91	242
3	0.86	0.89	0.88	271
4	0.96	0.96	0.96	253
5	0.94	0.90	0.92	239
6	0.85	0.95	0.90	127
7	0.88	0.91	0.90	116
micro avg	0.92	0.92	0.92	1633
macro avg	0.91	0.92	0.92	1633
weighted avg	0.92	0.92	0.92	1633

Fig 8.5 Classification report of Emotional analysis

Application performance –

For test case 1 –

Expected: Calm throughout and happy with the agent at the end as the problem is solved.

Prediction:

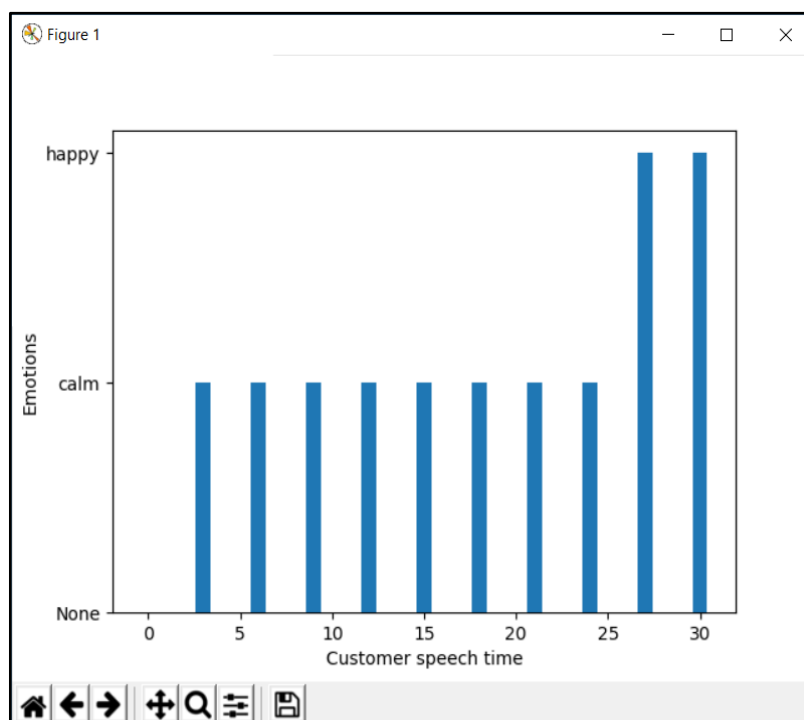


Fig 8.6 Prediction graph of Emotional analysis - Test Case 1

Match: 100%

For test case 2 -

Expected: Calm, Sad, Angry, Calm, Calm, Angry, Angry, Angry

Prediction:

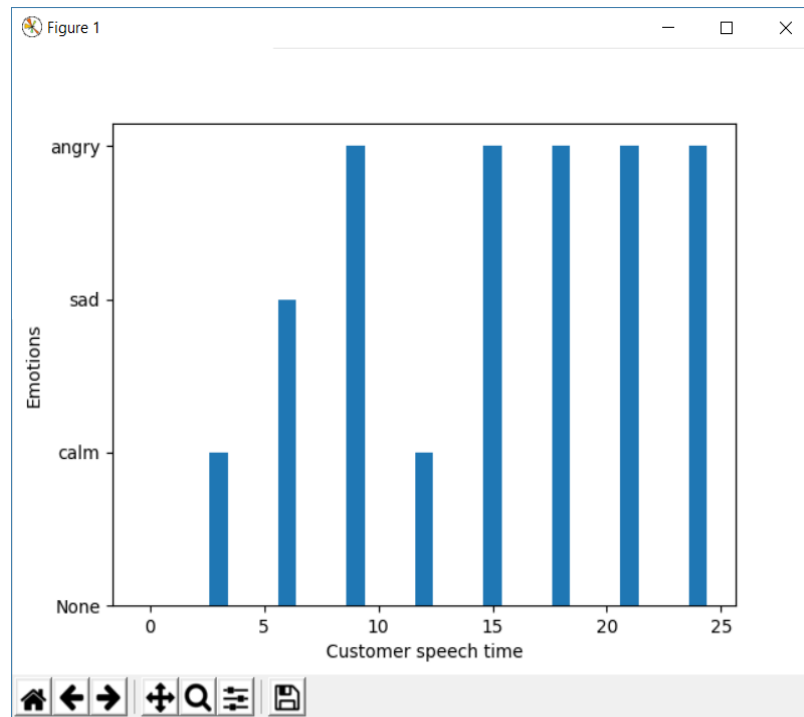


Fig 8.7 Prediction graph of Emotional analysis - Test Case 2

Match accuracy:85%. 6 out of 7 emotions predicted correctly.

Similarly, the emotional analysis for the rest 18 test cases achieved the following accuracy individually –

Call ID	Accuracy
1	100%
2	85%
3	90%
4	100%
5	100%
6	100%
7	100%
8	100%
9	100%
10	100%
11	100%
12	100%
13	85%
14	100%
15	100%
16	100%
17	90%
18	82%
19	90%
20	100%

Average accuracy across 20 calls –96.1%

Speech to text transcription performance –

For test case 1 –

Expected:

Hello Good morning This is Veerus from Waze call center How may I help you. Alright Could I know the registered sim card number. This number is registered under the name Rhea correct. No network is due to suspension of service as you have not paid your bill since 3 months. For resuming your service you can pay online or visit your nearest wazestore. Will that be all. Thank you for calling hope you have a nice day.

Predicted:

hello good morning this is us from Waste Call Centre how may I help you. alright I know the registered SIM card number. this number is registered under the name Riya correct. no network is due to suspension of service as you have not paid your bill since three months. call receiving your service you can play online on visit your nearest Play Store. well that we all. thank you for calling hope you have a nice day.

Accuracy – 89%

For test case 2 –

Expected:

Hello, How may I help you. Did you try removing the sim card and putting it back again. Are you sure you know how to do it properly. Alright, Could I know the registered sim card number. This number is registered under the name Rhea, correct. You have not paid your bill and expect network. What what You idiots tell this reason only when you don't pay. Now go and pay the bill. Yes, bye.

Predicted:

hello how may I help you. did you try removing the SIM card and putting it back again. are you sure you know how to do it properly. alright could I know the registered SIM card number. this number is registered under the name Riya correct. you have not paid your bill and expect network. what what you idiot tell this reason only when you don't pay now 4 and pay the bill. yes bye.

Accuracy: 90%

Similarly, the speech to test transcription for the rest 18 test cases achieved the following accuracy individually –

Call ID	Accuracy
1	89%
2	90%
3	91%
4	82%
5	85%
6	92%
7	85%
8	81%
9	90%
10	81%
11	93%
12	92%
13	94%
14	89%
15	94%
16	93%
17	94%
18	94%
19	74%
20	93%


Average accuracy across 20 calls – 90%

Agent analysis and hot topic analysis -

This is done by comparing the feedback form filled by a person listening the conversation and the system's performance on the same conversation.


The feedback form for two cases are shown below -

For test case 1 –

Agent Analysis	
Call ID - <u>1</u>	Agent ID - <u>1</u>
Topic of the call - <u>Network</u>	
On the basis of the call you have heard, please select the most appropriate answer -	
1. Banned words usage -	Yes <input type="checkbox"/> No <input checked="" type="checkbox"/>
2. Repeating Sentences -	Yes <input type="checkbox"/> No <input checked="" type="checkbox"/>
3. Greeting -	Yes <input checked="" type="checkbox"/> No <input type="checkbox"/>
4. Closing -	Yes <input checked="" type="checkbox"/> No <input type="checkbox"/>
5. Call attempt -	Successful <input checked="" type="checkbox"/> Unsuccessful <input type="checkbox"/>
Notes (Optional) - -	
Call examined by: <u>Muskaan Khanna (Employee at Firstsource BPO)</u>	
Date : <u>17th March, 2020.</u>	Sign : 

Fig

For test case 2 –

Agent Analysis	
Call ID - <u>2</u>	Agent ID - <u>2</u>
Topic of the call - <u>Network</u>	
On the basis of the call you have heard, please select the most appropriate answer -	
1. Banned words usage -	Yes <input checked="" type="checkbox"/> No <input type="checkbox"/>
2. Repeating Sentences -	Yes <input type="checkbox"/> No <input checked="" type="checkbox"/>
3. Greeting -	Yes <input type="checkbox"/> No <input checked="" type="checkbox"/>
4. Closing -	Yes <input type="checkbox"/> No <input checked="" type="checkbox"/>
5. Call attempt -	Successful <input checked="" type="checkbox"/> Unsuccessful <input type="checkbox"/>
Notes (Optional) - -	
Call examined by: <u>Muskaan Khanna (Employee at Firstsource BPO)</u>	
Date : <u>17th March, 2020.</u>	Sign : 

Fig

The system calculated agent analysis is shown through the DynamoDB database below –

<input type="checkbox"/>	Call ID	CustomerContact	AgentID	Banned words	Call Date	Call Duration	Call Time	Call attempt	Closing	Greeting	Rating	Repeating sentences	Topic
<input type="checkbox"/>	001	9933612376	01	No	01/01/2020	0	1135	Successful	Yes	Yes	5	No	Network
<input type="checkbox"/>	002	7669039211	04	Yes	01/01/2020	0	1425	Unsuccessful	No	Yes	2	No	Network
<input type="checkbox"/>	003	8874599234	01	No	01/01/2020	0	1150	Successful	Yes	Yes	5	No	SIM
<input type="checkbox"/>	004	8212599234	03	Yes	01/01/2020	0	1200	Unsuccessful	Yes	Yes	4	No	SIM
<input type="checkbox"/>	005	9923451134	02	No	02/01/2020	0	1310	Successful	Yes	Yes	5	No	SIM
<input type="checkbox"/>	006	9923857232	01	No	02/01/2020	0	1133	Successful	Yes	Yes	5	No	Network
<input type="checkbox"/>	007	9694772408	03	Yes	02/01/2020	0	1221	Unsuccessful	No	Yes	2	No	Network
<input type="checkbox"/>	008	9867699838	02	Yes	02/01/2020	0	1102	Unsuccessful	No	Yes	2	No	Network
<input type="checkbox"/>	009	786859403	04	No	03/01/2020	0	1550	Successful	Yes	Yes	5	No	SIM
<input type="checkbox"/>	010	9970654406	01	Yes	03/01/2020	0	1422	Unsuccessful	No	Yes	3	No	Network
<input type="checkbox"/>	011	7786950487	04	No	03/01/2020	0	1300	Successful	Yes	Yes	5	No	Network
<input type="checkbox"/>	012	8958375844	02	No	03/01/2020	0	1140	Successful	Yes	Yes	5	No	Network
<input type="checkbox"/>	013	9860594785	03	No	04/01/2020	0	1138	Successful	Yes	Yes	5	No	Network
<input type="checkbox"/>	014	9224718392	01	No	04/01/2020	0	1200	Successful	Yes	Yes	5	No	SIM
<input type="checkbox"/>	015	9738117384	04	No	04/01/2020	0	1000	Successful	Yes	Yes	5	No	Deactivation
<input type="checkbox"/>	016	9978593993	01	No	04/01/2020	0	1520	Successful	Yes	Yes	5	No	Network
<input type="checkbox"/>	017	7685849449	03	No	05/01/2020	0	1650	Successful	No	Yes	4	No	Network

Fig

The system was right on 18 out of 20 test cases for agent analysis and 17 out of 20 cases for hot topic analysis.

Accuracy of agent analysis – 90%

Accuracy of hot topic analysis – 85%

Chapter 9

Conclusion

With the help of our system, Customer-Agent call analysis can be done which will help call centers to present a reliable monitoring system resulting in accurate performance measurements by analyzing all the incoming and outgoing calls, reduce total call time, correctly route the customers to respective agents to solve their needs and increase customer satisfaction.

The future scope of our project can be enhanced by including voice activity detection which will help negate all the parts of the speech where there is no activity and also talk over analysis that will help to mine information directly from the speech.