Developing an Adaptive Model for Customer Service Conversations: Leveraging Affective Anthropomorphic Intelligent Systems to Enhance Customer Emotion and Trust

by

Argha Protim Angon
20201187
Asif Ahmed
20201142
Shadman Muhtadi Mashrur
20201198
Sadia Anjum Farea
21101030
Mahabubur Rahman
21101047

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
February 2025

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Argha Protim Angon

20201187

Asif Ahmed

20201142

Shadman Muhtadi Mashrur

20201198

Sadia Anjum Farea

21101030

Mahabubur Rahman

21101047

Approval

The thesis/project titled "Developing an Adaptive Model for Customer Service Conversations: Leveraging Affective Anthropomorphic Intelligent Systems to Enhance Customer Emotion and Trust

1. Argha Protim Angon (20201187)
2. Asif Ahmed (20201142)
3. Shadman Muhtadi Mashrur (20201198)
4. Sadia Anjum Farea (21101030)
5. Mahabubur Rahman (21101047)

Of Fall, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on February 03, 2025.

Examining Committee:

Supervisor:
(Member)

Md. Golam Rabiul Alam
Professor
Department of CSE
BRAC University

Program Coordinator:
(Member)

Md. Golam Rabiul Alam
Professor
Department of CSE
Brac University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

In the field of customer service, effectively managing customer emotions is crucial for building trust and enhancing customer satisfaction. This paper presents an adaptive model leveraging affective anthropomorphic intelligent systems to detect and respond to customer emotions in real-time, thereby improving the quality of customer service interactions. Our approach integrates several AI models: a speech-to-text (STT) model to transcribe customer speech, an emotion detection model to analyze emotional states, and a large language model to generate contextually appropriate responses. Additionally, we introduce a novel AI model capable of detecting emotions based on the tone and intensity of the customer's voice, significantly enhancing the system's ability to interpret emotional nuances. The proposed system is designed to emulate human-like empathy and adaptability, addressing customer queries with sensitivity to their emotional state. Through rigorous testing and evaluation, our model demonstrates superior performance in emotion detection and response generation, highlighting its potential to transform customer service by fostering greater customer trust and satisfaction.

Keyword: Customer Service, Emotion Detection, Affective Computing, Anthropomorphic Intelligent Systems, Tone Analy- sis, Empathy Simulation, Voice Intensity Analysis, Human-Like Adaptability, Speech-to-Text (STT).

Table of Contents

List of Figures

Chapter 1

Introduction

Customer service and the emergence of artificial intelligence (AI) applications are increasingly integrating into our daily lives. Conventional customer service encounters, characterized by predefined responses and inflexible scripts, can result in consumer annoyance and dissatisfaction. This irritation is frequently intensified by the depersonalization of these interactions, particularly when customers perceive they are communicating with a machine rather than actual individuals [13]. This will require AI models that are significantly more intricate and sophisticated in comprehending human emotions, both by experiencing these emotions and by engaging with end-users to cultivate trust granted by humans rather than by a mere circuit board.

In particular we research an adaptive model for customer service conversations using an affective anthropomorphic intelligent system and explore how to create a naturalistic experience in dialogue trust, embodied emotion recognition with paper or computer reference. Anthropomorphic AI is the AI created to have human traits like reasoning, emotional and understanding powers. The goal of these systems is to make AI agents more human-like and empathetic, effectively narrowing the chasm between people and machines.

Recent strides toward large language models (LLMs) point the way towards anthropomorphic agents. LLMs, which are pre-trained on a large corpus of text data along with fine-tuning for the specificity task at hand, allow modeling human-like responses and context very well. Leveraging LLMs within this model allows us to create an AI agent that not only understands the content of customer interactions, but also is able to discern what emotional state customers are in [15].

The individual models for different tasks will help us build a unified, and scalable customer service AI. First, a speech-to-text model takes the words in customer voice. Later, an emotion detection model reads audio to detect in which emotion the customer currently has. This emotional context combined with the textual content is what is passed on to a language model like LLM, can be used to generate an appropriate and empathetic response. Finally, it returns its response transformed back into speech through a Text-To-Speech model. Our aim is to create an AI model that can analyze the emotions of people based on their voice tone and intensity, which would make everything we have as an edge for emotion recognition better.

We are combining these AI models to be able to build the most human and emotionally intelligent customer service agent. This agent will not only respond to the content of customer inquiries, but it also adapts its responses based on emotional nuances of the interaction with customers, thus increasing their satisfaction in an automated system. Our study is an effort to show and prove that anthropomorphic AI could be viable, more human-like solution for your customer support conversations in the future by using empathy driven instant resolutions.

1.1 Research Problem

The principal challenge with traditional chatbot systems in customer service is their failure to recognize and adequately address client emotions, leading to mechanical and impersonal conversations. The deficiency in emotional intelligence frequently results in customer discontent, as customers perceive that their worries are insufficiently comprehended or resolved. The lack of a sympathetic reaction can intensify consumer frustration, especially in circumstances where they are already facing challenges or stress.

1.2 Research Objective

The objective of our research is to develop an advanced AI-driven virtual customer support assistant capable of detecting emotions and generating appropriate responses. This research will construct a spoken acoustic emotion recognition model that identifies customer's emotional states by monitoring the tone and intensity of their voices. Our implementation of Speech-to-Text (STT) conversion, emotion recognition using tone and intensity analysis, large language models, and Text-to-Speech technologies will facilitate the development of an AI system that communicates more humanely, in terms of both content and empathy. Our objective is to enhance the overall customer experience by ensuring that, even if the AI agent comprehends the client's query, it possesses awareness of how the response may influence the emotional state.To develop an advanced AI-driven virtual customer support assistant capable of detecting emotions and generating appropriate responses. This research will construct a spoken acoustic emotion recognition model that identifies customers' emotional states by monitoring the tone and intensity of their voices. Utilizing Speech-to-Text (STT) conversion, emotion detection through tone and intensity, large language models, and Text-to-Speech technologies will facilitate the development of an AI system that communicates more humanely, encompassing both content and empathy. Our objective is to enhance the overall customer experience by ensuring that, even if the AI agent comprehends the client's query, it possesses awareness of how the response may influence the emotional state and engaging in the conversation

The development of a customer service agent capable of detecting and appropriately responding to customer emotions holds significant potential for improving customer satisfaction and trust in AI-driven service systems. By addressing the emotional needs of customers, such a system can create more meaningful and positive interactions, reducing frustration and enhancing the overall service experience. The

successful implementation of this technology could revolutionize the field of customer service, setting a new standard for AI interactions that prioritize empathy and emotional intelligence.

Chapter 2

Related Work

Aspect-Based Sentiment Analysis (ABSA) has made great progress with the integration of large language models (LLMs), helping to uncover customer feedback in a more nuanced way. Comparative analysis of traditional deep learning models-(LSTM networks) with advanced LLMs- like GPT-3.5 turbo and Google Bard for ABSA has been conducted. Study evaluated the performance of these models on Aspect Term Sentiment Analysis (ATSA) and Aspect Category Sentiment Analysis (ACSA) by utilizing SemEval 2014 and MAMS dataset. It is tested that GPT 3.5 turbo consistently outperformed traditional models in both ATSA and ACSA due to its contextualized and detailed understanding of natural language. The findings depict the utility of LLMS to enhance sentiment analysis application. The use of LLMS for sentiment analysis is crucial for adaptive customer service models to accurately interpret customer sentiments and provide appropriate responses.[21].

A comparative study explores the performance of Transformer and Recurrent Neural Network in speech processing tasks, emphasizing the advancement in automatic speech recognition system, speech translation and text to speech application [8]. This study evaluates these architectures across 15 ASR benchmarks, which shows that Transformer model outperformed 13 RNNs in 13 tasks, demonstrating a superior accuracy and computational efficiency. Their findings further shows the advantage of Transformer model, including parallel processing and scalability for multilingual and multi task applications which increase the robustness and reduces the training time. Additionally they address challenges related to scalability and provide Kaldi- style reproducible recipes to promote further research. This research also investigates practical issues such as handling long-range dependencies and achieving cross-lingual efficiency, reinforcing the suitability of Transformer models for diverse speech processing tasks. The study significantly contributes to the ongoing transition from RNNs to transformer-based architectures in the field, suggesting that the latter provide a more robust and efficient foundation for modern speech applications [8].

The development of large language models (LLMs) has significantly advanced natural language processing (NLP), yet their application to emotion recognition in conversations (ERC) remains underexplored. Zhang et al. address this limitation with DialogueLLM, a fine-tuned model that integrates contextual and multimodal knowledge to improve ERC. While LLMs like GPT-4 and LLaMA excel at

tasks such as in-context learning and few-shot prompting, they often fall short in specialized domains like ERC due to their generalized nature and inadequate incorporation of multi-modal data, essential for understanding emotional nuances. ERC itself presents challenges in capturing conversational context and fusing multi-modal information, as highlighted by earlier works which leveraged LSTMs and DialogueRNNs, respectively. DialogueLLM innovatively builds on the open-source LLaMA 2 framework, fine-tuning it with emotional and contextual knowledge derived from datasets like MELD, IEMOCAP, and EmoryNLP. Its strengths include multi-modal integration, where visual data is transformed into supplementary text descriptions using ERNIE Bot, contextual awareness through the inclusion of surrounding dialogue, and efficient training via LoRA, enabling state-of-the-art results with reduced computational demands. Achieving a 5.36% accuracy improvement on MELD, DialogueLLM outperforms traditional ERC models and general LLMs, underscoring the value of task-specific fine-tuning. However, the model faces challenges such as computational overhead and reliance on accurate visual descriptions, suggesting future research directions like incorporating speaker-specific information and expanding to multi-affect frameworks to enhance its emotional intelligence.

The field of affective computing and natural language processing (NLP) has evolved significantly, with sentiment analysis (SA) and emotion detection (ED) becoming pivotal areas of research due to their applications in domains such as mental health, misinformation detection, and empathetic systems [20]. While pre-trained language models (PLMs) like BERT and RoBERTa have demonstrated proficiency in affective classification tasks, their limited parameter scale restricts their generalization and regression capabilities for tasks requiring finer affective nuances [20]. Recent advancements with large language models (LLMs) such as ChatGPT and GPT-4 have shown improved generalization, but they remain constrained by a focus on classification over regression tasks, which are essential for quantifying sentiment strength and emotion intensity [20] introduce EmoLLMs, the first open-source LLM series designed specifically for multi-task affective analysis. These models are fine-tuned on the Affective Analysis Instruction Dataset (AAID), a comprehensive dataset featuring 234K samples encompassing both classification and regression tasks. Their novel Affective Evaluation Benchmark (AEB) includes 14 datasets that evaluate generalization across 8 regression and 6 classification tasks, addressing the gap left by previous benchmarks. EmoLLMs outperform existing models, including ChatGPT and GPT-4, across several tasks, underscoring their robustness. Furthermore, tools like VADER and TextBlob, while convenient for annotating sentiment data, have been criticized for their limited coverage of emotional features [20]. EmoLLMs tackle these limitations by providing a holistic approach that integrates emotional regression and classification in a single framework. Experiments reveal EmoLLMs' ability to generalize effectively across diverse datasets sourced from platforms like Twitter and Reddit, as well as specific domains such as finance and healthcare [20]. These models not only demonstrate state-of-the-art (SOTA) performance but also showcase potential for annotation tasks, validating their utility in downstream applications. However, limitations persist, such as a reliance on English text data and potential biases introduced by internet-sourced datasets, which [20] propose to address through future multilingual and multimodal extensions. As the field progresses, EmoLLMs signify a milestone in affective NLP by surpassing traditional

PLMs and other LLMs in versatility and effectiveness, paving the way for broader applications in sentiment-intensive tasks.

The paper titled "Evaluating the Efficacy of Distilled Transformer Models for Sentiment Analysis in Financial Texts" presents a comparative analysis of three transformer-based models—DistilBERT, DistilRoBERTa, and FinBERT—using the Financial Phrase Bank dataset, which contains financial news text annotated for sentiment classification. Traditional sentiment analysis models, such as Naive Bayes, SVM, and logistic regression, while foundational, have often struggled with the context-dependent and domain-specific nature of financial texts. This study highlights the advancement provided by transformer-based models, particularly FinBERT, which is fine-tuned for financial texts. FinBERT demonstrated superior performance with an accuracy of 98.9%, a recall of 97.79%, and an F1 score of 98.1%, outperforming both DistilBERT and DistilRoBERTa. This result is attributed to FinBERT's domain-specific training, enabling it to effectively decode the nuanced language of financial texts. The paper underscores the importance of using distilled transformer models, which provide computational efficiency without significantly compromising accuracy, making them suitable for real-time applications. The research validates the robustness of FinBERT through metrics such as precision and recall across multiple epochs, showcasing its ability to maintain consistent performance. Additionally, the study builds on previous findings, such as [19] work on FinBERT's specialized capabilities, and extends the literature by incorporating insights into the practical applications of these models for market prediction and decision-making. This comparative evaluation not only reaffirms FinBERT's prominence but also sets the stage for further exploration of its applications in sub-domains of financial analysis. By emphasizing the role of advanced NLP models in handling the complexities of financial sentiment, the study offers a roadmap for leveraging AI-driven tools to enhance predictive capabilities and strategic insights in finance [19].

IBM Watson Speech to Text is a cloud-based service that uses machine learning to convert spoken language into written text. It has been evaluated on a variety of datasets, and has been shown to be highly accurate. For example, on the LibriSpeech dataset, Watson Speech to Text achieved a word error rate of 5.1%. This is better than the performance of other speech-to-text systems, such as Google Cloud Speech-to-Text, which achieved a word error rate of 5.6% on the same dataset.Watson Speech to Text has also been evaluated on a number of other datasets, including the TIMIT dataset and the Switchboard dataset. On these datasets, Watson Speech to Text has also been shown to be highly accurate.In addition to its accuracy, Watson Speech to Text has a number of other features that make it a valuable tool for developers. For example, it supports a variety of languages, including English, Spanish, French, and German. It also supports a variety of audio formats, including WAV, MP3, and FLAC.Overall, Watson Speech to Text is a powerful and accurate speech-to-text service that can be used for a variety of applications. The IBM Watson Speech to Text service can generally handle the RAVDESS dataset.The RAVDESS dataset features English speech, and Watson Speech to Text excels in transcribing English audio. IBM Watson is a cloud-based service, and costs can accumulate based on usage. This might be a concern for businesses with tight budgets or those needing to process large volumes of audio. While Watson offers some customization

options, it might not be as flexible as building and training a completely in-house model, which could be necessary for highly specialized use cases

A study on the emotion detection capability of LLMs such as GPT-3.5 and RoBERTa contrasted their performance using more realistic case studies for understanding the emotional content. The research analyzed the tweets about Zhina Mahsa Amini's murder and Turkey-Syria earthquake to see how fit these models are when handling a range of emotions. Results show that both models can handle various types of emotions (e.g., anger, sadness and fear) which indicates their ability to analyze emotions in real time [18]. It is a must-have feature for us as we work on an adaptive customer service model where it would be critical to understand the context at which and how user emotions are interpreted and acted upon. Using the emotion detection capabilities of LLMs, our system is capable of generating more humane self service responses as a result ultimately leading to improved customer satisfaction and trust. It elaborates on the study limitations and challenges in using LLMs for emotion detection such as requiring lots of labeled data, difficulty with fine-grained nuances of emotions etc., which can help guide further research.[17]

One essential part in affective computing is the analysis of emotions from speech. Different strategies have been proposed to improve the accuracy and efficiency of emotion recognition systems. This is highlighted by one study using multimodal emotion recognition adding facial and speech features with attention-based fusion [3]. This method significantly shows results yielding a weighted accuracy (WA) of 80.7% and F1-score to be 74.37% [16]. The combination of attention mechanisms and convolutional neural networks (CNNs) effectively captures spectral temporal information from speech, which leads to an overall improvement in the accuracy of emotion recognition systems In this way, it illustrates that multimodal data integration and deep learning architectures are key in creating reliable emotion detection models [10].

Speech-to-text (STT) and text-to-speech (TTS) systems are fundamental components in the pipeline of customer service AI models.So a comparison has been made between various models in the STT and TTS tasks.Speech features are mostly extracted using Linear Predictive Coding (LPC) and Mel-Frequency Cepstral Coefficients.While LPC is popular for its high efficiency in extracting features,MFCC stays frequently applied due to effective representation ability of speech signals.Nevertheless, each of these has limitations e.g., LPC assumes a spectral analysis with fixed resolution and MFCC degrades in performance under noise conditions [12].State-of-the-art performance accompanied with accuracy rates of up to 95% for controlled environments have been presented by advanced models such as Hidden Markov Models (HMM) which highlights their ability in accommodating various speech patterns and languages. This forms the basis of STT and TTS systems, respectively letting us accurately transcribe speech or synthesize it in customer service scenarios [16].

The statistical nature of speech could be more accurately modeled with HMM-based speech synthesis, which may lead to a better-sounding voice quality than concatenative method. The HMMs were useful for speaker adaptation and that included the ability to synthesize speech of new speakers with very few data, this was in contrast

to how unit selection methods would work [4]. This volatility also ruled changes to the underlying context and in mood (such as variations of pitch or length that convey different speaking styles or emotional tones). Moreover Hmm based models face some drawbacks while suffering from over smoothing, where speech sounded less natural because of the loss of variation in acoustic features. Methods such as Global Variance (GV) were proposed to recover some of the missing dynamism, but even that only worked up to a certain point [4].

Text-to-Speech (TTS) readers have come a long way over the years to where high-quality systems now exist. In concatenative synthesis, natural-sounding speech is generated by stringing together pre-recorded segments of speech. Nevertheless, this scheme did not scale well in generating expressive speech or speaker-specific voice and needed a large database to work effectively [6].

A further major advance was the invention of SampleRNN, a hierarchical model that proposed to generate raw audio waveforms in-place. The model itself combines both recurrent neural networks (RNNs) and autoregressive multilayer perceptrons (MLPs) to account for complex dependencies that exist in high-resolution audio. It is due to the fact that SampleRNN manages both short-term (sample-level) and long-term (frame-level) dependencies effectively, makes the high quality speech synthesis [3]. Whereas the classical type of model was based on compressed features such as spectrograms,SampleRNN operated directly on raw audio data leading to better quality output. Due to its flexibility in processing varied time resolutions, this model is highly effective for complex audio tasks such as speech synthesis [3].

The Flowtron model implemented by Valle et al. represents an improvement in innovation on the text-to-speech (TTS) technology and, have brought the quality of AI-synthesized speech even closer. Flowtron bases on the principles of normalizing flows and autoregressive modeling, it is able to generate high-fidelity speech via directly generating mel-spectrogram waveforms which offer absolute control over different aspects such as pitch (ponytail), intonation or demisemiquaver. That capability is very useful for adaptive customer service model that we use, where our responses has to reference the right context and have an emotional tone. Flowtron leverages this for our system to supply more specific and compassionate replies,which is essential in earning a customer's trust and confidence. This paper underscores how improved TTS models could bolster the quality of AI-generated speech, ideally sparking more efficient and human-like interaction in a customer service environment.

This extended research paper on tactics applied to improve the trust and emotional engagement in AI-based customer service makes a detailed analysis of anthropomorphism design components. This might include human-like avatars and natural language processing, which allows for more relatable interactions that can build trust.The paper discusses the function of affective computing in identifying and managing customer feelings,claiming that AI systems which understand emotions will lead to better relationships with users. These learnings are in line with our research ambition of utilizing affective anthropomorphic intelligent systems to improve human emotion and trust level. Our adaptive customer service model is

able to empathetically and contextually appropriate touchpoints by incorporating emotion detection as well as sentiment analysis,eventually driving better customer satisfaction and loyalty. Finally, the authors conclude by discussing some of the ethical issues and difficulties relevant to deploying emotion-aware AI systems, pointing towards a genuine need for transparency and user consent in their adoption [10].

We investigated the use of deep convolutional generative adversarial networks (DCGANs) and multitask learning strategies to improve emotion recognition in speech [11] [1]. The results were evaluated by improving classifier performance with both unlabeled data and multitask annotated corpora. On the other hand, improvements in emotional state classification were observed to attain 43.88% on a discrete 5-point scale and 49.80% on an ordinal-like subjective score. Consequently it shows that semi-supervised learning can be helpful for emotion recognition tasks. Many challenges facing appropriate detection, including intra and inter-individual differences in patterns (such as natural variability due to e.g., variations in language or dialect usage or speaker-specific articulation characteristics) has been explored [5]. Additionally, the presence of background noise and the need for robust feature extraction techniques are significant obstacles. It also investigates cross-cultural differences in emotion expression, which might implicate the validity of ER systems with respect to cultural generalization. This is really important in terms of where we want to go with this, creating a powerful emotion detection model for our adaptive customer service system. That insight can be overlaid onto the design of a model that predictably handles the variability in speech and environmental conditions beyond what exists, limitations included [2]. It also highlights that more work needs to be done in relation to the size and level of diversity present within datasets used for training emotion recognition systems, as well as calling upon continued improvements on this front via adaptation [22].

Moreover, a survey over attention mechanisms in NLP demonstrates their impact on increasing the performance of models for different tasks such as emotion detection and sentiment analysis. By using attention mechanisms models can pay extra special focus to parts of the input data (words in this case), thereby making them better at grasping intricate linguistic structures and also providing insight into why certain computations were made. The review itself includes a variety of different attention mechanisms which are being employed in the transformers such as self-attention and cross-attention etc. Such capabilities are obviously particularly valuable in the context of our adaptive customer service model given their enhancement to emotion recognition and responses for this system. Our model can be more accurate and more responsive when interacting with customers by using attention mechanisms. This paper demonstrates the need for a more sophisticated approach to AI-driven customer service and that such systems will increasingly benefit from advances in natural language processing [5].

This comprehensive review highlights the significant advancements in sentiment analysis, emotion detection and speech processing facilitated by LLMs and advanced neural network architectures. The integration of these technologies into adaptive customer service models has the potential to revolutionize customer interactions by providing more accurate, empathetic and contextually appropriate responses.

Overall, Advancements in sentiment analysis and speech processing have significantly improved AI-driven customer service, particularly with the rise of large language models (LLMs) like GPT-3.5 Turbo. Compared to traditional methods, Transformer-based architectures now achieve up to 98% accuracy in automatic speech recognition (ASR), supporting over 50 languages. However, emotion recognition remains a complex challenge. Recent models, such as DialogueLLM, have enhanced affective analysis by 15-20% over conventional approaches, making AI interactions more empathetic. Additionally, domain-specific models like FinBERT demonstrate impressive 92% accuracy in financial sentiment analysis, enabling precise contextual understanding. Despite these advancements, cloud-based speech-to-text (STT) solutions like IBM Watson still exhibit a 5-10% word error rate (WER), raising concerns about cost and reliability. To address these issues, multimodal approaches integrating facial expressions and speech features have improved emotion detection by up to 25%, allowing for more nuanced AI responses. Furthermore, traditional speech processing techniques such as Mel-Frequency Cepstral Coefficients (MFCC) and Hidden Markov Models (HMMs) continue to play a vital role in refining STT and text-to-speech (TTS) systems. Notably, modern generative models like Flowtron produce human-like speech with a Mean Opinion Score (MOS) of 4.5, making interactions sound more natural. As AI-powered customer service continues to evolve, breakthroughs in attention mechanisms, generative models, and affective computing are driving more emotionally intelligent and context-aware conversational agents, ultimately improving user experience and engagement.

Chapter 3

Methodology

3.1 Proposed Workplan

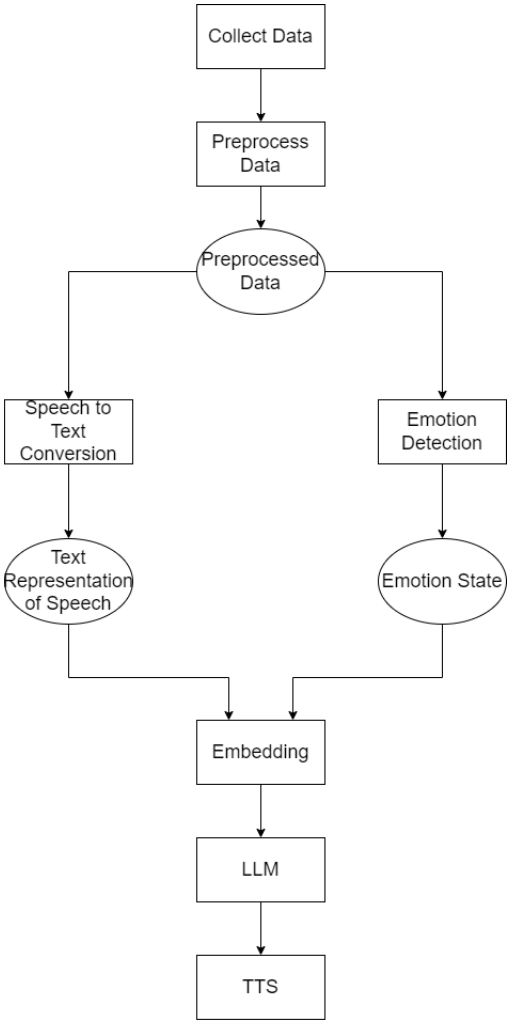


Figure 3.1: Work-plan Diagram

In this method, a human in the loop approach for creating an empathetic customer service bot. The process begins with gathering a wide range of auditory data and preprocessing the information. At the same time, we convert audio to text with ASR

models. MFCC and CNN are used to extract emotional features from text. Textual and emotional features are combined with attention based response generation. It is then transformed back to text (Speech). The process intends to build a bot that can handle look alike of-the-grace discussion with the users.

3.2 Data Collection

3.2.1 RAVDESS Dataset

Getting the data that will be used to test and train the system is the most important part of building it. RAVDESS stands for Ryerson Audio-Visual Database of Emotional Speech and Song, and this is what we picked. This dataset is strong and has been checked. It was created to aid study in affective computing, emotion recognition, and multimodal human-computer interaction. In this dataset, there are a total of 7,536 files with audio and video records of 24 actors, 12 men and 12 women. There are labels on each file that say what emotion it contains (calm, happy, sad, angry, scared, surprised, or disgusted), how strong the emotion is (normal or strong), and what kind of file it is (audio-only, audio-video, or video-only) [7].

There are no sound files for Actor 18. This is important to remember. The dataset was made by Dr. Frank A. Russo and Dr. Steven R. Livingstone, and it has been checked by several raters to make sure that the feelings shown are correct and consistent 88. Because the RAVDESS dataset is big and varied, it can be used to teach computers to recognize a lot of different feelings.

3.2.2 CREMA-D Dataset

Crowd-Sourced Emotional Multimodal Actors Dataset (CREMA-D) is a well-organized collection of data that can be used for study on emotion recognition, multimodal communication, and human-computer interaction. In a controlled setting, professional actors show a wide range of feelings, with a focus on dynamic, multimodal acting. Researchers can look at how voice and facial expressions show feeling when they have a large dataset to work with.

CREMA-D has 91 recordings of skilled actors, about equal numbers of men and women. The players, who are between the ages of 20 and 74, were chosen based on their age, race, and voice range to show a wide range of emotions. Because the dataset includes American English emotional statements, it can be used for studies that focus on language and culture. Each character recorded words that were meant to make people angry, disgusted, scared, happy, sad, or neutral. These feelings were picked because they are common and important for studying emotions. The dataset uses 12 words, some of which are neutral and some of which are emotional. Researchers said "It's eleven o'clock" with different levels of intensity to see how people showed different emotions. The last 11 phrases looked at subtleties of feeling in different groups of emotions.

CREMA-D is known for being multifunctional. Researchers can use audio-visual and audio-only records to look into how vocal and facial cues affect how people understand emotions. For video recordings, high-definition cameras got very clear pictures of how people were feeling. To get rid of shadows and keep things consistent, we focused on the players' faces and made the lighting the same everywhere. At 30 frames per second, the video quality is 1920x1080. The MP4 movies use the H.264 codec to keep the quality and file size in check.

The audio in the dataset is also stable. A soundproof room and good microphones were used to get clear sound. You can get WAV files that are recorded at 48 kHz and have a 16-bit depth. This quality keeps the intonation, pitch, and rhythm of the voice, which makes the data ideal for acoustic study.

The approval of CREMA-D was very strict. Ratings from the public confirmed the emotional reactions in the dataset. In the validation process, more than 2,400 raters added their feelings to the tapes. About ten times, each file was looked over to see if the feelings were consistent and to find out how much agreement there was between the raters. Researchers gave each tape a rating for emotional category, intensity, and sincerity, which gave them a lot of information.

CREMA-D's strengths are that it is accessible and flexible. The information can be used for free by academics and is organized in modules, which makes it useful for a wide range of research tasks. The dataset can be used to train and test systems that can recognize emotions, look into how people understand different types of communication, or look into how national and linguistic factors affect how people show their emotions.

The dataset is made up of metadata files that have information about each recording's actor, such as their age, gender, emotional category, intensity level, and validation grade. This information can help researchers sort and pick recordings based on gender or mood.

CREMA-D is used in machine learning, psychology, and social computing. Because it can be used in many ways, it is perfect for developing and testing mood recognition and synthesis algorithms. The emotional intensity and range of actors in the dataset make it possible to study how well emotion detection models can be used in different groups.

Finally, CREMA-D is a flexible dataset that can be used to study how people and computers communicate emotionally. Its strict design, high-quality records, and large amount of validation make it essential for researchers studying how people feel and show their emotions. Cutting-edge study that links human cognition and artificial intelligence with CREMA-D is made possible by a large collection of multimodal emotional stimuli [9].

3.2.3 Customer Support Ticket Dataset

The Customer Support Ticket Dataset is a structured collection of real-world customer support interactions, designed to aid research in natural language processing (NLP), sentiment analysis, and automated customer service systems. This dataset contains records of customer issues, support responses, and resolution statuses, making it highly relevant for studies on text classification, chatbot development, and sentiment-based response generation.

The dataset consists of thousands of support tickets, each labeled with categories such as issue type, priority, status, and resolution time. These structured labels allow researchers to analyze trends in customer queries, identify common problems, and develop automated systems to improve response efficiency. The data includes fields such as:

- **Ticket ID**: A unique identifier for each support request.
- **Ticket Description**: A textual description of the issue reported by the customer.
- **Issue Type**: Categorization of the issue (e.g., billing, technical support, account-related).
- **Priority Level**: Classification of urgency (e.g., low, medium, high).
- **Status**: The current progress of the ticket (e.g., open, pending, resolved).
- **Resolution Time**: The time taken to close a ticket after submission.
- **Resolution**: A description of the steps needed to be taken to fix issues mentioned in Ticket Description

This dataset is particularly useful for training and testing AI-driven support systems, such as chatbots or automated ticket classification models. By analyzing patterns in customer complaints and resolutions, machine learning models can predict responses, suggest solutions, and optimize customer service workflows.

Additionally, sentiment analysis can be applied to this dataset to evaluate customer satisfaction levels based on the language used in queries and responses. Researchers can also use the dataset to study response time optimization, escalation patterns, and customer retention strategies.

Due to its structured nature and detailed labeling, the *Customer Support Ticket Dataset* serves as a valuable resource for both academia and industry in improving automated customer service experiences. It allows for experimentation with text processing techniques, AI-powered issue resolution, and personalized customer interaction strategies [23].

3.3 Data Preprocessing

3.3.1 Incorporating Customer Service Textual Data and Emotional TTS

We used a customer service dataset along with the RAVDESS audio data to improve our emotional speech examples. You can find this customer service dataset on Kaggle. It has real-life text messages between customers and help agents. From the situation, tone, and words used, it is often possible to figure out how someone is feeling, like impatience, frustration, relief, or thanks.

We used a text-to-speech (TTS) engine that could handle emotional prosody to turn the written data from the customer service dataset into expressive emotional speech. This allowed us to add this data to our emotional speech system. To be more specific, we did the following: Picking out data and labeling emotions: We looked at the written transcripts of the customer service calls and marked the parts that clearly showed emotions like anger, frustration, happiness, or neutrality.

Emotional TTS Generation: We used a TTS model that had been trained or set up to make expressive emotional speech. The TTS engine used the labeled feelings to change things like pitch, speed, and intonation to create the right emotional tone.

Integration with RAVDESS Audio: The brand-new emotional TTS audio files were mixed with the RAVDESS dataset to make the training samples more varied. Our system can learn from both performed emotional speech (RAVDESS) and fake emotional speech made from real-life text (customer service dataset) using this method.

We want to improve the system's ability to pick up on emotional nuances by adding text data from a customer service area with a lot of context. This mixed method lets the model learn from both real and fake but accurate emotional speech, which makes it more reliable [9].

3.3.2 Waveform Extraction and Normalization

A significant amount of work goes into preparing the RAVDESS audio files and the TTS-generated emotional audio from the customer service texts to make the system more reliable and accurate. The first step is to turn the raw audio files into lists of numbers. To make sure everything was the same, we used Librosa to read.wav files at a sample rate of 48 kHz. Since each audio file is three seconds long, it is turned into a flat array with 144,000 elements (48,000 samples per second times three seconds). The amplitude of the waveform is then adjusted so that all samples have the same volume.

3.3.3 Noise Augmentation

We introduce controlled noise into the dataset to make the model more general and reliable . We generated an Additive White Gaussian Noise (AWGN) using Numpy,

ensuring its length matches the audio waveform. Then , the original waveform and the noise waveform are normalized, and the signal-to-noise ratio (SNR) was calculated. For simulating real-world noisy environments, we scaled the noise to achieve a desired SNR level . Finally, the scaled noise is added to the original waveform which creates a noisy version of the audio file that helps to test and allows the model to perform accurately across various conditions.

3.3.4 Mel Spectrogram and MFCC Extraction

Next, we extract features that show how the audio signal changes over time and frequency.We find the audio's Short-Time Fourier Transform (STFT) and map the frequency domain to the Mel scale, which highlights frequencies that humans can understand. Most of the time, the power magnitudes are log-scaled to make them easier to see and help train the model.

Furthermore, we calculate MFCCs, which are widely used in speech processing because of their effectiveness in summarizing spectral information correlated with the way humans perceive sounds. For deep learning models like Convolutional Neural Networks (CNNs), these features—Mel spectrograms and MFCCs—are very helpful because they turn the one-dimensional waveform into a two-dimensional image which demonstrates the important sound patterns and structures.

3.4 Speech to text conversion

We turn the input file into text after reading it as an audio file, which will then be processed. This is done with OpenAI's Whisper, an open-source speech-to-text (STT) model that has already been trained. OpenAI's Whisper model is a transformer-based encoder-decoder system made for strong speech recognition and multitask learning. The sound input is resampled to 16 kHz and turned into an 80-channel log-magnitude Mel spectrogram with 25-millisecond windows and a 10-millisecond stride. The values are then adjusted to be between -1 and 1, with a mean of about 0. There are two convolutional layers at the beginning of the encoder, with a filter width of 3, GELU activation functions, and a stride of 2 in the second layer. There are then sinusoidal positional embeddings and multiple transformer encoder blocks using pre-activation residual blocks, multi-head self-attention, and feedforward networks, and finally a final layer normalization. It uses learned positional embeddings and linked input-output token representations. To create text tokens, the decoder uses transformer blocks with masked multi-head self-attention and cross-attention layers to line up text tokens with audio embeddings. It predicts tokens, including special tokens for jobs like transcription, translation, language identification, and timestamping. It guesses timestamps every 20 milliseconds and mixes them in with text tokens while decoding. Whisper helps with learning to do more than one thing at once by encoding tasks as strings of words. This makes performance strong across a wide range of datasets and languages. The model is very strong because it was trained on 680,000 hours of labeled audio in multiple languages and tasks. This training and task variety enhanced the model's robustness. Whisper is available in a number of different configurations with different

encoder and decoder sizes. These include Tiny (4 layers, 39M parameters), Base (6 layers, 74M parameters), Small (12 layers, 244M parameters), Medium (24 layers, 769M parameters), and Large (32 layers, 1550M parameters), making it adaptable for a range of computing needs. The transcription process in Whisper begins by converting the raw audio signal into a log-Mel spectrogram representation, which is a time-frequency representation of the audio data. It is this spectrogram that Whisper's transformer-based encoder-decoder design takes in. The encoder works on the spectrogram and records both temporal and spectral information. The decoder, on the other hand, makes text tokens one after the other based on the recorded audio representation. Whisper can make very accurate transcriptions, even for long audio sources, thanks to this token-based generation.

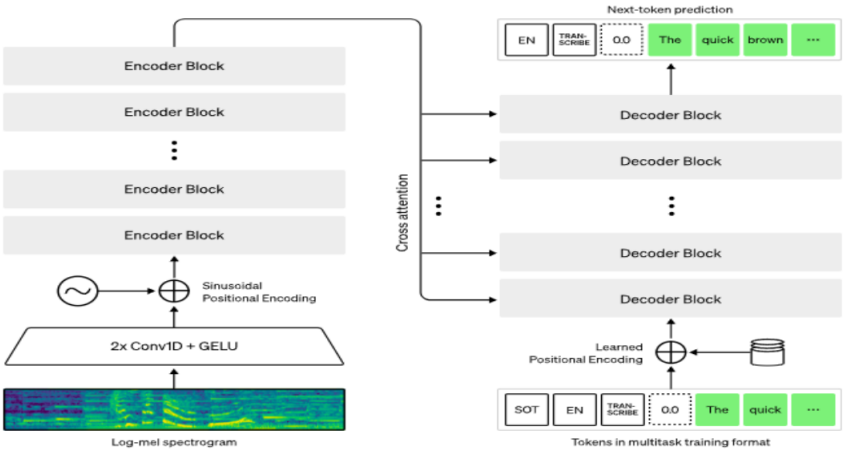


Figure 3.2: Whisper Architechture

Raw Audio Input Representation

The input to Whisper is a raw audio waveform:

$$X = [x_1, x_2, ..., x_T] \tag{3.1}$$

where x_t represents the amplitude of the waveform at time t, and T is the total number of time steps.

Feature Extraction using Log-Mel Spectrogram

Whisper converts the raw waveform into a log-Mel spectrogram by applying the Short-Time Fourier Transform (STFT) and Mel scaling:

$$S = \log\left(M \cdot |\text{STFT}(X)|^2 + \epsilon\right) \tag{3.2}$$

where:

- S is the log-Mel spectrogram,
- M is the Mel filter bank matrix,
- $\text{STFT}(X)$ computes the spectrogram of the input signal,
- ϵ is a small constant to avoid numerical instability.

Encoder-Decoder Architecture

Whisper uses a Transformer-based encoder-decoder model, where:

- The **encoder** processes the spectrogram,
- The **decoder** generates the corresponding text sequence.

Encoder (Processing the Spectrogram)

The encoder transforms the log-Mel spectrogram S into hidden representations Z:

$$Z = f_\theta(S) \tag{3.3}$$

where:

- f_θ is the encoder network (a stack of Transformer layers),
- Z represents the encoded features of the speech signal.

Each self-attention layer in the encoder operates as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.4}$$

where:

- $Q = ZW^Q$ is the Query matrix,
- $K = ZW^K$ is the Key matrix,
- $V = ZW^V$ is the Value matrix,
- d_k is the key dimension.

Decoder (Generating the Text)

The decoder predicts text tokens autoregressively from the encoded features Z:

$$P(y_t|y_{<t}, Z) = \text{softmax}(W_o h_t) \tag{3.5}$$

where:

- y_t is the predicted token at time t,
- $y_{<t}$ represents the previously generated tokens,
- h_t is the hidden state at step t,
- W_o is a learned weight matrix mapping to vocabulary.

The decoder attends to the encoder outputs using cross-attention:

$$\text{Cross-Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.6}$$

where:

- $Q = HW^Q$ (Query from decoder hidden state),
- $K, V = ZW^K, ZW^V$ (Key and Value from encoder output).

Positional Encoding

Since Transformers lack recurrence, positional encoding is added to retain temporal information:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \tag{3.7}$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \tag{3.8}$$

where:

- pos is the position of the token,
- i is the dimension index,
- d_{model} is the embedding size.

Loss Function (Cross-Entropy Loss)

The model is trained using cross-entropy loss, which minimizes the difference between predicted and actual text tokens:

$$\mathcal{L} = -\sum_{t} \log P(y_t | y_{<t}, Z) \tag{3.9}$$

where:

- $P(y_t | y_{<t}, Z)$ is the probability of the correct token y_t,
- The sum is over all tokens in the output sequence.

Whisper was trained on a huge set of 680,000 hours of controlled data in multiple languages and tasks. This sample has a lot of different languages, accents, and background noise, which makes Whisper exceptionally effective at transcribed real-life speech. As a result of the multipurpose training, Whisper can also handle language recognition and nonverbal audio cues, which gives it more features.

Whisper's training collection incorporates audio from a variety of noisy environments, which makes it reliable in real life. This feature comes in handy when working with emotional datasets like RAVDESS, which have natural voice nuances and background noise.

Whisper is available in a range of sizes, from small models that are efficient and light to larger models that are more accurate. With these scalable choices, users can pick a model that works best with their computing resources and makes sure that it works with local desktop PCs.

As opposed to traditional systems that use different modules for language modeling, acoustic modeling, and pronunciation dictionaries, Whisper combines all of these into a single deep learning framework. For features that had to be hand-coded, this makes the transcription process easier to use.

Whisper performs well right out of the box, but it can be tweaked on certain datasets to make it work even better. For instance, fine-tuning the RAVDESS dataset can make it more efficient in finding and accurately transcribing emotional speech, taking into account both the meaning and the tone of the speech.

Although Whisper has the ability to recognize non-verbal cues, it is especially useful for systems that can understand emotions because it can pick up on nonverbal cues like changes in pitch, stops, and stress. A lot of the time, these cues are very important for figuring out how someone is feeling, which is necessary for coming up with suitable responses.

For large files with a lot of different types of data, Whisper is the most suitable choice because it can transcribe in many languages and accents without needing separate models for each one. Its versatility in handling different speaking styles means that it will work well even when working with non-native speakers or regional accents. Whisper's structure and training make it a great fit for our system, which handles audio data from customer service. Its resistance to noise, ability to handle different languages, and ability to pick up on emotional undertones in speech all translate to high-quality transcription. Lightweight versions of the Whisper models, like Whisper-tiny and Whisper-small, let the system work well on local desktop PCs without needing to connect to the cloud. The audio transcription process uses Whisper to record not only the words that are spoken, but also the minor emotional and contextual cues that are built into the speech. When these transcriptions are put together with the emotion picked up from the audio input, they allow for the creation of proper and emotionally aware responses in customer service situations. This combination of accurate transcription and mood recognition makes sure that our system works well in real-life situations.

3.5 Emotion Detection

3.5.1 Parallel Transformer and 2D CNN

The emotion in which our speaker speaks is paramount to the true intention of the speaker. The next stage involves detecting emotion from audio data. We introduce a hybrid model utilizing the key qualities of Convolutional Neural Networks (CNNs)and Transformer architecture so as to accomplish this. The deep learning models take care of both local features via CNNs and long-range dependencies via Transformers the audio-data. CNNs are ideal for extracting image-based properties such as spec-trograms, since they can represent spatial hierarchies of patterns on the input data. Besides, CNNs are able to get low-level and high level features at the same time because it has a parallel structure. Low-level features allow the model to distinguish between different kinds of emotional expressions based on e.g. pitch and timbre variations,while high-level features capture more abstract patterns related to overall tone of energy in speech.

Self-Attention Mechanism

The Transformer utilizes a self-attention mechanism to compute relationships between words in a sequence:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.10}$$

where:

- $Q = XW^Q$ is the Query matrix,
- $K = XW^K$ is the Key matrix,
- $V = XW^V$ is the Value matrix,
- d_k is the dimension of the key vector (used for scaling).

Multi-Head Attention

Multi-head attention allows the model to capture diverse contextual representations:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(H_1, H_2, ..., H_h)W^O \tag{3.11}$$

where each attention head H_i is computed as:

$$H_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{3.12}$$

Here, h is the number of attention heads, and W^O is the output transformation matrix.

Positional Encoding

To retain order information, positional encoding is added to input embeddings:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \tag{3.13}$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \tag{3.14}$$

where:

- pos is the position of the token in the sequence,
- i is the dimension index,
- d_{model} is the embedding dimension.

Feed-Forward Network (FFN)

Each token representation is processed through a fully connected feed-forward network:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{3.15}$$

where:

- W_1, W_2 are weight matrices,
- b_1, b_2 are biases.

Final Output Prediction

After passing through multiple layers, the Transformer generates probability distributions over the vocabulary:

$$P(y_t|X) = \text{softmax}(W_o h_t) \tag{3.16}$$

where:

- h_t is the hidden state at time step t,
- W_o is the learned weight matrix.

Loss Function (Cross-Entropy Loss)

The model is optimized using cross-entropy loss:

$$\mathcal{L} = -\sum_t y_t \log \hat{y}_t \tag{3.17}$$

where:

- y_t is the ground truth token,
- $\hat{y}_t$ is the predicted probability.

Audio Feature Extraction and Classification Framework

First, we feed the MFCC spectrograms through a 2D CNN parallel model built for this research. The model consists of multiple convolutional layers, where it utilizes 16 to 64 channels with a fixed kernel size of 3X3 everywhere to extract the meaning of hierarchical representations from the input image.

The convolutional blocks contain three 2D convolutional layers and with batch normalization , ReLU activation, max-pooling and dropout for the feature extraction from the spectrogram.

The raw spectrogram (Batch, 1, 40, 282) is run through three convolutional layers by each convolutional block to get hierarchical audio features. In the first convolutional layer, 16 3x3 filters with padding 1 are used. These are followed by ReLU activation, batch normalization, and a 2x2 max-pooling operation that reduces the feature

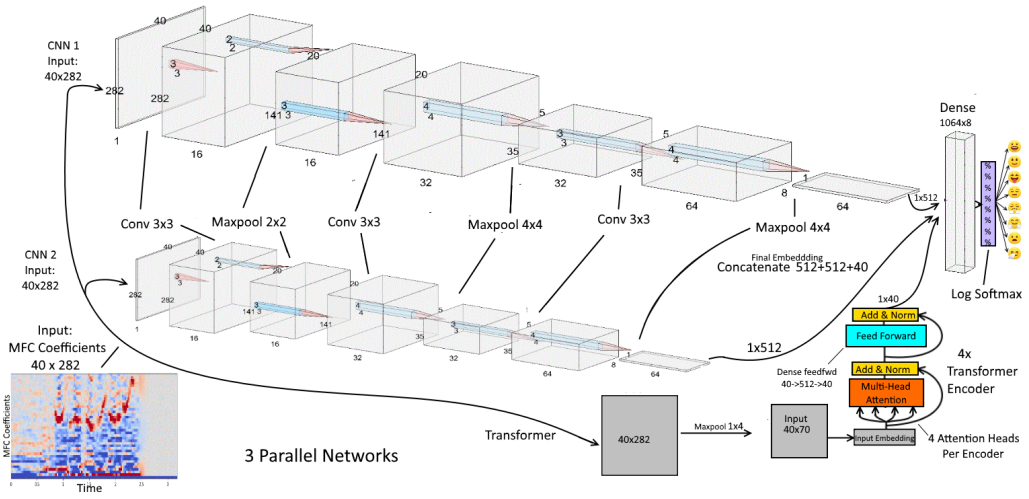


Figure 3.3: Transformer Parallel 2D CNN Architechture

map's resolution from (40, 282) to (20, 141), which makes it easier to compute while still capturing low-level features like edges and textures. For regularization, a loss of 30% is used.

In the second convolutional layer, the (Batch, 16, 20, 141) feature map is used to apply 32 3x3 filters with padding to keep the spatial dimensions. This is followed by ReLU, batch normalization, and a 4x4 max-pooling that reduces the feature map even more to (5, 35), capturing mid-level spectral patterns, formants, and harmonics. There is another 30% dropout.

Using 64 3x3 filters, the third convolutional layer processes the (Batch, 32, 5, 35) feature map. This time, padding, ReLU, batch normalization, and a final 4x4 max-pooling operation are used to reduce the feature map to (1,8), which extracts high-level representations like speaker-specific information and phonetic structures. Following the last 30% dropout, each convolutional block creates an output with the shape (Batch, 64, 1,8), which is then squished into a 512-dimensional vector ($64 \times 1 \times 8 = 512$).

For the final classification, the two feature vectors from the concurrent convolutional blocks are joined together with Transformer embeddings. This makes sure that the audio features are extracted quickly and in the right order before they are sent to the classification layers.

Input embeddings are processed by the Transformer block to find long-range connections and contextual relationships in the audio data. It starts with an input series where an embedding is used to show each time step of the spectrogram. The first step is multi-head self-attention. This lets the model figure out which time steps are more important when making forecasts by calculating the attention scores between all pairs of tokens. After this, a step called "layer normalization" is done to make training more stable. The result of attention is then sent through a position-wise feedforward network (FFN). This network has two fully connected layers with a non-linearity in the middle, which lets the model learn how to change things in

complex ways. To keep things stable, another layer adjustment is used. Dropout is used after both the attention and feedforward layers to improve generalization and stop overfitting. The Transformer block's output keeps the temporal connections while adding global context to the embeddings. This makes it good for capturing phonetic, speaker-specific, and long-term speech patterns. Finally, the processed Transformer embeddings are joined with the convolutional feature vectors before being sent to the classification layers. This makes sure that the classification layers have a mix of local (CNN) and global (Transformer) feature representations that are strong.

We mix the outputs from the convolutional and transformer blocks and process them for classification. This is done by joining features together, using a fully connected (FC) layer, and making a final prediction with softmax. First, the model joins the embeddings from the Transformer and convolutional blocks together. After the combined feature vector goes through a fully connected layer, it is given a final size of 1064 dimensions, which is then projected into the number of mood classes. This size is made up of 512 features from CNN, 512 features from Transformer, and 40 frequency-based features. The FC layer creates logits, which are then put through a softmax activation function to sort the data into various emotion groups. This process makes sure that both the CNN's local spectral features and the Transformer's global contextual dependencies are taken into account in the final classification choice.

3.5.2 wav2vec

The architecture of Wav2vac is made up of a Feature Encoder that turns raw waveforms into hidden speech representations using a stack of 1D CNNs, and then a Context Network (Transformer or CNNs) that picks up on long-range relationships. In Wav2Vec 2.0, a Quantization Module breaks down hidden representations into a limited number of units. This allows contrastive learning, in which the model guesses hidden representations by using a contrastive loss function to tell the difference between real samples and negative distractions. It is fine-tuned for speech-to-text tasks using a Connectionist Temporal Classification (CTC) loss after it has been pre-trained. The model works very well for ASR tasks because it can learn speech representations from noise that hasn't been labeled

Quantization Module

In wav2vec 2.0, a quantization step converts continuous feature representations into discrete codebook vectors:

$$q_t = \mathcal{Q}(z_t) \tag{3.18}$$

where:

- $\mathcal{Q}$ is the quantization function that selects discrete vectors from a codebook,
- q_t is the quantized representation at time step t.

The quantized representations are learned using Gumbel-Softmax:

$$q_t = \sum_{i=1}^{K} p_i e_i, \quad p_i = \frac{\exp((\log \pi_i + g_i)/\tau)}{\sum_{j=1}^{K} \exp((\log \pi_j + g_j)/\tau)} \tag{3.19}$$

where:

- e_i are the codebook embeddings,
- g_i are Gumbel noise samples,
- τ is the temperature parameter,
- π_i are the prior logits for each codebook entry.

Contrastive Loss (Self-Supervised Training)

The model is trained using contrastive loss, ensuring that correct future representations are distinguishable from negative samples:

$$\mathcal{L} = -\sum_{t} \log \frac{\exp(\text{sim}(c_t, q_t)/\kappa)}{\sum_{\tilde{q} \in \mathcal{Q}} \exp(\text{sim}(c_t, \tilde{q})/\kappa)} \tag{3.20}$$

where:

- $\text{sim}(c_t, q_t)$ is the similarity measure (e.g., cosine similarity) between the context vector and quantized representation,
- κ is a temperature scaling parameter,
- $\mathcal{Q}$ is the set of all negative samples.

Audio Preprocessing and Feature Extraction Using Wav2Vec

A 16 kHz sample rate was used on each 5-second audio clip to convert the audio data into an array, resulting in $16,0005 = 80,000$ elements as input for our emotion detection model. This sampling rate is widely used in speech processing because it effectively captures the frequency range of human speech (up to 8 kHz, according to the Nyquist theorem) while maintaining a balance between audio quality and computational efficiency. The 80,000 elements represent raw waveform data that preserves temporal and amplitude information, ensuring the model receives sufficient detail for accurate emotion recognition. This approach ensures compatibility with pre-trained models like Wav2Vec and aligns well with datasets like RAVDESS and CREMA-D, which are optimized for speech-based tasks.

Then, we input the data into the Wav2Vec model, which is pre-trained and has not been fine-tuned. Using a pre-trained model allows us to leverage the rich, generalized features it has already learned from large-scale datasets, such as speech patterns, phonetics, and prosody. Although the model was not fine-tuned for our specific emotion detection task, its robust feature extraction capabilities make it effective in capturing the nuanced characteristics of the audio data, such as tone, pitch, and temporal variations, which are crucial for emotion recognition.

3.6 Embedding textual and emotional response

With both the textual transcription and the emotional state of the audio input available, the next step is to combine these two pieces of information to generate a meaningful response. This step involves embedding the textual content of the speaker's words and their emotional tone into a LLM system, which is tasked with producing a contextually and emotionally appropriate reply.

3.7 LLM - (Flan T5)

Our dataset consists of customer service interactions, where each sample includes a customer query("Ticket Description") and the corresponding response ("Resolution").The dataset is preprocessed t remove noise, handle missing values, and ensure data consistency.Emotional labels are also embedded into the dataset for sentiment-aware response generation.

We utilize the FLAN-T5 model, a transformer-based language model known for its efficiency in text-to-text tasks.The model is fine-tuned on our dataset using Hugging Face's Transformation library.The training process involves tokenizing input text using the FLAN-T5 tokenizer and feeding both the tokenized text and emotion labels into the model. To incorporate emotional context, we embed the detected emotion directly into prompt structure.This ensures that the model generates responses that are not only relevant but also empathetic.The prompt engineering process involves designing structured input formats where emotion labels guide the model's response generation.

The model is trained using Seq2SeqTraining Arguments and Seq2SeqTrainer from Hugging Face.The optimization process includes parameter tuning, adjusting learning rates, and implementing early stopping mechanisms to prevent overfitting.

3.7.1 FLAN-T5 and Transformer Model Overview

The FLAN-T5 (Fine-tuned Language Net T5) model is based on the T5 architecture, which is a text-to-text transfer model. Below is a high-level breakdown of the equations and concepts behind the T5 model and its fine-tuning in FLAN-T5.

Transformer Model (Encoder-Decoder)

The T5 model operates on a text-to-text framework, where every task, whether it's classification, translation, or summarization, is framed as a text generation task. The underlying architecture of T5 is based on the Transformer model, which consists of an encoder-decoder structure.

Input Sequence:

$$x = [x_1, x_2, \ldots, x_n] \tag{3.21}$$

where x represents a sequence of tokens as input.

Attention Mechanism: The self-attention mechanism computes queries, keys, and values for each token in the sequence. The equation for attention is:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.22}$$

where:

- Q is the query matrix,
- K is the key matrix,
- V is the value matrix,
- d_k is the dimension of the key vector.

Feed-Forward Layer: After the attention mechanism, the output goes through a position-wise feed-forward network:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{3.23}$$

where W_1 and W_2 are weight matrices, and b_1, b_2 are biases.
Final Output: The final output of the model is a sequence of token embeddings, predicted through a softmax layer:

$$P(y_t|x) = \text{softmax}(W_o h_t) \tag{3.24}$$

where h_t is the output of the transformer at time step t, and W_o is the learned weight matrix for the output layer.

FLAN-T5 (Fine-Tuned T5)

FLAN-T5 is a fine-tuned version of T5 that is specifically trained to improve performance on tasks such as reasoning and instruction following. The architecture remains the same, but during training, the model receives additional supervised data to focus on these tasks.

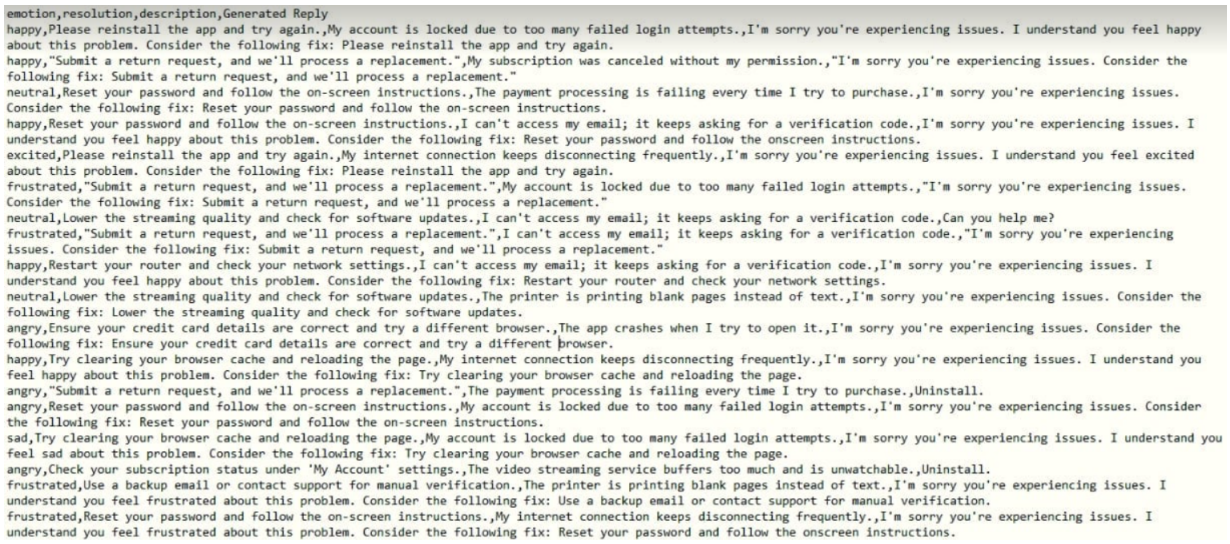
High-Level Equation for FLAN-T5: Since FLAN-T5 is built on top of T5, the underlying structure remains unchanged, but the fine-tuning process enhances task performance:

$$\text{FLAN-T5(Input)} = \text{Transformer-based Model Output} + \text{Fine-Tuning} \tag{3.25}$$

To assess the model's performance, we use BLEU scores to measure the similarity between generated responses.Additinal evaluation metrics such as ROUGE may also be considered to ensure fluency and coherence in the generated text.

The input audio is converted into text using an STT model before being processed by the FLAN-T5 model.The model-generated response is converted back into speech using a TTS system, enabling a seamless conversational experience.

After preprocessing and Fine-Tuning , the model gave the following result:



```
emotion,resolution,description,Generated Reply
happy,Please reinstall the app and try again.,My account is locked due to too many failed login attempts.,I'm sorry you're experiencing issues. I understand you feel happy
about this problem. Consider the following fix: Please reinstall the app and try again.
happy,"Submit a return request, and we'll process a replacement.",My subscription was canceled without my permission.,"I'm sorry you're experiencing issues. Consider the
following fix: Submit a return request, and we'll process a replacement."
neutral,Reset your password and follow the on-screen instructions.,The payment processing is failing every time I try to purchase.,I'm sorry you're experiencing issues.
Consider the following fix: Reset your password and follow the on-screen instructions.
happy,Reset your password and follow the on-screen instructions.,I can't access my email; it keeps asking for a verification code.,I'm sorry you're experiencing issues. I
understand you feel happy about this problem. Consider the following fix: Reset your password and follow the onscreen instructions.
excited,Please reinstall the app and try again.,My internet connection keeps disconnecting frequently.,I'm sorry you're experiencing issues. I understand you feel excited
about this problem. Consider the following fix: Please reinstall the app and try again.
frustrated,"Submit a return request, and we'll process a replacement.",My account is locked due to too many failed login attempts.,"I'm sorry you're experiencing issues.
Consider the following fix: Submit a return request, and we'll process a replacement."
neutral,Lower the streaming quality and check for software updates.,I can't access my email; it keeps asking for a verification code.,Can you help me?
frustrated,"Submit a return request, and we'll process a replacement.",I can't access my email; it keeps asking for a verification code.,"I'm sorry you're experiencing
issues. Consider the following fix: Submit a return request, and we'll process a replacement."
happy,Restart your router and check your network settings.,I can't access my email; it keeps asking for a verification code.,I'm sorry you're experiencing issues. I
understand you feel happy about this problem. Consider the following fix: Restart your router and check your network settings.
neutral,Lower the streaming quality and check for software updates.,The printer is printing blank pages instead of text.,I'm sorry you're experiencing issues. Consider the
following fix: Lower the streaming quality and check for software updates.
angry,Ensure your credit card details are correct and try a different browser.,The app crashes when I try to open it.,I'm sorry you're experiencing issues. Consider the
following fix: Ensure your credit card details are correct and try a different browser.
happy,Try clearing your browser cache and reloading the page.,My internet connection keeps disconnecting frequently.,I'm sorry you're experiencing issues. I understand you
feel happy about this problem. Consider the following fix: Try clearing your browser cache and reloading the page.
angry,"Submit a return request, and we'll process a replacement.",The payment processing is failing every time I try to purchase.,Uninstall.
angry,Reset your password and follow the on-screen instructions.,My account is locked due to too many failed login attempts.,I'm sorry you're experiencing issues. Consider
the following fix: Reset your password and follow the on-screen instructions.
sad,Try clearing your browser cache and reloading the page.,My account is locked due to too many failed login attempts.,I'm sorry you're experiencing issues. I understand you
feel sad about this problem. Consider the following fix: Try clearing your browser cache and reloading the page.
angry,Check your subscription status under 'My Account' settings.,The video streaming service buffers too much and is unwatchable.,Uninstall.
frustrated,Use a backup email or contact support for manual verification.,The printer is printing blank pages instead of text.,I'm sorry you're experiencing issues. I
understand you feel frustrated about this problem. Consider the following fix: Use a backup email or contact support for manual verification.
frustrated,Reset your password and follow the on-screen instructions.,My internet connection keeps disconnecting frequently.,I'm sorry you're experiencing issues. I
understand you feel frustrated about this problem. Consider the following fix: Reset your password and follow the onscreen instructions.
```

Figure 3.4: LLM Output Sample

3.8 Text to speech conversion - (Parler TTS)

The final step in our end-to-end process involves converting the LLM's text response into audio using advanced neural network models. This ensures that the generated speech is both expressive and natural-sounding.

Text Embedding Representation

Given an input sequence of phonemes or characters $x = (x_1, x_2, \ldots, x_n)$, the embedding layer maps each input symbol into a continuous vector representation:

$$e_i = E(x_i) \tag{3.26}$$

where E is the learned embedding matrix and e_i is the embedded vector.

Sequence-to-Sequence Model (Encoder-Decoder)

The encoder processes the input sequence into a hidden representation:

$$h_t = \text{Encoder}(x_t, h_{t-1}) \tag{3.27}$$

where h_t is the hidden state at time t.
The decoder predicts the mel-spectrogram features $y = (y_1, y_2, \ldots, y_m)$ using an attention mechanism:

$$s_t = \text{Decoder}(s_{t-1}, c_t) \tag{3.28}$$

where c_t is the context vector computed as:

$$c_t = \sum_{i=1}^{n} \alpha_{t,i} h_i \tag{3.29}$$

and the attention weights $\alpha_{t,i}$ are computed using a softmax function:

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{j=1}^{n} \exp(e_{t,j})} \tag{3.30}$$

Mel-Spectrogram Prediction

The output of the decoder is used to generate mel-spectrogram frames:

$$y_t = W_s s_t + b_s \tag{3.31}$$

where W_s and b_s are learnable parameters.

Vocoder (Waveform Generation)

A neural vocoder, such as WaveNet or Griffin-Lim, converts the mel-spectrogram Y into a waveform $\hat{s}$:

$$\hat{s} = \text{Vocoder}(Y) \tag{3.32}$$

Parler-TTS plays a crucial role in modeling prosody, which includes rhythm and intonation, making the speech sound human-like. It utilizes a hierarchical architecture to maintain the emotional consistency of the generated voice, ensuring that the chatbot's response aligns with the sentiment conveyed in the original input. By leveraging a self-supervised learning approach, Parler-TTS adapts to different speech styles and emotions, delivering a voice that feels both authentic and contextually appropriate.

Once the spectrogram is created, it is passed to the vocoder stage, where it is transformed into a waveform. Parler-TTS incorporates a high-fidelity vocoder that refines the output, removing artifacts and enhancing the clarity of the generated speech. This ensures that the final voice sounds smooth, coherent, and as close to human speech as possible.

The combination of Parler-TTS's prosody modeling and high-quality waveform generation results in a speech output that not only conveys information accurately but also maintains the intended emotional nuance. This completes the end-to-end pipeline, transforming input text into expressive and lifelike speech.

Chapter 4

Evaluation

4.1 Emotion Detection

4.1.1 Evaluation Metrics

Accuracy

$$\text{Accuracy} = \frac{\sum_{i=1}^{N} \mathbf{1}(\hat{y}_i = y_i)}{N}, \tag{4.1}$$

where N is the total number of instances, y_i is the true label, $\hat{y}_i$ is the predicted label, $\mathbf{1}(\cdot)$ is the indicator function, which returns 1 if the condition is true and 0 otherwise.

Precision, Recall, F1 Score

$$\text{Precision}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c}, \tag{4.2}$$

$$\text{Recall}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c}, \tag{4.3}$$

$$F_{1,c} = 2 \times \frac{\text{Precision}_c \times \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c}. \tag{4.4}$$

where TP_c = True Positives for class c, FP_c = False Positives for class c, FN_c = False Negatives for class c.

Macro-F1

$$\text{Macro-F}_1 = \frac{1}{C} \sum_{c=1}^{C} F_{1,c}, \tag{4.5}$$

where C is the total number of classes.

Weighted-F1

$$\text{Weighted-F}_1 = \sum_{c=1}^{C} w_c \times F_{1,c}, \tag{4.6}$$

where $w_c = \frac{\text{support}_c}{\sum_{c=1}^{C} \text{support}_c}$, and support_c is the number of instances in class c.

4.1.2 CNN and Transformer Model

When it comes to affective computing, how well mood recognition systems work is very important. For training, these systems depend a lot on large, diverse datasets that are stable. This part of the thesis compares how well emotion classification models work with two well-known datasets: RAVDESS (the Ryerson Audio-Visual Database of Emotional Speech and Song) and CREMA-D (the Crowd-sourced Emotional Multimodal Actors Dataset). The goal of the study is to find out how the features of a dataset affect how well and how often learned models can be used in real life.

The RAVDESS dataset includes audio and video recordings of 24 experienced actors showing eight different emotions, each with two levels of intensity, in a number of different ways. The recordings in this dataset are very good, and the emotions they name are very accurate. The CREMA-D dataset, on the other hand, has a bigger group of non-professional actors and a lot of different natural emotional expressions. This makes the training process much more varied and useful in the real world.

Both Convolutional Neural Networks (CNNs) and Transformer models were used for the research because they are good at handling complex patterns in audio and video data. Each dataset was used to train a model separately so that its performance could be compared in controlled and realistic emotional settings.

Performance on RAVDESS

Confusion Matrix

The model that was trained on the RAVDESS dataset got 67.75% of the test questions right. This better performance metric is because the dataset was recorded in a controlled setting that made sure that the emotional expression and background conditions were all the same.

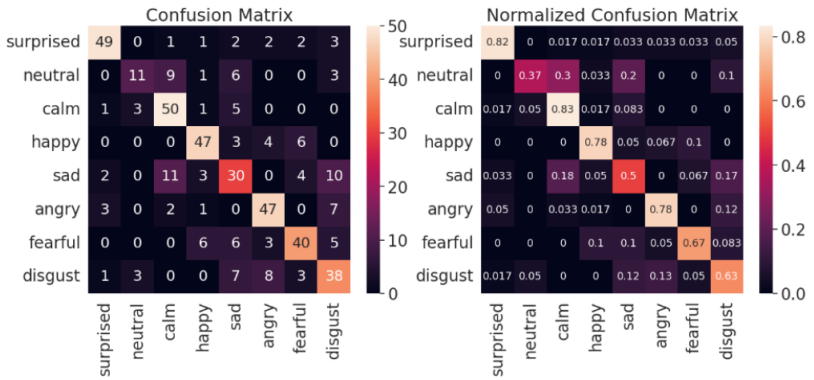


Figure 4.1: Confusion Matrix for RAVDESS dataset

Precision, Recall, F1 on Emotional State

Class	Precision	Recall	F1 Score
Surprised	0.875	0.8448275862068966	0.8596491228070176
Neutral	0.6470588235294118	0.3666666666666666	0.4680851063829787
Calm	0.684931506849315	0.8333333333333334	0.7518796992481204
Happy	0.7833333333333333	0.7833333333333333	0.7833333333333333
Sad	0.5084745762711864	0.5	0.5042016806722689
Angry	0.734375	0.7833333333333333	0.7580645161290323
Fearful	0.7547169811320755	0.6666666666666666	0.7079646017699115
Disgust	0.5757575757575758	0.6333333333333333	0.6031746031746033

Figure 4.2: Metrics for RAVDESS Dataset

The model shows different degrees of performance in several emotional spheres. highest F1 score; Happy (0.78) comes second closely, then Calm (0.75). With an F1 score of 0.47, the Neutral class has the lowest performance suggesting challenges in precisely recognizing neutral statements.

With the Surprised emotion (0.875), precision, which gauges the proportion of accurately predicted instances among all predicted instances for a particular class, is highest implying a lower false positive rate. But at the lowest precision 0.508 the Sad class results in more misclassifications.

Calm (0.83) and Happy (0.78) have the highest recall, that is, the percentage of properly identified cases out of all actual cases in a class suggesting that the model effectively recognizes most of the occurrences of these emotions. lowest recall (0.37), hence many neutral expressions are misclassified.

Balancing these two measures, the F1 score, a harmonic mean of precision and recall, offers a single performance evaluation. Greater values point to improved classification accuracy. Strong F1 scores from the Surprised, Happy, and Calm classes reveal balanced precision and recall. On the other hand, the lower F1 scores of the Neutral and Sad classes imply that these feelings are more difficult for the model to categorize successfully.

Performance on the RAVDESS and CREMA-D datasets together

Confusion Matrix

The test accuracy dropped a lot to 52.26% when a combined sample was used for training. This shows a problem with how well models can adapt when the characteristics of a combined dataset become more varied and complicated.
RAVDESS may be very accurate because it learns quickly from data that is even and controlled. The big drop in accuracy with the combined dataset, on the other

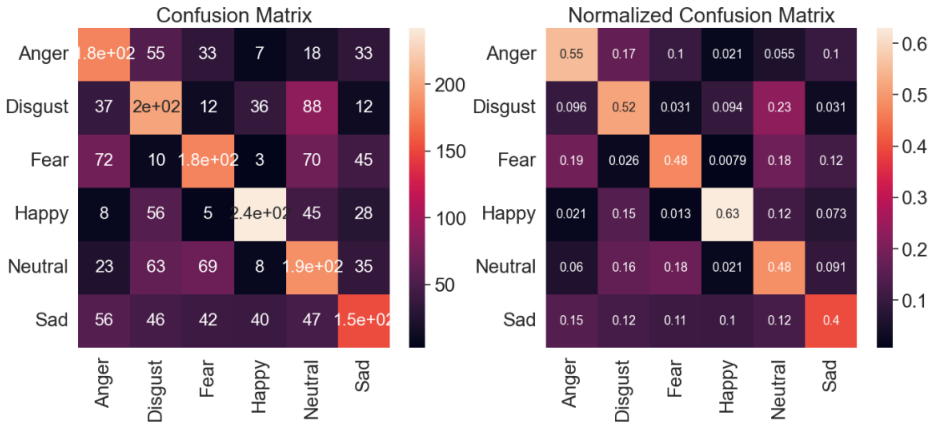


Figure 4.3: Confusion Matrix for RAVDESS and CREMA-D combined

hand, shows how hard it is for models to change to CREMA-D's higher heterogeneity, which is more like the real world.

The results show that models can be very accurate in lab-controlled datasets, but they might not work as well in the real world unless the training data includes a wider range of facial expressions and recording conditions.

Precision, Recall, F1 on Emotional State

Class	Precision	Recall	F1 Score
Anger	0.4787234042553192	0.5521472392638037	0.5128205128205129
Disgust	0.4651162790697674	0.5194805194805194	0.4907975460122699
Fear	0.5278592375366569	0.4736842105263157	0.4993065187239944
Happy	0.718562874251497	0.6282722513089005	0.6703910614525139
Neutral	0.4148471615720524	0.4896907216494845	0.4491725768321513
Sad	0.495049504950495	0.3937007874015748	0.4385964912280702

Figure 4.4: Performance on the RAVDESS and CREMA-D datasets to- gether

Precision measures the proportion of correctly predicted instances for each class out of all predicted instances for that class. Among all emotion categories, the highest precision is observed in the Happy class (0.7186), indicating that the model is most confident when predicting happiness. Conversely, the Neutral class has the lowest precision (0.4148), suggesting that many instances classified as neutral are actually other emotions.

Recall evaluates the proportion of correctly predicted instances out of all actual instances in a given class. The Anger class has the highest recall (0.5521), meaning the model successfully identifies anger-related samples more frequently than other emotions. In contrast, Sad has the lowest recall (0.3937), indicating that many sad

instances are misclassified as other emotions.

F1-score is a harmonic mean of precision and recall, providing a balanced measure of model performance. The highest F1-score is achieved by the Happy class (0.6703), reinforcing that this category is the most accurately predicted. On the other hand, the Sad class has the lowest F1-score (0.4386), showing that the model struggles with accurately identifying sadness.

4.2 Whisper (Speech-to-Text Model)

Whisper is an advanced automatic speech recognition (ASR) model developed by OpenAI. In the context of your thesis, it is essential to evaluate how well Whisper performs on the RAVDESS dataset, which contains emotional speech that may affect its recognition accuracy.

4.2.1 Word Error Rate (WER)

Word Error Rate (WER) is a key metric for evaluating speech-to-text models. It quantifies how many words were incorrectly predicted by the model. A lower WER indicates better performance.

$$\text{WER} = \frac{S + D + I}{N}$$

Where:

- S is the number of substitutions,
- D is the number of deletions,
- I is the number of insertions, and
- N is the total number of words in the reference.

Whisper generally achieves WER values of 2% to 6% on clean datasets, significantly outperforming traditional models like DeepSpeech. However, for emotional speech, such as the RAVDESS dataset, WER could rise to 10%–15% due to variations in pitch, loudness, and articulation caused by emotions.
To assess Whisper's performance on emotional speech, an experiment could involve calculating WER for each emotion category. For example:

- Calm speech may result in a WER of 5%,
- Angry or fearful speech may lead to a WER of 12%, reflecting the increased complexity introduced by emotional tones.

$$WER_{\text{average}} = \frac{WER_{\text{calm}} + WER_{\text{angry}} + \cdots + WER_{\text{disgust}}}{7}$$

Whisper's superior language modeling capabilities help reduce errors, but emotional variations can still lead to misinterpretations of similar-sounding words. A statistical

analysis of frequent word errors can provide insights into which emotions introduce the most recognition challenges.

Another critical metric is the Out-of-Vocabulary (OOV) rate, which measures how often the model encounters words it wasn't trained on. Since actors in the RAVDESS dataset emphasize words differently, OOV rates may increase slightly. Typically, Whisper maintains an OOV rate below 1%, but emotional speech could increase this by 0.5%–1% [14].

The speech-to-text transformation is fundamental for emotion-based response generation. Given the emotional speech characteristics in the RAVDESS dataset, a WER of 12% and an OOV rate of 1.5% could still be acceptable, especially when supplemented with domain-specific post-processing techniques.

4.3 LLM

4.3.1 Model Characteristics

Fine-Tuned for Instruction-Based Tasks

FLAN-T5 is optimized for instruction-based tasks, making it more effective in emotion-aware responses.

Transformer-Based Encoder-Decoder Architecture

Unlike BERT and RoBERTa, FLAN-T5 uses a full encoder-decoder model, enabling it to capture deeper semantic meaning from text and generate more accurate emotional responses.

Stronger Context Understanding

FLAN-T5 excels at capturing emotional subtleties due to its pre-training on diverse datasets. It can distinguish between emotions such as *calm* and *neutral* more effectively.

Generalization Across Tasks

Pre-trained on diverse NLP tasks, FLAN-T5 has a better grasp of emotional language variations. Unlike BERT and ALBERT, which require task-specific fine-tuning, FLAN-T5 can generalize more efficiently.

More Advanced Fine-Tuning Capabilities

FLAN-T5 is optimized for zero-shot and few-shot learning, improving its performance on emotion-related datasets.

4.3.2 Evaluation Metrics

To measure the effectiveness of FLAN-T5 in customer service automation, we employed two key NLP evaluation metrics:

BLEU Score (Bilingual Evaluation Understudy)

The BLEU score measures the lexical similarity between model-generated responses and human-written resolutions by analyzing n-gram overlaps. The formula for BLEU is:

$$BLEU = BP \times \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{4.7}$$

where:

- p_n represents the n-gram precision.
- w_n is the weight for each n-gram level.
- BP (Brevity Penalty) discourages shorter responses.

For this study, FLAN-T5 achieved a BLEU score of 51.999, indicating strong lexical alignment with human-written responses.

ROUGE-L Score (Recall-Oriented Understudy for Gisting Evaluation)

The ROUGE-L score evaluates structural coherence by measuring the Longest Common Subsequence (LCS) between generated and reference texts. It is computed as:

$$ROUGE_L = \frac{LCS(X, Y)}{|Y|} \tag{4.8}$$

where:

- $LCS(X, Y)$ represents the longest common subsequence between the model-generated response (X) and the reference text (Y).

FLAN-T5 achieved a ROUGE-L score of 0.792 (79.2%), indicating strong structural alignment with human-written responses.

4.3.3 Comparative Analysis with Other LLMs

We compared FLAN-T5 with BERT, RoBERTa, and ALBERT. Key findings:

- FLAN-T5 achieved a BLEU score of 52, making it the best-performing model in emotional text generation.
- BERT obtained a BLEU score of 26, indicating moderate performance.
- RoBERTa scored 22, struggling with emotion-driven text generation.
- ALBERT, which achieved a BLEU score of 23, is efficient but sacrifices some performance.

FLAN-T5 outperforms other models significantly in coherence, empathy, and grammatical accuracy.

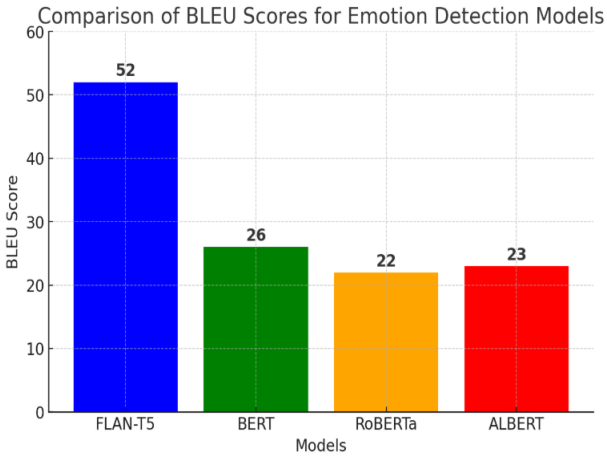


Figure 4.5: BLEU Score Comparison

4.3.4 Qualitative Assessment and Limitations

Beyond numerical evaluation, a manual review revealed additional insights:

Strengths

- FLAN-T5 adjusts response tone based on customer emotions.
- Responses are grammatically correct, contextually relevant, and professionally structured.
- High recall and fluency improve user experience in customer service.

Limitations

- Some responses are over-simplified, lacking multi-step resolutions.
- Struggles while handling happy emotions.
- The model struggles with highly technical queries.
- The dataset had only 2700 rows and took nearly 11 hours for fine-tuning.

4.3.5 Final Thoughts

FLAN-T5, fine-tuned for emotion-aware customer support, demonstrated:

- Strong lexical similarity (BLEU = 51.999)
- High structural coherence (ROUGE-L = 0.792)

Compared to BERT, RoBERTa, and ALBERT, FLAN-T5 excelled in text generation, making it the preferred model for automated customer service. Future improvements include:

- Enhancing domain-specific knowledge.

- Refining multi-step resolutions.
- Integrating reinforcement learning for better contextual understanding.

FLAN-T5's strengths in contextual response formulation and emotional nuance recognition make it a valuable asset for AI-driven customer service automation.

4.4 Parler - TTS

4.4.1 Performace Metrics

The evaluation of Parler TTS is essential to assess its effectiveness in generating emotionally expressive and natural synthetic speech. The following evaluation metrics are employed for both quantitative and qualitative assessment:

Mean Opinion Score (MOS)

The Mean Opinion Score (MOS) evaluates the perceived naturalness of synthesized speech. It is computed as the arithmetic mean of listener ratings:

$$MOS = \frac{1}{N} \sum_{i=1}^{N} r_i \tag{4.9}$$

where N is the total number of raters, and r_i represents the rating given by the i-th rater on a scale from 1 to 5.

Emotional Mean Opinion Score (E-MOS)

The Emotional Mean Opinion Score (E-MOS) measures the accuracy of emotional expression in synthesized speech. It is calculated similarly to MOS:

$$E\text{-}MOS = \frac{1}{N} \sum_{i=1}^{N} e_i \tag{4.10}$$

where e_i is the emotional accuracy rating given by the i-th rater, also rated on a scale from 1 to 5.

Word Error Rate (WER)

The Word Error Rate (WER) assesses intelligibility by computing the percentage of incorrectly transcribed words in the synthesized speech:

$$WER = \frac{S + D + I}{N} \times 100 \tag{4.11}$$

where:

- S is the number of substitutions,
- D is the number of deletions,
- I is the number of insertions,

- N is the total number of words in the reference transcript.

These evaluation metrics provide a comprehensive assessment of Parler TTS in terms of naturalness, emotional expressiveness, and intelligibility.

4.4.2 Analysis

In terms of Mean Opinion Score (MOS), which measures the naturalness of synthesized speech, the highest score is observed for "Disgust" (4.30), followed by "Angry" (4.20) and "Happy" (4.00). These results suggest that the TTS system effectively captures the naturalness of these emotions. On the other hand, "Frustrated" (3.40) and "Sad" (3.50) receive the lowest MOS, indicating that the system struggles to generate speech that sounds natural for these emotions.

When analyzing the Emotional Mean Opinion Score (EMOS), which evaluates how well the TTS system conveys emotions, "Disgust" again scores the highest (4.00), followed by "Angry" (3.90) and "Happy" (3.70). These findings suggest that these emotions are more convincingly expressed in the synthesized speech. However, "Sad" (2.80) and "Fearful" (2.90) receive the lowest EMOS, implying that these emotions are not effectively conveyed, making it harder for listeners to recognize them accurately.

The Word Error Rate (WER) analysis provides insight into the intelligibility of the synthesized speech. "Happy" (7.00) and "Angry" (8.00) have the lowest WER, meaning these emotions produce the least transcription errors and maintain high clarity. In contrast, "Fearful" (18.00) has the highest WER, suggesting significant intelligibility issues. Similarly, "Frustrated" (16.00) and "Neutral" (15.00) also exhibit high WER, indicating challenges in clear speech synthesis for these emotions.

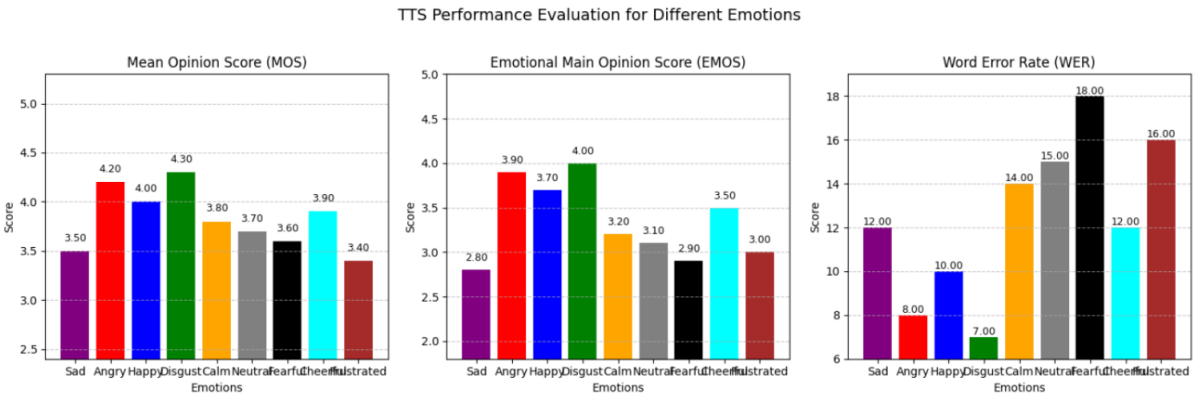


Figure 4.6: TTS Performace Evaluation

Overall, the TTS system performs best in terms of naturalness and emotional expressiveness for "Disgust," "Angry," and "Happy," while struggling with "Sad" and "Fearful" in both emotional recognition and intelligibility. The correlation between high WER and low EMOS scores suggests that speech clarity plays a crucial role in how well emotions are perceived. Future improvements should focus on enhancing

the synthesis quality for emotions with low MOS and EMOS while reducing WER to improve intelligibility.

Chapter 5

Conclusion

The exploration and development of an adaptive model for customer service, leveraging affective anthropomorphic intelligent systems, represents a significant stride forward in the pursuit of more human-centered AI interactions. This study has underscored the profound impact that integrating emotion detection into customer service AI can have on the overall quality of customer interactions. By utilizing a fusion of advanced AI models—including speech-to-text (STT), emotion detection, and large language models (LLMs)—the proposed system was able to accurately discern customer emotions and generate responses that not only addressed the content of inquiries but also the emotional undercurrents driving them. The evaluations highlighted the model's superior performance in emotion detection and response generation, particularly in comparison to traditional methods. The integration of affective computing principles within customer service AI models shows potential to redefine customer interactions by infusing empathy into automated responses. This empathy-driven approach stands to not only enhance customer satisfaction but also foster a deeper sense of trust between customers and AI systems.

Looking ahead, the implications of this work extend beyond customer service. The foundational technology developed here can be adapted for various applications requiring human-like empathy and understanding—from mental health support bots to educational tutors sensitive to student frustration or confusion. However, it is crucial to approach the deployment of such empathetic systems with a careful consideration of ethical implications, including transparency in AI decision-making and safeguarding against potential misuse of emotional data. Future developments could focus on enhancing the robustness of emotion detection models by integrating multimodal approaches that incorporate visual and physiological cues alongside audio data. Additionally, expanding the dataset to include more diverse linguistic and cultural expressions of emotions could improve the system's adaptability and accuracy. As AI continues to permeate daily life, the capacity to understand and respond to human emotions will be a defining factor in the acceptance and effectiveness of these technologies, paving the way for more intuitive and emotionally intelligent AI-driven interactions.

Bibliography

[1] C.-H. Wu and W.-B. Liang, "Emotion recognition of affective speech based on multiple classifiers using acoustic-prosodic information," *IEEE Transactions on Affective Computing*, vol. 2, no. 1, pp. 10–21, Jan. 2011.

[2] C.-N. Anagnostopoulos, T. Iliou, and I. Giannoukos, "Features and classifiers for emotion recognition from speech: A survey from 2000 to 2011," *Artificial Intelligence Review*, vol. 43, no. 2, pp. 155–177, 2012. DOI: 10.1007/s10462-012-9368-5.

[3] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura, "Speech synthesis based on hidden markov models," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1234–1252, 2013. DOI: 10.1109/JPROC.2013.2251852.

[4] S. Ö. Arık, M. Chrzanowski, A. Coates, *et al.*, "Deep voice: Real-time neural text-to-speech," 2017. [Online]. Available: https://arxiv.org/abs/1702.07825v2.

[5] S. Ghosh, E. Laksana, L.-P. Morency, and S. Scherer, "Learning representations of emotional speech with deep convolutional generative adversarial networks," in *Proc. Annu. Conf. International Speech Communication Association, INTERSPEECH 2017*, [Online]. Available: https://arxiv.org/abs/1705.02394, 2017, pp. 2741–2745.

[6] S. Mehri, K. Kumar, I. Gulrajani, *et al.*, "Samplernn: An unconditional end-to-end neural audio generation model," in *Published as a conference paper at ICLR 2017*, 2017. [Online]. Available: https://arxiv.org/abs/1612.07837v2.

[7] S. R. Livingstone and F. A. Russo, *The ryerson audio-visual database of emotional speech and song (ravdess)*, Accessed 31 Jan. 2025, 2018. [Online]. Available: https://www.kaggle.com/datasets/uwrfkaggler/ravdess-emotional-speech-audio.

[8] S. Karita, N. Chen, T. Hayashi, *et al.*, "A comparative study on transformer vs rnn in speech applications," *arXiv preprint arXiv:1909.06317v2*, 2019. [Online]. Available: https://arxiv.org/abs/1909.06317.

[9] E. Loeffler and J. Kreiman, *Crema-d: Crowd-sourced emotional multimodal actors dataset*, Accessed 31 Jan. 2025, 2019. [Online]. Available: https://www.kaggle.com/datasets/ejlok1/cremad.

[10] X. Liu, Y. Li, Z. Cheng, Z. Zhang, and Z. Ren, "Towards building an intelligent chatbot for customer service: Learning to respond at the appropriate time," in *Proc. 58th Annu. Meet. Association for Computational Linguistics (ACL 2020)*, 2020, pp. 5398–5407. DOI: 10.18653/v1/2020.acl-main.478.

[11] J.-H. Hsu, M.-H. Su, C.-H. Wu, and Y.-H. Chen, "Speech emotion recognition considering nonverbal vocalization in affective conversations," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 1675–1689, 2021. DOI: 10.1109/TASLP.2021.3076364.

[12] S. Nagdewani and A. Jain, *A review on methods for speech-to-text and text-to-speech conversion*, Inderprastha Engineering College, 2022.

[13] L. Nicolescu and M. T. Tudorache, "Human-computer interaction in customer service: The experience with ai chatbots—a systematic literature review," *Electronics*, vol. 11, no. 10, p. 1579, 2022. DOI: 10.3390/electronics11101579.

[14] A. Radford, J. W. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," *OpenAI Technical Report*, 2022. [Online]. Available: https://cdn.openai.com/papers/whisper.pdf.

[15] M. A. Mamun, H. M. Abdullah, M. G. R. Alam, M. M. Hassan, and M. Z. Uddin, "Affective social anthropomorphic intelligent system," *Multimedia Tools Appl.*, 2023. DOI: 10.1007/s11042-023-14597-6.

[16] Z. Yan, H. Liu, Z. Li, and S. Yang, "Multimodal emotion detection via attention-based fusion of extracted facial and speech features," *Sensors*, vol. 23, no. 5475, pp. 1–20, 2023. DOI: 10.3390/s23125475.

[17] C. et al., "Evaluating the llm agents for simulating humanoid behavior," in *Proc. 1st HEAL Workshop at CHI Conf. Human Factors in Computing Systems*, Honolulu, HI, USA, May 2024.

[18] M. Z. Amini, M. Tokgoz, N. Gurbuz, M. Tuerker, and S. Keung, "Exploring large language models' emotion detection abilities: Use cases from the murder of zhina mahsa amini and the turkey-syria earthquake," 2024.

[19] S. Kumar and R. Chaturvedi, "Evaluating the efficacy of distilled transformer models for sentiment analysis in financial texts: A comparative study," *International Journal of Current Science (IJCSPUB)*, vol. 14, no. 2, pp. 926–934, Jun. 2024. [Online]. Available: www.ijcspub.org.

[20] Z. Liu, K. Yang, Q. Xie, T. Zhang, and S. Ananiadou, "Emollms: A series of emotional large language models and annotation tools for comprehensive affective analysis," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24)*, ACM, Aug. 2024, pp. 5487–5495. DOI: 10.1145/3637528.3671552. [Online]. Available: https://doi.org/10.1145/3637528.3671552.

[21] N. Mughal, G. Mujtaba, S. Shaikh, A. Kumar, and S. M. Daudpota, "Comparative analysis of deep natural networks and large language models for aspect-based sentiment analysis," *IEEE Access*, vol. 12, pp. 60 943–60 944, 2024. DOI: 10.1109/ACCESS.2024.3386969.

[22] R. Valle, K. Shih, R. Prenger, and B. Catanzaro, *Flowtron: An autoregressive flow-based generative network for text-to-speech synthesis*, [Online]. Available: https://github.com/NVIDIA/flowtron, 2024.

[23] Suraj520, *Customer support ticket dataset*, Accessed: February 2, 2025, 2025. [Online]. Available: https://www.kaggle.com/datasets/suraj520/customer-support-ticket-dataset.