Mohamed Khider University of Biskra
Faculty of Exact Sciences
Department of Computer Science

MASTER THESIS

Speciality : Master - Artificial Intelligence - Ref : P1600/2026

TITLE

Call Transcription: Customer Sentiment Analysis and Conversation Summarization

Prepared and defended by

Ilef-doriane LOUAFI

Defense date : 09/06/2026

DEFENSE COMMITTEE

Full Name	Grade	Role
Zohra HMIDI	MCB	**President**
Nouer El Houda ZERARKA	MCB	**Supervisor**
Sarra Babahenini	MCB	**Examiner**
Chahra ADISSA	Prof	**Incubator Representative**

Academic Year : 2025/2026

Dedications and Acknowledgments

First and foremost, I would like to express my deepest gratitude to Allah for giving me the strength, patience, and determination to complete this work.

I would like to express my deepest gratitude and sincere appreciation to my supervisor, Dr. Nourelhouda Zerarka, for her continuous guidance, support, and encouragement throughout the preparation of this work. Her valuable advice, constructive feedback, and patience have greatly helped me improve my project. I am truly thankful for her time, dedication, and for the knowledge she shared with me during this journey.

I would also like to warmly thank Dr. Djihane Houfani for her valuable help in improving the academic quality of my work. Her remarks and advice helped me better understand how to organize my ideas, write more academically, and present my research in a clearer and more professional way.

I would also like to extend my sincere thanks and appreciation to the honorable president and members of the jury for accepting to examine and evaluate my work. Their time, attention, and valuable remarks are highly appreciated.

My heartfelt thanks go to my family, especially my parents, and my grandfather, for their endless love, support, and sacrifices. Their encouragement has always been my greatest motivation.

I am also grateful to my teachers and professors for the knowledge and experience they shared with me during my academic journey.

Finally, I would like to thank my friends and everyone who supported me, directly or indirectly, during this work. Your help, encouragement, and kind words meant a lot to me.

Thank you all.

Abstract

Customer service plays an important role in maintaining the relationship between companies and their clients. With the increasing volume of customer interactions, especially through phone-based support, manually reviewing and analyzing every call has become difficult and time-consuming. These conversations contain valuable information about customer problems, service quality, and emotional state, but this information often remains hidden inside raw audio recordings.

This thesis proposes a system that uses Artificial Intelligence, Machine Learning, and Deep Learning techniques to analyze customer service calls automatically. The system is composed of three main modules: speech transcription, speech-based sentiment analysis, and dialogue summarization. The transcription module converts spoken audio into written text, the sentiment analysis module detects the dominant emotional state from the speech signal, and the summarization module generates a concise summary of the interaction.

The proposed system was developed and evaluated using several datasets and models adapted to each task. Emotional speech datasets were used for the sentiment analysis module, speech transcription data was used to evaluate the transcription component, and dialogue and customer service datasets were used to fine-tune the summarization model. The different modules were then integrated into a desktop application that allows users to process live or recorded calls and view the generated transcript, summary, and sentiment result.

By combining these components, the system aims to transform raw customer service audio into structured and useful information. This can assist agents and supervisors in reviewing calls more efficiently, understanding customer needs, identifying emotional signals, and improving follow-up actions. Overall, the proposed work provides a practical foundation for intelligent customer service call analysis and demonstrates the potential of AI-based tools in improving customer support operations.

الملخص

تلعب خدمة العملاء دورا مهما في الحفاظ على العلاقة بين الشركات وعملائها، خاصة مع تزايد حجم التفاعلات اليومية عبر المكالمات الهاتفية ومنصات الدعم المختلفة. وبسبب هذا الحجم الكبير من المكالمات، أصبح تحليل كل تفاعل يدويا عملية صعبة وتستغرق وقتا طويلا. تحتوي هذه المكالمات على معلومات مهمة حول مشاكل العملاء، جودة الخدمة، والحالة العاطفية للمتصل، إلا أن هذه المعلومات غالبا ما تبقى غير مستغلة داخل التسجيلات الصوتية الخام.

تقترح هذه المذكرة نظاما يعتمد على تقنيات الذكاء الاصطناعي، التعلم الآلي، والتعلم العميق من أجل تحليل مكالمات خدمة العملاء بطريقة آلية. يتكون النظام من ثلاث وحدات رئيسية: وحدة تحويل الكلام إلى نص، وحدة تحليل المشاعر انطلاقا من الإشارة الصوتية، ووحدة تلخيص الحوار. تقوم وحدة التفريغ الصوتي بتحويل الكلام المنطوق إلى نص مكتوب، بينما تعمل وحدة تحليل المشاعر على تقدير الحالة العاطفية الغالبة من صوت العميل. أما وحدة التلخيص فتقوم بإنتاج ملخص قصير ومفيد لمحتوى المكالمة.

تم تطوير النظام وتقييمه باستخدام عدة مجموعات بيانات ونماذج مناسبة لكل مهمة. استعملت قواعد بيانات الكلام العاطفي لتطوير وحدة تحليل المشاعر، واستعملت بيانات التعرف الآلي على الكلام لتقييم وحدة التفريغ الصوتي، كما استعملت مجموعات بيانات الحوارات وخدمة العملاء لتدريب نموذج التلخيص. بعد ذلك، تم دمج هذه الوحدات داخل تطبيق مكتبي يسمح بمعالجة المكالمات المباشرة أو التسجيلات الصوتية، ثم عرض النص المستخرج، الملخص، ونتيجة المشاعر.

يهدف النظام المقترح إلى تحويل مكالمات خدمة العملاء من تسجيلات صوتية غير منظمة إلى معلومات واضحة ومنظمة يمكن الاستفادة منها بسهولة. يمكن أن يساعد هذا النظام الأعوان والمشرفين على مراجعة المكالمات بشكل أسرع، فهم احتياجات العملاء، ملاحظة الإشارات العاطفية، وتحسين المتابعة وجودة الخدمة. وبذلك، توفر هذه المذكرة أساسا عمليا لنظام ذكي لتحليل مكالمات خدمة العملاء، وتبرز إمكانية استخدام أدوات الذكاء الاصطناعي لدعم وتحسين عمليات خدمة العملاء.

الكلمات المفتاحية: خدمة العملاء، الذكاء الاصطناعي، تحويل الكلام إلى نص، تحليل المشاعر الصوتية، تلخيص الحوار، التعلم العميق.

Glossary and List of Abbreviations

Term	Definition
ML	Machine Learning; a field of AI where systems learn patterns from data in order to make predictions or decisions.
DL	Deep Learning; a subset of machine learning based on neural networks with multiple layers.
ASR	Automatic Speech Recognition; the process of converting spoken language into written text.
SER	Speech Emotion Recognition; the task of identifying emotional states from speech signals using acoustic and prosodic information.
MFCC	Mel-Frequency Cepstral Coefficients; acoustic features used to represent the spectral characteristics of speech.
ZCR	Zero-Crossing Rate; an audio feature that measures how often a signal crosses the zero-amplitude axis within a frame.
WER	Word Error Rate; an evaluation metric for speech recognition that measures word-level errors between a predicted transcript and a reference transcript.
CER	Character Error Rate; an evaluation metric for speech recognition that measures character-level errors between a predicted transcript and a reference transcript.

Term	Definition
CNN	Convolutional Neural Network; a neural network architecture commonly used to extract local patterns from structured data such as images, audio features, or spectrograms.
LSTM	Long Short-Term Memory; a recurrent neural network architecture designed to model sequential data and long-term dependencies.
Capsule Network	A neural network architecture that represents features as vectors and aims to preserve spatial or structural relationships between them.
FLAN-T5	An instruction-tuned version of the T5 text-to-text transformer model, used for tasks such as summarization and text generation.
BART	Bidirectional and Auto-Regressive Transformer; a sequence-to-sequence transformer model used for text generation and summarization.
CRM	Customer Relationship Management; systems or tools used by organizations to manage customer interactions and related information.
GUI	Graphical User Interface; the visual interface through which users interact with an application.

List of Tables

Table **Page**

1.1 Main acoustic features used in speech-based sentiment analysis 18
1.2 Comparison of State-of-the-Art SER Models and Their Dataset Focus . 23

2.1 Comparison of State-of-the-Art ASR Models and Their Characteristics 28
2.2 Comparison of State-of-the-Art Text Summarization Models 38

3.1 Datasets considered for the speech-based sentiment analysis module . . 44
3.2 Datasets considered for the speech transcription module 49
3.3 Data sources used for the text summarization module 54

4.1 Approximate train-test split for the main sentiment datasets 68
4.2 Phase 1 results using the paper-based MA-CapsNet-DA architecture on clean data 74
4.3 Classification report for the CASIA + Emo-DB clean test set 75
4.4 Phase 2 architecture comparison on the Emo-DB + SAVEE clean test set 76
4.5 Phase 3 performance of the proposed enhanced architecture 77
4.6 Classification report of the proposed enhanced architecture on the clean test set 78
4.7 Comparison between the paper results and the reproduced results on clean multi-dataset configurations 79
4.8 Overall ASR evaluation results on Common Voice 88
4.9 ASR results by accent on Common Voice development and test subsets 88
4.10 Data split used for the text summarization module 93
4.11 Tokenization configuration for the text summarization module 95

4.12 Learning configuration of the text summarization module 96
4.13 Summarization results across the fine-tuning stages 98
4.14 Final call transcript summarization setup 99
4.15 Final test results of the call transcript summarization model 99
4.16 Generation statistics of the final summarization model 100
4.17 Main architectural layers of the desktop application 106
4.18 Audio capture configuration used by the desktop application 107
4.19 Integrated AI modules in the desktop application 109
4.20 Stored data for completed call sessions 111
4.21 Main pages of the desktop application 113

List of Figures

Figure **Page**

1.1 Multichannel ecosystem of customer service [1] 5
1.2 Evolution of customer service [2] 6
1.3 PBX phone system [3] 7
1.4 Relationship between Artificial Intelligence, Machine Learning, Deep Learning, Natural Language Processing, and Large Language Models [4] 10
1.5 The different uses of NLP [5] 11
1.6 The outputs of sentiment analysis [6] 12
1.7 Multi-modal sentiment analysis examples [7] 14
1.8 The process of sentiment analysis [8] 16

2.1 Example speech processing pipeline [9] 26
2.2 Dialogue summary example [10] 30
2.3 The difference between extractive and abstractive summarization [11] . 35

3.1 Data flow 40
3.2 Use case diagram of the customer service call analysis system 41
3.3 Architecture of the sentiment analysis model 47
3.4 Flan T5 workflow [12] 56

4.1 Data discovery and splitting for the sentiment analysis model 69
4.2 Feature extraction 70
4.3 Model performance at different SNR 78
4.4 Performance of the proposed sentiment analysis model 81
4.5 Data mapping for the transcription model 83
4.6 WER Implementation 85

4.7 Transcription Model Evaluation Implemetation 87
4.8 ASR evaluation results on different accents 89
4.9 Length of the 92k Real-World Call Center Scripts 92
4.10 Data splitting implementation for the summarization model 93
4.11 Data filtering and cleaning for the summarization model 94
4.12 Tokenizer Loading 94
4.13 Model loading for the summarization model 95
4.14 ROUGE metrics 96
4.15 Fine tuning results for the summarization model 101
4.16 Example summary 102
4.17 Application Dashboard and Settings 103
4.18 Class diagram of the desktop application implementation 104
4.19 Component diagram of the desktop application architecture 105
4.20 Application Live Call Analysis 107
4.21 Sequence diagram of the live call processing flow 108
4.22 Application Call History Page 110
4.23 Database structure used for call history storage 111
4.24 Application Call Processing 112
4.25 Customer and Agent display during a call 113

Table of Contents

Dedications and Acknowledgments i

Abstract ii

Glossary and List of Abbreviations iv

List of Tables vi

List of Figures viii

General Introduction 1

1 Foundational Concepts and Sentiment Analysis in Customer Service 4

1.1 Introduction 4

1.2 Customer Service 5

1.2.1 Evolution of Customer Interaction 6

1.2.2 Call Centers 7

1.2.3 The Role of AI in Enhancing Customer Experience 8

1.3 Fundamentals of Artificial Intelligence and Natural Language Processing 9

1.3.1 Artificial Intelligence 10

1.3.2 Natural Language Processing 11

1.4 Sentiment Analysis 12

1.4.1 Core Concepts 12

1.4.2 Modalities of Sentiment Analysis 13
1.4.3 Domain Applications 14
1.5 Speech Sentiment Analysis 15
1.5.1 Foundational Pillars of Speech Analysis 17
1.5.2 Importance of Sentiment Detection from Speech 19
1.5.3 Importance of Speech in Customer Service 20
1.5.4 Technical Challenges 20
1.5.5 Current Approaches and Methodologies in Sentiment Analysis 22
1.6 Conclusion 23

2 Speech Transcription and Text Summarization in Customer Service 24
2.1 Introduction 24
2.2 Speech Transcription 25
2.2.1 Current Approaches and Methodologies in Speech Transcription 27
2.3 Text Summarization 28
2.3.1 Fundamentals of Dialogue Summarization 29
2.3.2 Applications in Different Domains 31
2.3.3 Extractive Summarization Approach 32
2.3.4 Abstractive Summarization Approach 33
2.3.5 Applications in Customer Service Operations 35
2.3.6 Key Challenges in Customer Service Contexts 36
2.3.7 Current Approaches and Methodologies in Text Summarization 36
2.4 Conclusion 38

3 System Design and Methodology 39
3.1 Introduction 39
3.2 System Overview 39
3.3 Sentiment Analysis Model 42
3.3.1 Datasets 42
3.3.2 Model Selection 44
3.3.3 Feature Extraction And Processing 45
3.3.4 Architecture 46

3.4 Speech Transcription Model 48
3.4.1 Datasets 48
3.4.2 Model Selection 50
3.4.3 Transcription Workflow 51
3.4.4 Architecture 52
3.5 Dialogue Summarization Model 53
3.5.1 Datasets 53
3.5.2 Model Selection 55
3.5.3 Summarization Workflow 55
3.5.4 Architecture 57
3.6 Conclusion 58

4 Implementation and Evaluation 59
4.1 Introduction 59
4.2 Development Environment and Tools 59
4.2.1 Hardware and Execution Environment 60
4.2.2 Kaggle 61
4.2.3 Python 61
4.2.4 Deep Learning Frameworks 62
4.2.5 Hugging Face Ecosystem 63
4.2.6 Librosa 63
4.2.7 Data Processing and Evaluation Libraries 64
4.2.8 Desktop Application Development 65
4.3 Implementation and Evaluation of the Sentiment Analysis Module . 66
4.3.1 Dataset Description 67
4.3.2 Data Splitting 68
4.3.3 Preprocessing Phase 69
4.3.4 Feature Extraction Phase 70
4.3.5 Model Learning 71
4.3.6 Evaluation Metrics 72
4.3.7 Experiments and Results 74
4.3.8 Comparison and Discussion 79
4.4 Implementation and Evaluation of the Speech Transcription Module 82

4.4.1 Dataset Description 82
4.4.2 Test Data Selection 82
4.4.3 Preprocessing Phase 83
4.4.4 Model Loading and Input Representation 84
4.4.5 Evaluation Metrics 84
4.4.6 Experiments and Results 85
4.4.7 Discussion 89
4.5 Implementation and Evaluation of the Text Summarization Module 90
4.5.1 Dataset Description 91
4.5.2 Data Splitting 92
4.5.3 Preprocessing Phase 93
4.5.4 Tokenization Phase 94
4.5.5 Model Learning 95
4.5.6 Evaluation Metrics 97
4.5.7 Experiments and Results 98
4.5.8 Discussion 100
4.6 System Interface and Deployment 102
4.6.1 Desktop Application Architecture 104
4.6.2 Audio Capture and Processing Flow 106
4.6.3 Model Integration 108
4.6.4 Data Storage and Call History 109
4.6.5 User Interface and Final Output Display 112
4.7 Conclusion 113

Bibliography **118**

General Introduction

With the rapid growth of online commerce and digital services, online customer support has become an essential component of modern business operations. While this model offers significant benefits in terms of accessibility and scalability, it has also led to a substantial increase in demand that traditional infrastructures often struggle to manage. This challenge is further amplified by the complexity of human communication, where intent and emotion are frequently expressed in fragmented or ambiguous forms. As a result, evaluating the efficiency of customer service has become a key factor in maintaining competitiveness.

Customer service interactions differ from ordinary text or speech data due to their high-stakes and problem-oriented nature. Customers express needs, frustrations, and urgencies, often using emotionally charged language. Effective handling of these interactions therefore requires the simultaneous understanding of both intent and sentiment. These exchanges generate large volumes of unstructured data, including text and audio, which can provide valuable insights if properly processed. Artificial intelligence offers promising solutions to exploit this information; however, existing systems still face limitations when dealing with sarcasm, multi-issue requests, and subtle emotional cues.

Manual analysis of such data is not feasible at scale, which highlights the need for AI-based solutions. A central technical challenge lies in balancing batch processing, suitable for retrospective analysis, with real-time processing, which enables proactive support during live interactions. Real-time systems demand models that are not only accurate but also computationally efficient, making them a decisive factor in the future of customer service automation.

Ultimately, customer satisfaction remains one of the most important yet challenging objectives for businesses. Organizations must adopt both proactive and

reactive strategies by anticipating recurring issues and responding effectively in real time. The ability to identify patterns in historical interactions and assess customer sentiment during conversations can significantly enhance service quality. Since satisfied customers are more likely to remain loyal, the strategic importance of intelligent customer service systems is evident.

The main objective of this work is to design and implement an intelligent system for customer service call analysis. The proposed system aims to transform raw call audio into useful information by generating a transcript, producing a summary, and detecting the dominant emotional state expressed in the customer's speech. This allows the interaction to be reviewed more easily and helps make the information contained in the call more accessible.

This study also aims to combine several artificial intelligence techniques in one practical workflow. The speech transcription component converts spoken language into written text. The sentiment analysis component analyzes the audio signal in order to estimate the customer's emotional state. The summarization component uses the generated transcript to produce a shorter representation of the conversation. These three components are then integrated into a desktop application that supports live call processing, recorded file processing, result display, and call history storage.

Through this work, the goal is not to replace human agents or supervisors, but to support them by reducing the amount of manual review required after customer service calls. By providing structured outputs from audio conversations, the system can help agents and supervisors understand customer needs, identify emotional signals, and review previous interactions more efficiently.

This thesis is organized into four main chapters. The first chapter presents the foundational concepts related to customer service, Artificial Intelligence, Natural Language Processing, sentiment analysis, and speech-based sentiment analysis. It explains the importance of customer service interactions and highlights the role of speech in understanding customer emotions.

The second chapter focuses on speech transcription and text summarization. It presents Automatic Speech Recognition as the process of converting spoken audio into written text, then discusses text summarization as a way to reduce long transcripts into concise and useful summaries. This chapter also presents the main challenges and current approaches related to these two tasks.

The third chapter presents the design and methodology of the proposed system. It explains the general workflow of the application and describes the three main modules: the sentiment analysis model, the speech transcription model, and the dialogue summarization model. It also discusses the datasets, model choices, processing steps, and overall architecture used in the proposed solution.

The fourth chapter presents the implementation and evaluation of the system. It describes the development environment and the tools used during the experiments. It then evaluates the three modules separately using suitable metrics, before presenting the final desktop application, its interface, processing flow, model integration, data storage, and call history features.

Finally, this work ends with a general conclusion that summarizes the main contributions of the work, discusses its limitations, and presents possible directions for future improvement.

Chapter 1

Foundational Concepts and Sentiment Analysis in Customer Service

1.1 Introduction

Customer service has become a major source of operational and strategic information for modern organizations. Through phone calls, emails, chats, and digital platforms, companies receive large volumes of customer interactions that contain both factual information and emotional signals. However, the scale and complexity of these interactions make manual analysis difficult, especially when the objective is to understand not only what customers say, but also how they feel and how their issues evolve during the interaction.

This chapter presents the foundational concepts needed to understand the context of the proposed work. It also discusses sentiment analysis, with a particular focus on speech-based sentiment analysis. Since the proposed system analyzes customer service calls, special attention is given to the emotional information carried by speech signals. These concepts provide the background needed for the system design and methodology presented in the following chapters.

1.2 Customer Service

Customer service is a critical and complex interface where businesses directly manage client relationships and gather real-time insights. It has evolved into a multichannel ecosystem phone, chat, and email as shown in Figure 1.1. Each generating large volumes of rich conversational data. The main challenge lies in managing this scale while maintaining quality, as traditional metrics often fail to capture the full narrative of customer sentiment and the underlying issues within each exchange [13].



Figure 1.1: Multichannel ecosystem of customer service [1]

The sheer volume of interactions makes manual analysis impractical, creating a strategic need for technological augmentation. The goal is to transform raw dialogues from a logistical burden into a structured asset for organizational learning and improvement. In this context, sentiment analysis plays an important role because it helps decode the emotional tone of customer interactions and provides a more complete picture of the customer experience.

1.2.1 Evolution of Customer Interaction

The historical progression of customer interaction as shown in Figure 1.2 reflects a shift from simple transactional exchanges to complex, data-driven relationships. The conceptual origins of customer service emerged when businesses recognized that service quality could serve as a primary differentiator in competitive markets. Early customer service was characterized by localized, personal interactions in which providers maintained direct relationships with clients, treating service not merely as a task but as a foundational business concept [14].

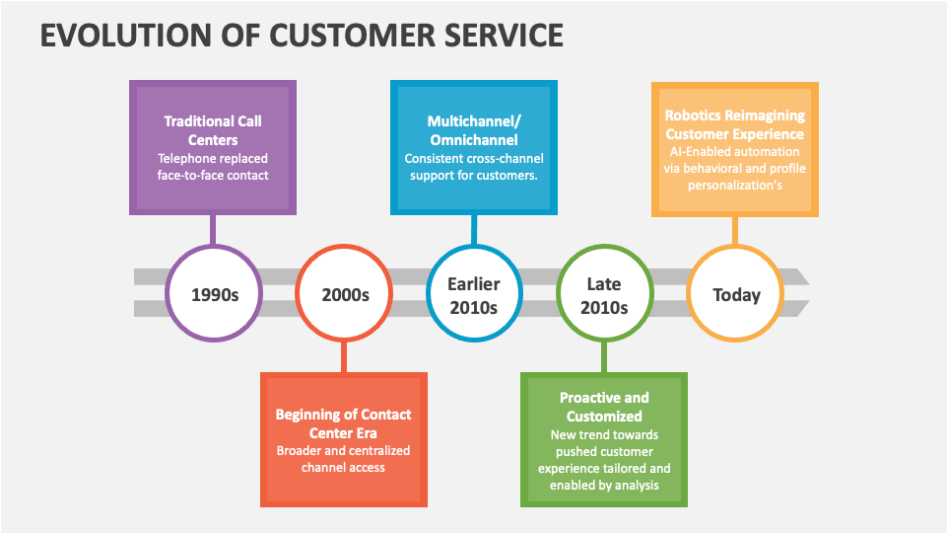


Figure 1.2: Evolution of customer service [2]

As commerce expanded, the Telephony Era introduced the first major technological shift in interaction volume. This period saw the professionalization of customer care as a distinct business function. Service centers evolved from secondary support units into critical hubs for maintaining customer loyalty through centralized telephony systems [15]. However, this increase in scale introduced a significant logistical burden, often at the expense of the personalized touch found in earlier eras.

The digital age further fragmented the interaction landscape, introducing multichannel environments such as email and live chat. This complexity required more sophisticated technological frameworks to manage the resulting data explosion.

Modern customer service has now entered an AI-driven phase, where the focus has shifted from simply providing communication channels to leveraging advanced architectures capable of interpreting those communications [16].

This current era emphasizes moving beyond basic automation toward intelligent systems capable of processing high volumes of conversational data in real time. It creates a strategic requirement for integrating specialized models, particularly those capable of decoding emotional subtext from speech and supporting more responsive customer service operations.

1.2.2 Call Centers

The call center serves as the primary centralized infrastructure for handling large volumes of telephonic requests, acting as the frontline of organizational communication. Historically, call center technology evolved from manual switchboards to complex automated environments. As shown in Figure 1.3, a major milestone was the implementation of Private Branch Exchange (PBX) systems and Automatic Call Distributors (ACD), which enabled efficient call routing based on agent availability and operational requirements [17].

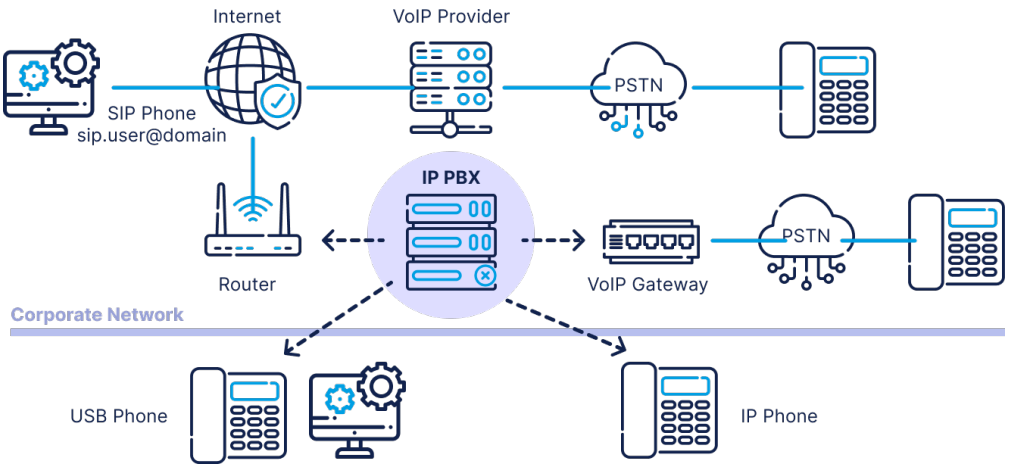


Figure 1.3: PBX phone system [3]

The technological trajectory of call centers has been marked by a transition from traditional telephony to advanced digital infrastructures integrating Computer Telephony Integration (CTI) and Interactive Voice Response (IVR). These

technologies transformed call centers from reactive service units into structured environments capable of collecting and managing large volumes of interaction data [18]. The adoption of Voice over IP (VoIP) and cloud-based solutions further enhanced scalability, enabling the recording, storage, and analysis of customer interactions to support advanced analytics and performance monitoring [19].

Today, call centers are no longer viewed solely as cost centers but as strategic assets capable of generating valuable business intelligence. The large-scale data produced through customer interactions—often extensive audio recordings—necessitates automated transcription and sentiment analysis systems. These technologies allow organizations to extract actionable insights, improving service quality and decision-making processes [16].

1.2.3 The Role of AI in Enhancing Customer Experience

The integration of Artificial Intelligence (AI) into customer service has changed the way organizations manage and improve customer experience. Instead of relying only on human agents and manual analysis, AI allows companies to automate repetitive tasks, provide faster responses, and process large volumes of customer data more efficiently [20].

AI-based systems can support customer service by using techniques such as Machine Learning (ML) and Natural Language Processing (NLP) to identify patterns in customer behavior, detect sentiment, and extract useful information from conversations [16]. This helps organizations move from reactive support, where issues are handled only after they appear, toward more proactive service strategies.

However, the use of AI also introduces challenges, especially the need for reliable data and the difficulty of correctly interpreting emotional nuance in automated systems. In this thesis, speech-based sentiment recognition is considered a key AI-based tool for improving customer experience, as it helps systems understand emotional tone directly from spoken interactions [21].

1.3 Fundamentals of Artificial Intelligence and Natural Language Processing

Artificial Intelligence (AI) and Natural Language Processing (NLP) are two important foundations of modern intelligent systems. AI focuses on designing systems capable of performing tasks that usually require human intelligence, such as learning, reasoning, problem-solving, and decision-making [22]. Over time, AI has moved from theoretical and rule-based approaches toward practical systems widely used in different domains [**?**].

NLP is a subfield of AI concerned with the automatic processing of human language. It enables machines to analyze, understand, and generate language in both written and spoken forms [23]. By combining linguistic knowledge with statistical and machine learning methods, NLP supports many applications, including sentiment analysis, speech recognition, and automatic summarization [24].

The relationship between AI, Machine Learning, Deep Learning, NLP, and Large Language Models is illustrated in Figure 1.4. As shown in the figure, NLP is closely connected to AI and overlaps with learning-based approaches, since many modern language systems rely on machine learning and deep learning techniques.

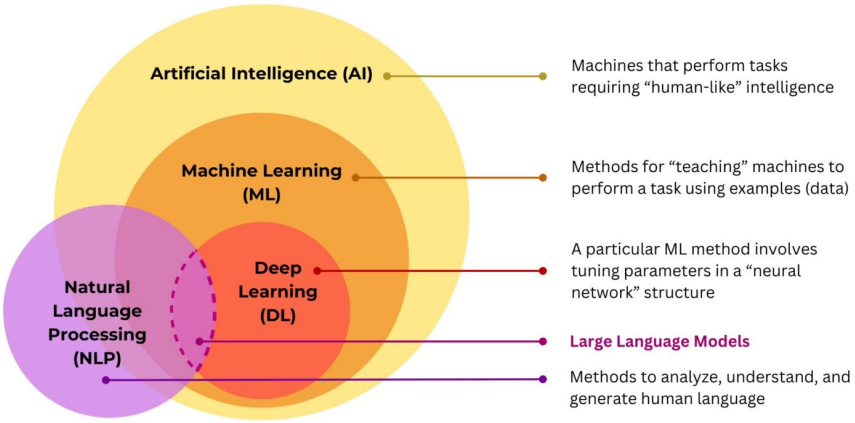


Figure 1.4: Relationship between Artificial Intelligence, Machine Learning, Deep Learning, Natural Language Processing, and Large Language Models [4]

Together, AI and NLP provide the technical basis for transforming unstructured communication data into structured and useful information. This is especially important in customer service, where interactions often contain both factual content and emotional signals.

1.3.1 Artificial Intelligence

Artificial Intelligence (AI) can be generally defined as the field that studies and develops systems capable of performing tasks normally associated with human intelligence. These tasks include learning from data, reasoning, solving problems, and making decisions [22]. In practice, AI aims to help machines process information and act in ways that support or automate human activities.

The field has evolved from early rule-based systems, which depended on pre-defined instructions, to machine learning approaches that allow systems to learn patterns directly from data [**?**]. More recent progress in deep learning has made AI effective in areas such as image recognition, speech processing, and natural language understanding.

In customer service, AI is useful because it can process large amounts of textual and audio data, detect patterns, and support decision-making. Most current AI applications are examples of narrow AI, meaning they are designed for specific tasks such as sentiment classification, transcription, or summarization.

1.3.2 Natural Language Processing

Natural Language Processing (NLP) is a branch of AI that focuses on enabling machines to process and understand human language [23]. Since language is often ambiguous, context-dependent, and informal. Figure 1.5 demonstrates NLP's capabilities for converting raw text or speech into information that machines can analyze.

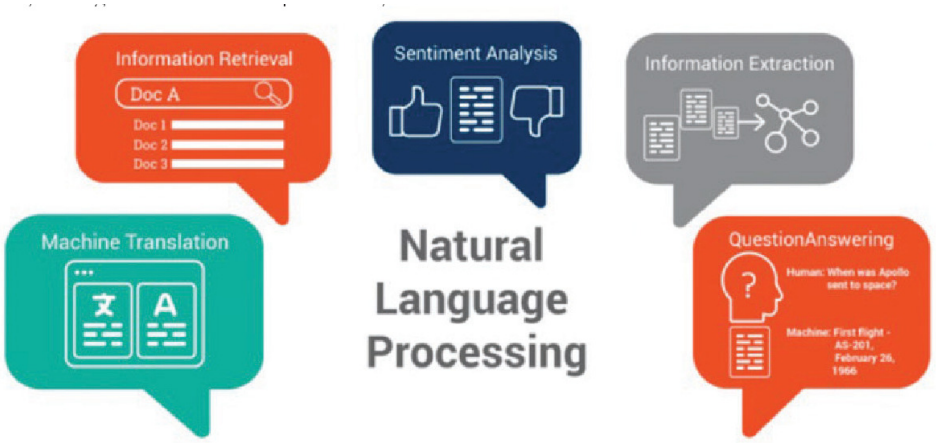


Figure 1.5: The different uses of NLP [5]

Early NLP systems relied mainly on handcrafted linguistic rules, but modern approaches increasingly use statistical methods and deep learning. These methods allow models to learn from large amounts of language data and capture meaning, context, and relationships between words or sentences [24].

NLP is used in many tasks, including text classification, sentiment analysis, machine translation, speech recognition, and summarization. In customer service, it helps extract useful information from conversations, identify customer issues, detect emotional tone, and generate structured outputs that support service improvement.

1.4 Sentiment Analysis

Sentiment analysis also known as "opinion mining" or "sentiment classification" is the computational identification and classification of opinions [25]. Its fundamental goal is to automatically determine the emotional tone behind data, whether expressed through text, speech, visual cues, or user behavior, usually resulting in the outputs shown in Figure 1.6, enabling systems to interpret subjective information at scale [21].



Figure 1.6: The outputs of sentiment analysis [6]

This process goes beyond identifying positive or negative words. It requires capturing the full spectrum of human expression, including subtle emotional states such as skepticism, urgency, disappointment, or satisfaction. Sentiment analysis must account for contextual dependencies, sarcasm, ambiguity, and implicit sentiment [26]. Moreover, there is often a gap between expressed language and underlying emotion, further complicating interpretation.

The value of sentiment analysis lies in its ability to transform unstructured data into meaningful insights about human attitudes and perceptions. By moving from descriptive to interpretive analysis, it supports applications such as customer feedback analysis, behavioral prediction, and decision-making, enabling systems to respond more effectively [27].

1.4.1 Core Concepts

The core concept of sentiment analysis is to systematically transform subjective expression into objective understanding. Behind every statement or review lies

a measurable emotional stance. The goal is not merely to read words but to computationally interpret the opinion or feeling they convey [25].

The process begins with identifying where an opinion is expressed. Once identified, the next step is classification—assigning the expression to a defined category. The simplest form is the classic positive, negative, or neutral classification. More nuanced models represent sentiment along continuous dimensions such as valence, meaning pleasantness, and arousal, meaning intensity [28].

A major advancement is the shift from global to targeted analysis. Instead of labeling an entire text with a single sentiment, the field distinguishes between document-level, sentence-level, and aspect-level sentiment analysis. This allows finer granularity: a customer may express negative sentiment about price while being positive about usability. Capturing such distinctions is essential for generating precise insights [29].

Ultimately, these concepts serve a strategic purpose: converting vast amounts of unstructured, emotionally rich data into structured information. This enables pattern identification, opinion measurement, and detection of satisfaction drivers, supporting informed decision-making [27].

1.4.2 Modalities of Sentiment Analysis

The methods and challenges of sentiment analysis vary significantly depending on the medium, as each modality encodes emotional information in a different way [21].

When dealing with text, such as emails, reviews, or social media posts, detection focuses on language. It analyzes lexical choices, syntactic structures, and contextual relationships. However, this approach must take into account linguistic phenomena such as sarcasm, irony, and cultural variation [26].

With audio or speech, detection shifts to the human voice. In this case, the focus is how emotion is expressed. Cues are embedded in acoustic and prosodic features, which often reveal unspoken content. A major challenge lies in isolating

them from background noise and speaker rhythm [30].

Visual media relies on the interpretation of non-verbal cues. This includes analyzing facial expressions, body posture, and gestures. In video, this is often combined with audio and textual information, forming a multimodal representation. The primary difficulty lies in the variability and subjectivity of visual expressions [31].

The Figure 1.7 show two examples of sentiment analysis, where different modalities may reflect different emotional information [7].

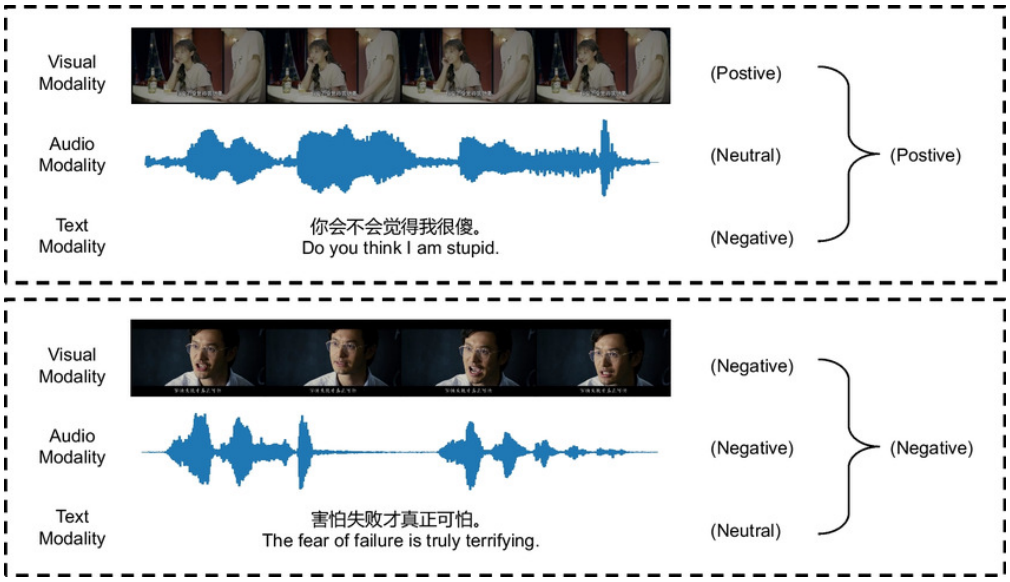


Figure 1.7: Multi-modal sentiment analysis examples [7]

Despite these differences, modern approaches across all modalities increasingly rely on machine learning techniques. These systems are trained on large annotated datasets, enabling them to generalize to new, unseen data [27].

1.4.3 Domain Applications

Sentiment analysis has powerful applications across diverse fields, transforming subjective opinions into strategic assets.

1. **Market Research and Product Development.** Sentiment analysis acts as a large-scale, real-time focus group. By analyzing reviews, social media discussions, and surveys, companies understand why products succeed or fail. It highlights desired features, pain points, and emerging trends, enabling data-driven innovation [32].

2. **Political and Public Sector Analysis.** Governments and organizations use sentiment analysis to gauge public opinion and monitor social stability. By assessing sentiment in news, social media, and public forums, they obtain a quantitative pulse on public mood, providing feedback on policy reception and early warnings of opinion shifts [33].

3. **Brand and Reputation Management.** Sentiment analysis functions as an early detection radar for brand perception. It distinguishes between neutral mentions, praise, and criticism, enabling proactive reputation management and rapid response to crises [34].

4. **Customer Service and Experience.** In customer service, sentiment analysis is a diagnostic and operational tool. It enables real-time prioritization by detecting highly negative sentiment and escalates cases accordingly. It also transforms large volumes of interactions into structured insights about recurring issues and pain points, supporting holistic relationship management [35].

Across these domains, sentiment analysis converts emotional data into actionable intelligence, enabling organizations to be more responsive and aligned with user needs.

1.5 Speech Sentiment Analysis

Speech Sentiment Analysis (SSA) focuses on the domain of spoken language as shown in Figure 1.8, aiming to identify and interpret emotional states directly from audio signals. Unlike text-based approaches, which rely mainly on lexical and syntactic information, SSA incorporates additional layers of information embedded

within speech, such as tone, pitch, rhythm, and intensity. These cues often reveal nuances that may not be explicitly expressed through words alone [36].

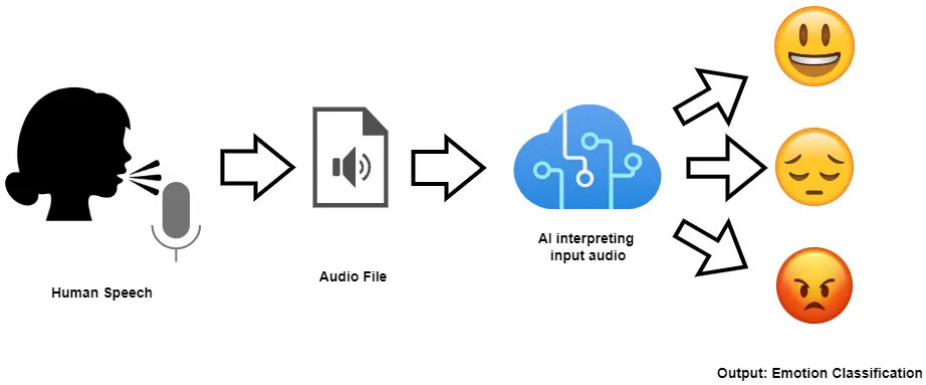


Figure 1.8: The process of sentiment analysis [8]

The transition from text to speech analysis introduces a significant increase in complexity. Spoken language is inherently more dynamic and less structured than written text, often containing noise, interruptions, and overlapping speech.

In the context of customer service, speech serves as a primary medium of interaction between customers and service agents. These interactions are rich in emotional content, as customers often express frustration, urgency, or satisfaction through both their words and vocal delivery. Accurately capturing these emotional cues is essential for understanding customer experience and improving service quality.

Furthermore, the integration of SSA into intelligent systems opens the door to real-time analysis, where emotional states can be detected as interactions unfold. This capability has significant implications for proactive customer support, enabling systems to assist agents during live conversations or trigger automated responses based on detected sentiment. As a result, SSA plays a crucial role in bridging the gap between raw conversational data and actionable intelligence, forming a foundational component of modern AI-driven customer service solutions.

1.5.1 Foundational Pillars of Speech Analysis

The following features shown in Table 1.1 form the foundational pillars for computationally discerning emotion from speech:

Table 1.1: Main acoustic features used in speech-based sentiment analysis

Feature	Emotional Correlation	Technical Role
Pitch (Fundamental Frequency)	A wider variation in pitch and a higher average pitch often indicate heightened emotional states such as excitement, agitation, or surprise. In contrast, a narrower pitch range and lower pitch are commonly associated with sadness or disengagement [37].	Pitch is one of the most important indicators of emotional arousal, helping distinguish between high-energy and low-energy affective states [38].
Speech Rate and Rhythm (Tempo)	A faster speech rate is often associated with emotions such as anxiety, excitement, or anger, while slower or irregular speech patterns may indicate sadness, hesitation, or cognitive effort [36].	Speech rate and rhythm provide insight into urgency, confidence, and emotional intensity, making them key temporal indicators in affective speech analysis.
Voice Quality / Timbre	Variations in voice quality are strong indicators of specific emotional states. For example, breathy or weak voices are often associated with sadness, while tense or harsh vocal qualities are linked to anger or stress [39].	Voice quality helps differentiate between emotions with similar arousal levels, such as anger and fear, by capturing fine-grained phonatory characteristics.
Loudness (Intensity / Amplitude)	Increased loudness is typically associated with high-arousal emotions such as anger or enthusiasm, while lower intensity often correlates with sadness or uncertainty [37].	Loudness serves as a key indicator of emotional intensity and helps track dynamic changes in affect during speech.
Spectral Features (Resonance)	Emotional states influence vocal tract configuration, resulting in measurable spectral variations. For example, stress or anxiety may introduce instability or perturbations in the signal, such as jitter and shimmer [36].	Spectral features provide subtle physiological indicators that are difficult to consciously control, making them valuable for detecting genuine emotional expression.

In practice, these features are not analyzed in isolation. The effectiveness of speech-based sentiment detection relies on a fusion of multiple acoustic and prosodic features. This multi-feature integration is essential for capturing the complexity and variability of human emotional expression [38, 40].

1.5.2 Importance of Sentiment Detection from Speech

The human voice is one of the most natural and expressive channels of communication, conveying emotional information beyond the literal meaning of words. This enables a more immediate and nuanced understanding of a speaker's emotional state [30]. This capability holds significant importance across several domains where spoken interaction is central.

1. **Healthcare and Telemedicine.** In clinical settings, vocal characteristics can serve as important indicators of psychological and emotional conditions. Speech-based sentiment analysis has been used to assist in the detection and monitoring of mental health disorders such as depression and anxiety, by analyzing variations in tone, pitch, and speech patterns over time [41]. In telemedicine, it can also support clinicians in assessing patient engagement, distress, or comprehension, thereby enhancing remote diagnostic capabilities and patient care.

2. **Education and E-Learning.** In both traditional and digital learning environments, analyzing student sentiment through speech provides valuable insights into engagement, confusion, and emotional response. This information can be used to adapt teaching strategies, personalize learning experiences, and improve educational outcomes by identifying when learners require additional support [42].

3. **Market Research and Focus Groups.** Spoken feedback in interviews and focus groups contains rich emotional information that extends beyond textual responses. Speech-based sentiment analysis enables the automatic evaluation of participants' emotional reactions, capturing spontaneous expressions such as enthusiasm, hesitation, or skepticism. This leads to deeper consumer insights

and more accurate assessments of product perception and user experience [36].

4. **Security and Forensic Analysis.** In domains such as law enforcement and security, detecting emotional cues such as stress, aggression, or deception from speech can support critical decision-making. Analysis of vocal patterns in emergency calls or investigative interviews can help identify high-risk situations and prioritize responses, enhancing situational awareness and operational effectiveness [43].

The ability to capture emotional signals enables the development of systems that are not only analytically intelligent but also emotionally aware and responsive.

1.5.3 Importance of Speech in Customer Service

Digital interactions such as emails and chat logs capture the logical structure of a problem, but spoken communication conveys emotional and relational dynamics that are critical to customer experience [44]. This is where the strategic value of speech-based sentiment analysis becomes evident.

The importance of speech lies in its unique capacity to convey nuance and intent beyond textual representation. Vocal cues provide critical information about a speaker's emotional state beyond spoken words [38]. For instance, identical phrases can convey vastly different meanings depending on how they are spoken, highlighting the limitations of text-only analysis in accurately capturing sentiment.

Consequently, focusing on speech audio is not an alternative to text analysis but a necessary extension for achieving a comprehensive understanding of customer interactions, which ultimately aims to improve customer satisfaction, trust, and loyalty [44].

1.5.4 Technical Challenges

Translating the cues of emotion into reliable computational insight presents a distinct set of challenges. These challenges arise from the inherent complexity of

human speech as well as the imperfect conditions under which real-world audio is captured [45].

1. **Acoustic Variability and Signal Noise.** The foundational audio signal is rarely clean. It is often contaminated by background noise, degraded by recording devices, and affected by transmission artifacts such as compression in telecommunication systems. These distortions can obscure subtle acoustic cues essential for emotion recognition, significantly reducing model accuracy [46].

2. **Speaker and Expressive Diversity.** A key challenge lies in distinguishing emotional expression from inherent speaker characteristics. Variations in pitch, speaking style, accent, age, and physiological differences introduce significant variability. Additionally, cultural norms influence how emotions are expressed, making it difficult to develop models that generalize across diverse populations [43].

3. **The Subjectivity of Ground Truth.** The annotation of emotional data is inherently subjective, as different human annotators may interpret the same vocal expression differently. This leads to inconsistencies in labeled datasets and introduces noise into the training process. Furthermore, cultural and contextual biases in annotation can result in models that do not generalize well across populations [36].

4. **The Dynamic and Contextual Nature of Conversation.** Real-world dialogue is complex and often includes interruptions, overlapping speech, and evolving emotional states. Emotional meaning is highly context-dependent, requiring models to capture temporal dependencies and conversational flow rather than relying solely on isolated utterances [47].

5. **The Complexity of Feature Integration.** Accurate emotion recognition requires the integration of multiple acoustic and prosodic features, such as pitch, intensity, spectral properties, and temporal dynamics. These features may sometimes provide conflicting signals, making it challenging to design

models that effectively fuse them into a coherent representation of emotional state [36].

These challenges collectively explain why sentiment detection from speech remains a complex and evolving research problem.

1.5.5 Current Approaches and Methodologies in Sentiment Analysis

Recent advances in Speech Emotion Recognition (SER) have been driven by the development of large-scale self-supervised models and specialized architectures designed to capture both acoustic and contextual emotional cues. Modern approaches can be broadly categorized based on their learning strategy and the nature of the datasets they are optimized for.

Self-supervised learning models such as HuBERT [48] and WavLM [49] have established strong baselines by learning robust speech representations from large unlabeled corpora. HuBERT leverages masked prediction of hidden units to model latent speech structures, making it particularly effective for spontaneous and naturalistic datasets such as MSP-Podcast. Building upon this paradigm, WavLM introduces a masked speech denoising objective, which enhances robustness in noisy and real-world environments, and demonstrates strong performance on datasets such as IEMOCAP.

In parallel, recent work has focused on fine-grained emotion recognition using large annotated datasets. The Empathic Insight-Voice framework [50] utilizes a two-stage architecture combining a pre-trained Whisper encoder with specialized emotion classification heads. This approach is trained on the EmoNet-Voice dataset, which contains over 5,000 hours of expert-annotated speech and enables the recognition of a wide range of emotional categories beyond traditional labels.

Additionally, task-specific architectures have been proposed to capture both temporal and spatial characteristics of emotional speech signals. Emo-STCapsNet [51] introduces a spatio-temporal capsule network that models dynamic speech

patterns and achieves state-of-the-art performance on acted emotion datasets such as RAVDESS.

Overall, the current state of the art demonstrates a clear trend toward combining large-scale pretraining with task-specific adaptations. Furthermore, performance differences across models are strongly influenced by the nature of the datasets used, which range from controlled datasets such as RAVDESS, to naturalistic datasets such as MSP-Podcast, and large-scale annotated corpora such as EmoNet-Voice.

Table 1.2 shows a summary of the prior discussed models, their strengths, and primary datasets:

Model	Core Focus	Primary Dataset
WavLM	Robustness to Noise / Real-world Speech	IEMOCAP
HuBERT	General Speech Representation Learning	MSP-Podcast
Empathic Insight-Voice	Fine-grained Emotion Recognition	EmoNet-Voice
Emo-STCapsNet	Spatio-temporal Modeling / High Accuracy	RAVDESS

Table 1.2: Comparison of State-of-the-Art SER Models and Their Dataset Focus

1.6 Conclusion

This chapter highlighted how Artificial Intelligence can support customer service by helping organizations process and interpret these interactions more efficiently.

It presented it as a method for transforming subjective expressions into structured information, while the speech-based dimension of sentiment analysis was discussed in greater detail through its challenges and potential because the proposed work focuses on customer service calls.

Chapter 2

Speech Transcription and Text Summarization in Customer Service

2.1 Introduction

After introducing the foundational concepts of customer service, Artificial Intelligence, and speech-based sentiment analysis, this chapter focuses on the two remaining technical components of the proposed system: speech transcription and text summarization. These components are essential because customer service calls must first be converted from spoken audio into written text before their content can be condensed into a useful summary.

This chapter first presents speech transcription and the main challenges associated with Automatic Speech Recognition in real-world customer service contexts. It then discusses text summarization, with a focus on dialogue summarization. These concepts establish the technical background needed for the design and methodology of the proposed system.

2.2 Speech Transcription

Before any analysis of dialogue content can occur, the spoken word must be converted into written text. This process, known as speech transcription or Automatic Speech Recognition (ASR), is the first step that creates the textual input required for later processing. In the context of the proposed system, transcription provides the transcript that will later be used by the summarization module.

However, speech transcription is not a simple conversion process. It presents several challenges, especially in real-world customer service calls:

1. **From Sound Waves to Language Units.** The primary task is mapping highly variable acoustic signals—the continuous flow of speech—onto discrete linguistic units such as phonemes and words. This is challenging due to variability in pronunciation, accents, speaking rate, and articulation. Additionally, ASR systems must filter out non-linguistic sounds such as breaths and background artifacts while preserving meaningful speech components [52].

2. **Acoustic and Environmental Noise.** Real-world audio, especially in customer service environments, is often degraded by background noise, channel distortion, and transmission artifacts. These factors significantly affect recognition accuracy, requiring robust noise-handling and speech enhancement techniques to isolate the speaker's voice from environmental interference [46].

3. **The Vocabulary and Context Problem.** ASR systems must handle domain-specific vocabulary, including technical terms, product names, and specialized jargon. Without domain adaptation, models may misrecognize or incorrectly transcribe critical terms. Incorporating contextual language models is therefore essential for improving transcription accuracy in specialized domains [53].

4. **Speaker Diarization: The "Who Spoke When" Problem.** Accurate transcription of multi-speaker conversations requires identifying speaker boundaries and assigning speech segments to the correct speaker. This task,

known as speaker diarization, is particularly challenging in the presence of overlapping speech, interruptions, and conversational dynamics [54].

5. **The Accuracy-Speed Trade-off in Real-Time Processing.** In real-time applications, ASR systems must produce transcripts with minimal latency. However, more accurate models often require greater computational resources and processing time. This creates a trade-off between transcription speed and accuracy, especially in latency-sensitive environments such as live customer support systems [55].

The general role of ASR in a speech processing pipeline is illustrated in Figure 2.1, where the audio signal is progressively transformed into textual information that can be used by downstream language-processing modules.

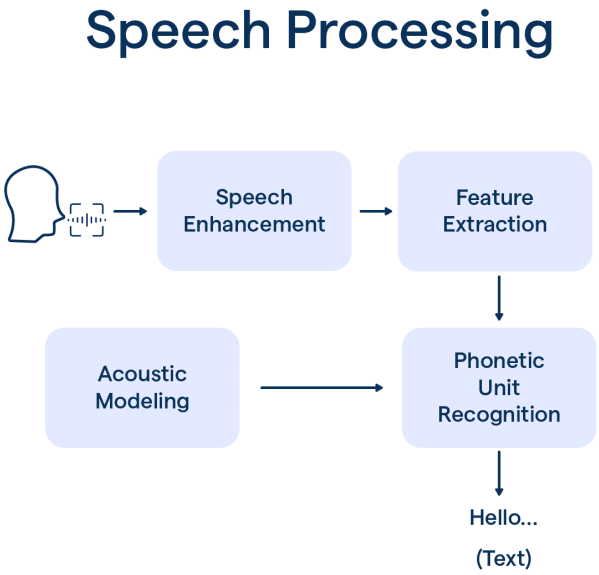


Figure 2.1: Example speech processing pipeline [9]

Errors introduced at this stage—such as misrecognized words, incorrect speaker attribution, or missing contextual cues—can propagate through the entire processing pipeline. This directly affects the quality of downstream tasks, especially dialogue

summarization, where the generated summary depends on the reliability of the transcript [55].

2.2.1 Current Approaches and Methodologies in Speech Transcription

Recent advancements in Automatic Speech Recognition have moved beyond traditional speech-to-text systems toward large-scale Audio-Language Models and efficient streaming architectures. Modern approaches focus not only on transcription accuracy but also on robustness, latency, multilingual capabilities, and semantic understanding.

Whisper Large-v3 [56] represents a major breakthrough in ASR through large-scale weak supervision. Trained on the OpenAI-680k dataset, comprising approximately 680,000 hours of multilingual audio with noisy transcripts, Whisper demonstrates that scale and diversity can outperform smaller curated datasets. The Turbo variant further improves efficiency by reducing decoding latency while maintaining high transcription accuracy, making it suitable for real-time applications.

Canary-1B-v2 [57] introduces a lightweight yet high-performance architecture based on the FastConformer encoder. Developed by NVIDIA, it is trained on multilingual datasets such as MLC, FLEURS, and Common Voice 15.0. Despite having fewer parameters compared to large models, it achieves competitive or superior Word Error Rates while offering improved throughput and built-in punctuation and capitalization, which are essential for downstream tasks such as summarization.

Qwen2-Audio [58] represents a shift toward unified audio-language modeling. It combines a Whisper-based encoder with a large language model decoder, enabling it to perform instruction-aware transcription. Unlike conventional ASR systems, it can adapt outputs based on contextual prompts, such as focusing on specific parts of a conversation, thereby bridging transcription and semantic understanding.

SeamlessM4T v2 [59] extends ASR into fully multilingual and multimodal communication. Trained on the SeamlessAlign dataset, which contains millions

of hours of aligned speech and text, it supports speech-to-text, speech-to-speech, and translation tasks. Its strengths include handling code-switching scenarios and performing speaker-aware processing, making it particularly relevant for real-world customer service environments.

Overall, the state of the art in speech transcription demonstrates a clear evolution toward scalable, multilingual, and context-aware systems. Table 2.1 summarizes several representative ASR models and highlights their main datasets, approximate performance ranges, and key strengths. The choice of model is strongly influenced by deployment constraints, such as latency requirements, language diversity, and the need for semantic understanding beyond raw transcription.

Model	**Primary Dataset**	**WER (Approx.)**	**Key Strength**
Whisper V3 Turbo	OpenAI-680k	∼3–5%	Robustness to noise
Canary-1B-v2	MLC / FLEURS / Common Voice	∼2–4%	High throughput
Qwen2-Audio	Universal Audio Dataset	∼1.6% (clean)	Semantic understanding
SeamlessM4T v2	SeamlessAlign	∼4–6%	Multilingual & diarization

Table 2.1: Comparison of State-of-the-Art ASR Models and Their Characteristics

2.3 Text Summarization

Text summarization is the computational discipline dedicated to condensing lengthy documents or conversations into their most significant points. Its purpose is to reduce information overload by distilling complex material into concise, readable, and informative synopses. Summarization is not random deletion; it is an interpretive process that preserves essential meaning, key facts, and the central narrative while reducing volume [60].

In the context of speech-based systems, text summarization focuses on condensing transcribed dialogue into concise and informative representations. Unlike traditional document summarization, the objective here is to capture the key elements of conversational exchanges, including the problem discussed, the actions taken, and the final outcome [61].

When the object of analysis shifts from prepared documents to spontaneous spoken dialogue—such as customer service interactions—the task becomes more complex. Speech is unstructured and dynamic, containing turn-taking, interruptions, repetitions, and emphasis. A critical prerequisite is converting speech into text through Automatic Speech Recognition [23]. The quality of the transcript directly affects the quality of the summary, as errors propagate into the summarization stage [62].

Once a reliable transcript is obtained, summarization techniques identify salient information and determine how it should be represented. This leads to the distinction between extractive and abstractive summarization. Extractive methods select relevant segments from the source text, while abstractive methods generate new sentences that paraphrase and synthesize meaning [60].

Ultimately, dialogue summarization imposes structure on naturally unstructured communication, transforming conversational data—often redundant and digressive—into concise, coherent, and actionable representations. These summaries capture both intent and outcomes, supporting knowledge management, decision-making, and service optimization [61].

2.3.1 Fundamentals of Dialogue Summarization

Summarizing customer service calls differs significantly from summarizing documents. While documents are structured, dialogues are spontaneous and unorganized. Human interactions include digressions, repetitions, disfluencies, and incomplete thoughts, making meaningful extraction more challenging [63].

This complexity arises from several characteristics:

- **Multiple participants**, typically a customer and an agent, requiring correct attribution of statements [62].
- **Implicit conversational flow**, progressing from greetings to problem description, troubleshooting, resolution, and closure.

- **Dual purpose**, resolving a functional issue while maintaining a positive relationship, introducing informational and emotional dimensions [61].

The core task is to identify and preserve the most relevant information: the customer's issue, steps taken, and final resolution. Summaries must also capture actionable elements such as follow-ups, deadlines, or transaction details, such as order numbers [64]. Figure 2.2 provides an example of how a dialogue can be reduced into a shorter summary while preserving the main point of the exchange.

Dialogue

Owen: hey, how's your apartment search going?
Monica: not so good. not getting many responses.
Owen: i'm sorry. did you try that website I sent u?
Monica: yeah, i did. no luck, but i'll keep trying.
Owen: have you ever tried one of those websites where you can go and meet potential roommates at organized events?
Monica: yes, i went to a couple. I met some nice people there, but things didn't work out in the end.
…
Owen: yeah. do you want to come over and take a look at the apartment some time? I can make dinner.
Monica: Sure, I'd love to.
Owen: When would be a good time?
Monica: I'm free thursday night.
Owen: OK, cool. thats a good time for me.
…
Monica: cool. alright, I'll see you thursday then.

Summary: Monica's looking for an apartment, but can't find anything. She'll visit Owen in his new place Thursday night, Owen'll make dinner.

Figure 2.2: Dialogue summary example [10]

Quality summaries are concise yet informative, accurate in representing facts, coherent, and logically structured. They often follow a problem-solution format, preserving chronological progression and clearly attributing actions to partici-

pants. This transforms complex exchanges into clear, actionable records supporting decision-making and operational continuity [60].

2.3.2 Applications in Different Domains

Automatic summarization has critical utility across many fields, transforming how information is processed and leveraged [60].

1. **Academic Research and Literature Review.** Summarization tools help researchers navigate overwhelming volumes of publications by providing concise overviews of relevant articles, accelerating the early stages of research and trend identification [65].
2. **Legal and Judicial Proceedings.** Legal professionals deal with extensive documentation. Summarization condenses depositions, case law, and rulings into manageable briefs, improving preparation efficiency and ensuring key arguments are not overlooked [66].
3. **News Media and Information Services.** Automated summarization powers news aggregation by producing digestible briefs from multiple articles. It enables rapid reporting and helps readers access essential facts quickly [67].
4. **Healthcare and Medical Practice.** Clinicians face large volumes of patient records and medical literature. Summarization condenses histories or research findings, supporting faster and more informed decision-making [68].
5. **Corporate and Financial Analysis.** Businesses rely on synthesizing information from reports, earnings calls, and strategy documents. Summarization highlights financial performance, strategic shifts, and risks, enabling faster, data-driven decisions [69].

Across these diverse domains, the common thread is the transformation of textual data from an operational burden into a strategic asset. Text summarization does not replace deep reading, but strategically enables it, ensuring that human

expertise is directed toward critical analysis and decision-making rather than being consumed by the preliminary task of information sifting.

2.3.3 Extractive Summarization Approach

The extractive approach to summarization operates on a straightforward principle: the summary should be assembled from the most important sentences that already exist within the original dialogue. The system identifies, ranks, and concatenates key segments, preserving the original wording [60]. This approach relies on several concepts:

1. **Feature-Based Importance Scoring.** This is the traditional engine for ranking sentences. It assigns importance based on quantifiable signals within the text. Key metrics include term frequency, sentence position, and the presence of cue phrases, all of which have been widely used in early summarization systems to approximate salience [70, 71]. The system's logic is built upon these observable, statistical clues.

2. **Structural Role Awareness.** In a dialogue, *who* says something is as crucial as *what* is said. Therefore, a critical adaptation involves weighting utterances based on speaker role. Statements from the customer often contain the problem description, while those from the agent include solutions and instructions. Leveraging dialogue structure in this way improves the relevance of extracted summaries [63].

3. **Graph-Based Centrality.** A more advanced perspective models the entire conversation as a network of ideas. Here, each sentence becomes a node, and connections are drawn between sentences that are semantically similar. The most important sentences are those that are most central within this network. This technique helps surface the most representative themes of the dialogue [72, 73].

4. **The Problem-Solution Trajectory.** For customer service, a good summary must trace a narrative arc. The system must identify both the segment where

the customer's core issue is expressed and the segment where a solution or resolution is provided. Dialogue summarization research highlights the importance of capturing this interaction flow to preserve meaning and usability [61].

5. **Action Item Extraction.** Beyond general narrative, practical applications require identifying commitments and follow-up actions. Extracting actionable items such as tasks, promises, or deadlines is essential for operational use, transforming summaries into tools for decision-making and workflow continuity [64].

While this approach guarantees factual accuracy by using the speaker's own words, its main limitation is coherence. Extracted sentences may lack natural flow when combined, resulting in summaries that can feel disjointed. Despite this, extractive methods remain valuable due to their transparency and reliability in preserving original information [60].

2.3.4 Abstractive Summarization Approach

The abstractive approach to summarization seeks to generate a new summary with wording that may differ from the original text by understanding its essential meaning. This makes abstractive summarization especially useful for conversational data, where the original exchange may contain repetitions, interruptions, and informal language [74].

1. **Semantic Understanding and Encoding.** Before any text is generated, the system must build a deep representation of the dialogue's meaning. This involves capturing contextual dependencies, speaker intent, and relationships between utterances using neural language models that encode the conversation into dense vector representations [75].

2. **The Encoder-Decoder Architecture with Attention.** This is the core engine for generation. The system is split into two components: an encoder that processes the input sequence and a decoder that generates the summary token

by token. The attention mechanism allows the decoder to dynamically focus on relevant parts of the input during generation, improving both relevance and coherence [76].

3. **Paraphrasing and Information Fusion.** This is the defining capability of abstractive summarization. The system learns to rephrase content and combine information from multiple parts of the dialogue into concise statements. Neural models such as sequence-to-sequence architectures enable this synthesis by learning semantic equivalence and compression across long contexts [77].

4. **Controllable Generation for Structure.** To ensure usefulness in practical applications, generation can be guided using structural constraints or prompts. Models can be conditioned to produce summaries following predefined formats, such as problem-solution-outcome structures, improving interpretability and alignment with business requirements [78].

5. **The Hallucination Problem and Faithfulness.** A major limitation of abstractive models is hallucination—the generation of content that is not supported by the source input. This issue arises from the generative nature of neural models and poses significant risks in high-stakes applications. Ensuring faithfulness and factual consistency has therefore become a central challenge in modern summarization research [79].

The strength of the abstractive approach lies in its ability to produce fluent and coherent summaries that capture the essence of complex interactions. However, this comes at the cost of potential inaccuracies. As illustrated in Figure 2.3, this differs from extractive summarization, where the summary is formed by selecting existing parts of the source text. As a result, its practical use requires careful balancing between generation quality and factual reliability, often motivating hybrid approaches that combine abstractive and extractive techniques [67].

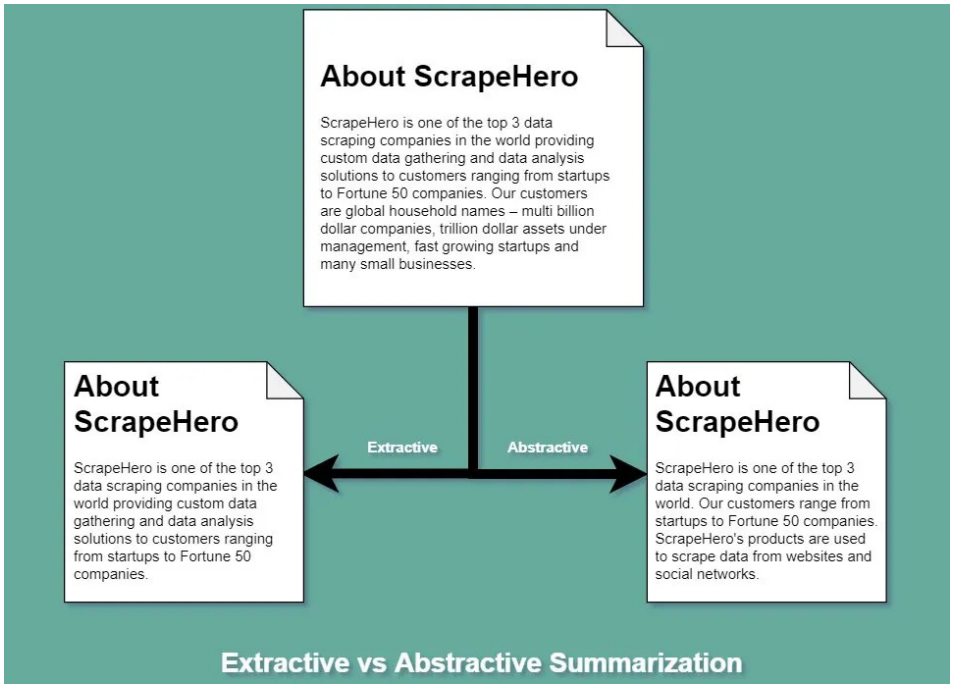


Figure 2.3: The difference between extractive and abstractive summarization [11]

2.3.5 Applications in Customer Service Operations

Transforming raw conversations into structured summaries can support several customer service operations. In daily work, summaries help agents keep track of the main issue, actions taken, and agreed outcomes. They are also useful when a case is transferred between agents or departments, since the customer does not need to repeat the same information again [64].

Summarization can also improve quality monitoring and agent learning. Instead of manually reviewing a small number of calls, supervisors can use summaries to examine a larger number of interactions, identify recurring issues, and detect training needs. Past summaries may also be reused to create case studies, improve knowledge bases, and highlight successful resolution strategies [61, 80].

From a management perspective, aggregated summaries provide a high-level view of customer concerns, operational bottlenecks, and service trends. When combined with sentiment analysis, they offer both factual and emotional insight into customer

interactions. This makes summarization valuable not only for documentation, but also for decision-making, service improvement, and integration with tools such as CRM systems [17, 21].

2.3.6 Key Challenges in Customer Service Contexts

Summarizing customer service dialogues is challenging because conversations are less structured than written documents. They often include multiple speakers, interruptions, repetitions, greetings, and irrelevant details that must be filtered out before the main issue can be identified [63]. Call length can also vary greatly, from short requests to long problem-solving interactions with changing context [61].

Another difficulty is that customer service calls are highly context-dependent. They may contain product names, account details, technical terms, and implicit references that are not always easy for an automatic system to interpret. A single call may also include several related problems, making it necessary to identify which details are central to the final summary [64].

The summarization system must therefore prioritize the most useful information, such as the customer's issue, actions taken by the agent, and whether the problem was resolved, escalated, or left pending. At the same time, practical deployment requires summaries to be produced quickly and consistently across many different calls. Evaluation also remains difficult, since several valid summaries can exist for the same conversation depending on the needs of agents, supervisors, or managers [47, 60].

2.3.7 Current Approaches and Methodologies in Text Summarization

The state of the art in text summarization has evolved significantly with the emergence of large language models, mixture-of-experts architectures, and agentic reasoning frameworks. Modern summarization systems are no longer limited to compressing text but are designed to preserve factual consistency, handle long conversational contexts, and capture underlying intent and emotional dynamics.

Llama 4 represents a major advancement through its Mixture-of-Experts architecture, enabling efficient scaling to hundreds of billions of parameters. Trained on datasets such as Spoken DialogSum, which includes emotion-focused summaries, Llama 4 excels in reasoning-based summarization. It can interpret the underlying causes of customer dissatisfaction and provide structured explanations of both problems and resolutions [81].

Qwen3-30B-A3B focuses on long-context summarization, supporting input sequences of up to 256,000 tokens. This capability allows the model to process extended conversations, such as long customer service calls or multiple interactions, without losing contextual coherence. It is commonly evaluated on datasets such as MediaSum and SAMSum, demonstrating strong performance across ROUGE and METEOR metrics [82].

BART-Large, particularly when fine-tuned with adaptive techniques such as Adaptive Fusion, remains a strong academic baseline. Despite being an earlier model, its encoder-decoder architecture continues to deliver high-quality abstractive summaries, especially when adapted to domain-specific datasets like SAMSum. Its strength lies in data efficiency and stable performance on smaller datasets [83, 84].

GPT-OSS-120B introduces reasoning-based summarization through Chain-of-Thought mechanisms. By generating intermediate reasoning steps before producing the final summary, the model improves factual consistency and reduces hallucination. Evaluated on datasets such as DialogSum and CS-Summarization-2026, it demonstrates strong performance on semantic similarity metrics such as BERTScore [85].

The main characteristics of these recent summarization models are summarized in Table 2.2, which compares their datasets, metrics, and main strengths.

Overall, the current state of the art in text summarization reflects a shift toward models that combine large-scale pretraining with reasoning capabilities and long-context processing. The effectiveness of these systems depends not only on architecture but also on the nature of the datasets, which increasingly emphasize

Model	**Dataset**	**Metric**	**Strength**
Llama 4 (Maverick)	Spoken DialogSum	ROUGE-L / G-Eval	Emotional reasoning
Qwen3-30B	MediaSum / SAMSum	ROUGE-2 / METEOR	Long-context (256K)
BART (AAF Fine-tuned)	SAMSum	ROUGE-1	Small-data efficiency
GPT-OSS-120B	DialogSum / CS-2026	BERTScore	Factuality / Low hallucination

Table 2.2: Comparison of State-of-the-Art Text Summarization Models

dialogue structure, emotional context, and real-world applicability.

2.4 Conclusion

This chapter presented the two main text-oriented components that support the proposed customer service call analysis system, their challenges, and their uses. It established the technical background for transforming spoken customer interactions into written transcripts and concise summaries. Together with the speech sentiment analysis concepts introduced in the previous chapter, these elements provide the foundation for the proposed system presented in the following chapter.

Chapter 3

System Design and Methodology

3.1 Introduction

This chapter presents the design and methodology of the proposed system for customer service call analysis. It explains how the system is organized to process call audio and generate three main outputs: a transcript, a summary, and a dominant sentiment label.

The chapter describes the overall architecture, data pipeline, selected models, and processing steps used in each module. It also explains how the sentiment analysis, speech transcription, and text summarization components are integrated into a unified workflow before the implementation and evaluation phase.

3.2 System Overview

The proposed system is designed as a desktop application for the real-time analysis of customer service calls. Its main objective is to transform a live spoken interaction into structured and interpretable outputs that can support agents and supervisors during or immediately after the conversation. By converting raw audio into a transcript, a summary, and a sentiment label, the system provides a more compact and useful representation of the call that can be reviewed and exploited more easily later on.

The general workflow begins with the capture of the incoming call audio. From this audio stream, the system follows two complementary processing branches. The first branch is dedicated to speech transcription, where the spoken content is converted into a textual transcript. This transcript is then passed to the text summarization module, which condenses the conversation into a concise representation of its essential content. The second branch performs speech emotion recognition directly from the audio signal, extracting acoustic features that capture vocal expression patterns such as spectral structure, rhythm, energy, and temporal variation. These features are processed by the proposed model to identify the dominant emotional state expressed during the call. This general data flow is illustrated in Figure 3.1.

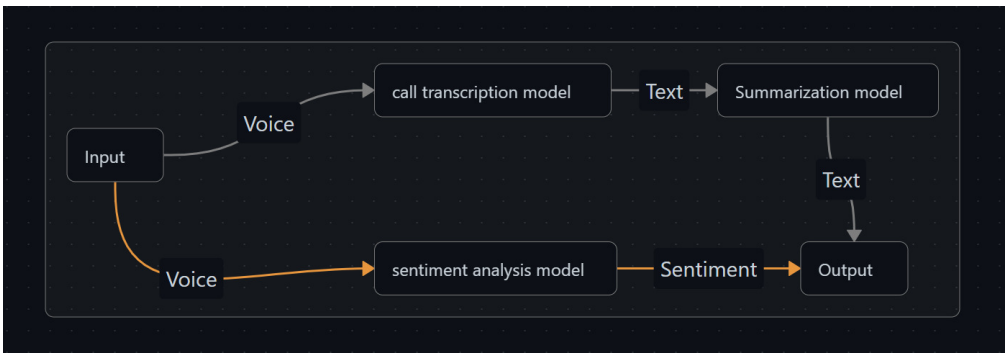


Figure 3.1: Data flow

The final output of the application therefore consists of three main elements: the full transcript of the call, a concise summary of the interaction, and the dominant sentiment label extracted from the speech signal. Together, these outputs provide a more complete view of the customer service exchange. The transcript preserves the factual content of the call, the summary highlights the most important points and outcomes, and the sentiment label provides an immediate indication of the emotional state characterizing the interaction.

The main interactions between the user and the system are summarized in Figure 3.2. This diagram shows the principal functions supported by the desktop application, including processing calls and reviewing the generated outputs.

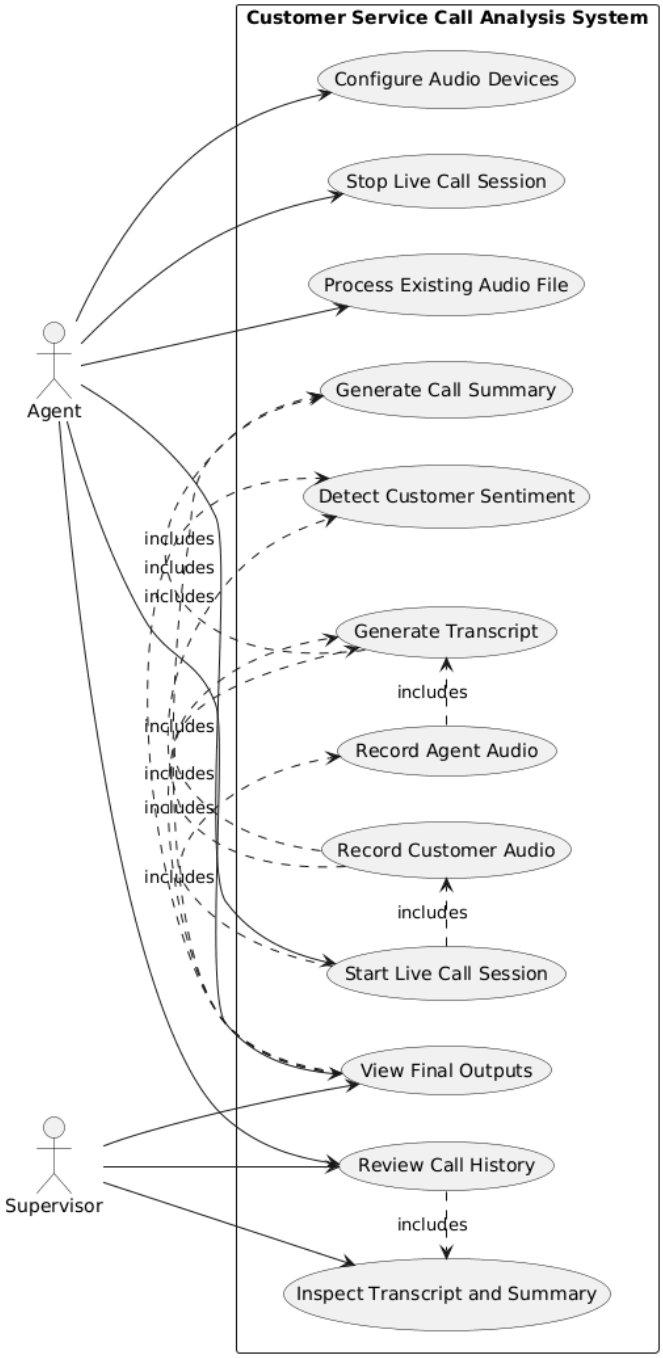


Figure 3.2: Use case diagram of the customer service call analysis system

The desktop application is designed to provide a practical interface through

which users can view processed call information in an accessible format. This design choice makes the system suitable for experimental validation while also reflecting a realistic customer service scenario, where agents or supervisors may require immediate access to call transcripts, summaries, and emotional indicators. Overall, the proposed architecture aims to bridge the gap between raw voice interactions and actionable customer service intelligence.

3.3 Sentiment Analysis Model

The sentiment analysis model represents the emotional interpretation component of the proposed system. Unlike approaches that infer sentiment from transcribed text, this module operates directly on the audio signal in order to preserve the vocal information that may be lost during transcription. This distinction is particularly important in customer service calls, where the emotional state of the customer is often expressed not only through words, but also through tone, rhythm, intensity, hesitation, and other acoustic cues.

Within the global pipeline, the role of this module is to analyze the incoming call audio and identify the most prominent sentiment expressed during the interaction. The output is not intended to replace the transcript or the summary, but rather to complement them by adding an emotional layer to the factual content of the conversation. In this way, the system is able to provide a more complete representation of the call, combining what was said with how it was expressed.

3.3.1 Datasets

The selection of datasets is an important step in the development of the sentiment analysis module, since the model learns emotional patterns directly from speech signals. The selected datasets must provide audio recordings with emotion labels. The goal is therefore to use datasets that offer sufficient emotional variety, audio quality, and speaker diversity to support the training and evaluation of a speech-based sentiment analysis model.

The main datasets selected for this module are Emo-DB, SAVEE, and RAVDESS.

The Berlin Emotional Speech Database, commonly known as Emo-DB, contains acted German emotional speech recorded under controlled conditions. Although relatively small, it provides clean and well-annotated utterances, making it useful for training and baseline evaluation in speech emotion recognition experiments [86].

The Surrey Audio-Visual Expressed Emotion dataset, known as SAVEE, is also used as one of the main datasets. It contains emotional speech recordings produced by male speakers and covers several basic emotions, including anger, disgust, fear, happiness, sadness, surprise, and neutrality. Its clear emotional labels and controlled recording conditions make it suitable for learning acoustic patterns associated with different emotional states [87].

RAVDESS is included as another main dataset because it provides high-quality, validated emotional speech recordings. It contains acted emotional expressions in North American English and covers emotions such as calm, happy, sad, angry, fearful, surprise, disgust, and neutral. Its balanced structure and widespread use in speech emotion recognition make it valuable for training and performance comparison [88].

In addition to these main datasets, CASIA and CREMA-D are used as testing datasets to evaluate the generalization ability of the model. CASIA provides emotional speech data in Mandarin Chinese, introducing linguistic variation and allowing assessment of the model's robustness beyond the languages used in training. CREMA-D is also useful for evaluation because it contains emotional speech recordings from a diverse group of actors with different demographic profiles, enabling assessment across a wider range of vocal characteristics [89].

The role of each dataset in the sentiment analysis module is summarized in Table 3.1.

Table 3.1: Datasets considered for the speech-based sentiment analysis module

Dataset	**Language**	**Type**	**Role in This Work**
Emo-DB	German	Acted emotional speech	Main dataset
SAVEE	English	Acted emotional speech	Main dataset
RAVDESS	English	Acted audio-visual speech	Main dataset
CASIA	Mandarin Chinese	Acted emotional speech	Testing dataset
CREMA-D	English	Multimodal acted speech	Testing dataset

Overall, the dataset strategy for this module combines controlled emotional speech datasets for model development with additional datasets for generalization testing. Emo-DB, SAVEE, and RAVDESS are used as the main resources because they provide clear emotional labels and structured audio samples, while CASIA and CREMA-D are used to examine how well the model performs on speech data with different linguistic and speaker characteristics.

3.3.2 Model Selection

The selection of the sentiment analysis model was guided by the requirements of the proposed system in terms of machine specific resources and by the constraints of speech emotion recognition datasets. Since the sentiment module operates on speech signals rather than on transcribed text, the selected model must be able to capture emotional patterns from acoustic features. In addition, SER datasets are often limited in size, which increases the risk of overfitting when using large deep learning architectures.

For this work, the selected model is based on MA-CapsNet-DA, a Max-Avg-Pooling Capsule Network with Data Augmentation proposed for speech emotion recognition [90]. This model was chosen because it combines a capsule-based architecture with data augmentation strategies designed to improve robustness when training on small emotional speech datasets. Instead of using raw audio directly, each audio file is first converted into frame-level acoustic features composed of MFCC coefficients and zero-crossing rate. This allows the model to learn from structured speech representations while keeping the input manageable.

The proposed implementation follows the general MA-CapsNet-DA design [90], but it is not an exact reproduction of the original architecture. After testing several architectural adjustments and additions, a denser version of the model was adopted.

Therefore, the selected model can be described as a denser, paper-aligned adaptation of MA-CapsNet-DA for speech emotion recognition. The design preserves the core principles of the reference model while adapting its capacity and implementation details to the experimental needs of this thesis.

3.3.3 Feature Extraction And Processing

The sentiment analysis module does not use raw audio signals directly as input to the model. Instead, each speech file is first converted into a structured acoustic representation that can be processed by the capsule-based architecture. This step is necessary because emotional information in speech is carried by measurable acoustic patterns such as frequency distribution, rhythm, energy changes, and short-term variations in the signal.

In this work, the feature extraction process follows the MA-CapsNet-DA approach [90]. Each audio recording is divided into short overlapping frames using a 25 ms window and a 10 ms shift. For every frame, a feature vector is extracted composed of 20 Mel-Frequency Cepstral Coefficients (MFCCs) and one Zero-Crossing Rate (ZCR) value. This produces a 21-dimensional feature representation for each frame. MFCCs are used because they provide a compact description of the spectral characteristics of speech, while ZCR captures the rate at which the signal changes sign, which can reflect variations in energy and voice texture.

After feature extraction, normalization is applied to make the input data more stable during training. The extracted features are standardized to have zero mean and unit variance. This step reduces the influence of scale differences between features and helps the model learn more effectively across multiple datasets. Since the selected datasets may differ in language, recording conditions, speaker characteristics, and audio quality, normalization is important for creating a more consistent input representation.

The processing pipeline also supports multi-dataset integration. Emotional speech samples from different datasets are mapped into a unified label space, allowing the model to learn from a broader range of speakers, languages, and recording conditions. Dataset specific filename formats are parsed and converted into common emotion labels such as anger, disgust, fear, happiness, neutral, sadness, and surprise. This harmonization step is essential because each dataset may use different naming conventions or emotion categories.

To improve robustness, noisy data augmentation is applied using the NOISEX-92 noise corpus [91]. Clean speech samples are combined with different noise types at multiple signal-to-noise ratio levels. This allows the model to learn from both clean and degraded speech conditions. The augmentation process therefore helps evaluate and improve the model's ability to generalize beyond controlled laboratory recordings.

Overall, the feature extraction and processing stage transforms heterogeneous emotional speech recordings into normalized acoustic feature which represent the input of the MA-CapsNet-based model and provide the foundation for learning emotional patterns from speech.

3.3.4 Architecture

The architecture selected for the sentiment analysis module is based on MA-CapsNet-DA [90]. This model was chosen because it combines a capsule-based design with data augmentation strategies intended to improve robustness when working with small emotional speech datasets and its low resource needs due to its compact architecture. The model receives structured frame-level acoustic features extracted from each audio file, mainly MFCC coefficients and zero-crossing rate. This representation allows it to learn from relevant speech characteristics while keeping the input more manageable.

The model architecture is organized around several successive blocks. First, the extracted acoustic feature matrix is passed through parallel convolutional modules. One branch applies convolution followed by max pooling, while the other applies

convolution followed by average pooling. The purpose of this parallel design is to capture complementary information from the acoustic representation: max pooling emphasizes the most dominant local patterns, while average pooling preserves more general tendencies across the feature map. These outputs are then followed by an additional convolutional module that further refines the extracted representations before they are passed to the capsule layers. The complete structure of this model is shown in Figure 3.3.

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 404, 21, 1)	0
conv_max_conv (Conv2D)	(None, 404, 21, 128)	1,280
conv_max_pool (MaxPooling2D)	(None, 202, 10, 128)	0
leaky_re_lu (LeakyReLU)	(None, 202, 10, 128)	0
conv_avg_conv (Conv2D)	(None, 202, 10, 128)	147,584
conv_avg_pool (AveragePooling2D)	(None, 101, 5, 128)	0
leaky_re_lu_1 (LeakyReLU)	(None, 101, 5, 128)	0
conv_module_conv (Conv2D)	(None, 101, 5, 128)	147,584
leaky_re_lu_2 (LeakyReLU)	(None, 101, 5, 128)	0
primary_caps (PrimaryCaps)	(None, 128, 8)	132,096
digit_caps (DigitCaps)	(None, 8, 16)	131,200
capsule_lengths (Length)	(None, 8)	0

Total params: 559,744 (2.14 MB)
Trainable params: 559,744 (2.14 MB)
Non-trainable params: 0 (0.00 B)

Figure 3.3: Architecture of the sentiment analysis model

The capsule part of the architecture is composed of a primary capsule layer and a digit capsule layer. Unlike standard neural network layers that treat activations independently, capsule layers are designed to preserve relationships between features. This is particularly useful in speech emotion recognition, where emotional cues are rarely represented by a single isolated feature. Instead, emotions are usually expressed through combinations of spectral characteristics, temporal dynamics, energy variations, and voice quality patterns.

In the primary capsule layer, the convolutional features are reorganized into capsule vectors that represent local emotional patterns in the speech signal. These capsules are then connected to the digital capsule layer, where each output capsule corresponds to one emotion class. Dynamic routing is applied between the capsule layers so that lower-level capsules can send stronger contributions to the most

relevant higher-level emotion capsule. The length of each digital capsule vector is then interpreted as the probability of the corresponding emotion class.

For this thesis, the implemented architecture follows the general design of MA-CapsNet-DA while adopting a denser version after testing different architectural adjustments. This adaptation aims to improve the model's ability to learn from multi-dataset training and to strengthen the model's ability to generalize across different emotional speech corpora while preserving the lightweight nature of the original capsule-based approach. This choice was made as a practical design decision during experimentation.

3.4 Speech Transcription Model

The speech transcription model represents the linguistic conversion component of the proposed system. Its role is to transform incoming call audio into a written transcript that can be read, stored, and later processed by the summarization module. Since the system is intended for real-time customer service calls, this module must not only recognize spoken words accurately but also produce text with sufficient speed and clarity to remain practically useful.

Within the global architecture, the transcription model operates directly on the same audio stream used by the sentiment analysis module. However, its objective is different, as it focuses on preserving the linguistic content of the conversation. These transcripts serve as the foundation for generating the final call summary. Any errors introduced at this stage, such as misrecognized words, missing phrases, or incorrect segmentation, can directly affect the quality and reliability of the summarization output.

3.4.1 Datasets

The speech transcription module requires datasets that provide paired audio recordings and textual transcriptions. Since the proposed system is intended to process real-time customer service calls, the selected datasets must support two complementary needs: Speaker diversity and conversational speech. For this reason,

the main datasets considered for this module are Mozilla Common Voice and the AMI Meeting Corpus.

Mozilla Common Voice is selected as the primary dataset due to its broad and diverse collection of speakers, accents, and recording conditions. It is a large-scale, community-driven speech corpus designed to support the development of speech recognition systems across multiple languages. Its crowdsourced nature makes it particularly useful for improving robustness, since the recordings are contributed by a wide variety of speakers rather than being limited to a small controlled group [92]. This diversity is important for customer service applications, where callers may differ significantly in accent, pronunciation, speech rate, microphone quality, and background noise.

The AMI Meeting Corpus is considered as a complementary dataset because it provides multi-speaker conversational recordings. Unlike single-speaker read-speech datasets, AMI contains meeting-style interactions with turn-taking, interruptions, and overlapping speech, which makes it closer to real-world dialogue scenarios [93]. Although meeting conversations are not identical to customer service calls, they share their structure, such as multiple participants, and conversational flow. Therefore, AMI is useful for evaluating how the transcription module handles dialogue-like audio, speaker changes, and less structured speech.

Table 3.2 summarizes the datasets considered for the transcription module and the role of each one in this work.

Table 3.2: Datasets considered for the speech transcription module

Dataset	**Type**	**Main Strength**	**Role in This Work**
Mozilla Common Voice	Crowdsourced read speech	Speaker and accent diversity	Main ASR dataset
AMI Meeting Corpus	Multi-speaker meetings	Conversational speech and speaker interaction	Complementary dataset

Together, Common Voice and AMI provide complementary strengths. Common Voice contributes scale, accent variation, and speaker diversity, while AMI contributes conversational structure and multi-speaker interaction. This combination supports the development and evaluation of a transcription model that must function not only on clean isolated speech, but also on realistic spoken interactions where variability and dialogue dynamics are present.

3.4.2 Model Selection

The model selected for the speech transcription module is Moonshine Tiny Streaming. This choice is mainly motivated by the real-time nature of the proposed system. Since the application is intended to process customer service calls as they occur, the transcription model must be lightweight, fast, and capable of producing usable text with minimal delay. In this context, a very large ASR model may provide strong accuracy, but it can also introduce latency and computational requirements that are less suitable for a desktop-based real-time prototype.

Moonshine is a family of automatic speech recognition models designed for low-latency transcription and voice interface applications. The Tiny variant is particularly relevant because it is optimized for resource-constrained environments while still maintaining competitive transcription performance[94].

The streaming version of Moonshine further supports the requirements of this thesis by processing audio incrementally instead of waiting for the full call to finish. This allows the system to generate partial or progressive transcription results during the conversation, which is essential for real-time customer service use cases. In addition, Moonshine Streaming uses architectural choices such as a lightweight audio frontend and a streaming encoder to support low-latency inference on edge-class hardware [95].

For these reasons, Moonshine Tiny Streaming is selected as a strong baseline for the proposed real-time processing pipeline.

3.4.3 Transcription Workflow

The processing pipeline of the speech transcription module is designed to prepare audio data for fine-tuning and evaluating the Moonshine Tiny Streaming model. The pipeline focuses on efficient data loading, audio-text preprocessing, model training, and transcription evaluation.

The first step consists of loading the pre-trained Moonshine Tiny Streaming model and its corresponding processor. The processor is responsible for converting raw audio signals and textual transcriptions into the format expected by the model. The use of a pre-trained model allows the system to start from an already learned speech representation instead of training an ASR model from scratch.

The second step concerns dataset preparation. In the experimental pipeline, speech samples are loaded in streaming mode in order to avoid downloading or storing the entire dataset at once. This is important for memory efficiency, especially when dealing with large-scale speech corpora. Before being used for training, the textual transcription is normalized by converting it to uppercase, removing unnecessary spacing, and ensuring a consistent textual format. Audio samples that exceed the maximum duration threshold are filtered out in order to maintain stable training and reduce computational cost.

After filtering and normalization, the selected samples are converted into Hugging Face dataset objects and the audio column is cast to a fixed sampling rate of 16 kHz. This step ensures that all audio inputs follow a consistent format. Each example is then transformed into three main components: the input audio values, the attention mask, and the target transcription labels. These processed elements represent the final training format used by the model.

A custom data collator is then used to prepare batches during training. Since speech recordings and transcriptions may have different lengths, padding is required to group multiple examples into the same batch.

To evaluate the model's transcription quality, the Word Error Rate (WER) is computed after applying a normalization step to both predictions and references.

This metric provides a direct measure of how many word-level errors are present in the generated transcript.

Overall, this processing pipeline ensures that the ASR model is prepared, trained, evaluated, and exported in a way that aligns with the real-time requirements of the proposed system.

3.4.4 Architecture

The architecture of the speech transcription module is built around the Moonshine Tiny Streaming model, accessed through the Hugging Face `AutoModelForSpeechSeq2Seq` and `AutoProcessor` interfaces. The model follows a sequence-to-sequence speech recognition design, where the input is an audio waveform and the output is a sequence of text tokens. This structure allows the transcription module to map continuous speech directly into written language.

At the input level, each training example is transformed into three main elements: `input_values`, `attention_mask`, and `labels`. The `input_values` represent the processed audio signal, the `attention_mask` indicates which parts of the sequence are valid input, and the `labels` represent the target transcription tokens used during training.

The model architecture can be understood as an encoder-decoder transcription system. The encoder processes the acoustic input and extracts meaningful speech representations from the waveform. These representations capture information related to phonetic content, temporal structure, and acoustic variation. The decoder then generates the corresponding transcription token by token, using the encoded speech representation as context. This design enables the model to learn the relationship between spoken audio patterns and their textual equivalents.

During inference, the architecture relies on the model's generation mechanism. The processed audio input is passed to the model, which produces a sequence of predicted token IDs. These IDs are then decoded by the processor into readable text.

The architecture is designed to balance transcription quality, speed, and resource efficiency. By using Moonshine Tiny Streaming as the core model, the system prioritizes low-latency transcription suitable for real-time customer service calls.

3.5 Dialogue Summarization Model

The text summarization model represents the final interpretation component of the proposed system. Its role is to transform the transcribed customer service call into a concise and informative summary that captures the essential content of the interaction. It aims to identify the most relevant information within the transcript.

The objective of the summarization module is not simply to shorten the transcript, but to produce a structured representation that is useful in a customer service context. This is especially important in situations where agents or supervisors may need to review previous interactions quickly without listening to the entire recording or reading the full transcript.

3.5.1 Datasets

The text summarization module requires dialogue-like textual data paired with concise summaries. However, customer service conversations are often available as raw transcripts rather than as fully annotated transcript-summary pairs. For this reason, some of the datasets used in this work are constructed through a teacher-model approach, where an existing summarization model is used to generate reference summaries for customer service dialogue data.

Before adapting the model to customer service data, two dialogue summarization datasets are used for initial fine-tuning: SAMSum and DialogSum. SAMSum contains messenger-style conversations paired with human-written summaries, making it useful for learning how to summarize short informal dialogues [96]. DialogSum provides dialogue-summary pairs across different everyday scenarios and is useful for learning general dialogue summarization behavior [97]. These datasets allow the summarization model to first learn the structure of dialogue summarization before being further adapted to customer service interactions.

Two main customer service data sources are then used to construct the domain-specific summarization dataset. The first is the *92k Real-World Call Center Scripts English* dataset, which contains a large collection of English call center conversations. This dataset is particularly relevant to the objective of this thesis because it reflects the structure and vocabulary of real customer service interactions, including inbound and outbound calls across different domains and speaker profiles [98]. The second source is the *Customer Support on Twitter* dataset, which contains large-scale customer support exchanges between users and companies on Twitter. Although this dataset is text-based rather than speech-based, it remains useful because it captures real customer complaints and support response patterns [99].

The full set of data sources used by the summarization module is presented in Table 3.3.

Table 3.3: Data sources used for the text summarization module

Source	**Data Type**	**Role in This Work**
SAMSum	Dialogue-summary pairs	Initial fine-tuning on informal dialogue summarization
DialogSum	Dialogue-summary pairs	Initial fine-tuning on general dialogue summarization
92k Real-World Call Center Scripts	Call center transcripts	Customer service domain adaptation using teacher-generated summaries
Customer Support on Twitter	Support exchanges	Customer service domain adaptation using teacher-generated summaries

A teacher model is then used to generate pseudo-reference summaries for the customer service datasets. In this work, the model `philschmid/bart-large-cnn-samsum` is used for this purpose. This model is based on BART, a denoising sequence-to-sequence transformer architecture designed for natural language generation and summarization tasks [83]. It is fine-tuned on the SAMSum dialogue summarization dataset, making it suitable for generating concise summaries from conversational text [100]. This strategy allows the creation of a task-specific summarization dataset aligned with the customer service context.

3.5.2 Model Selection

The model selection process for the text summarization module was guided by the need to generate concise, readable, and informative summaries from customer service transcripts. Since the objective is to produce a coherent reformulation of the interaction, the selected model must support abstractive generation.

FLAN-T5 was ultimately selected as the proposed summarization model. It is an instruction-fine-tuned version of T5, developed through large-scale instruction tuning across many tasks. This training strategy improves the model's ability to follow natural language instructions and generalize across different generation tasks [101]. In the context of this thesis, this is particularly useful considering the nature of daily spoken dialogues.

The choice of FLAN-T5 was also motivated by its higher capacity for new word generation during the experimental comparison. This ability is important for abstractive summarization because the model must synthesize the meaning of the conversation and express it in a more compact and readable form.

3.5.3 Summarization Workflow

The processing pipeline of the text summarization module is designed to fine-tune and evaluate a sequence-to-sequence model for generating concise summaries from customer service dialogues. The pipeline focuses on transforming raw dialogue text into a format suitable for abstractive summarization.

After loading the datasets, its data is divided into training, validation, and test splits. The text is then preprocessed and converted into model-ready inputs. Each dialogue is first normalized by removing unnecessary spacing and converting missing values into empty strings. Then, an instruction-style prefix is added to the input text. This prefix guides FLAN-T5 toward the desired task by explicitly asking it to summarize a customer service or dialogue transcript while including the main issue, important actions, and final outcome.

The tokenizer is then applied to both the dialogue inputs and the target

summaries. Dialogues are tokenized with a maximum input length, while summaries are tokenized with a shorter maximum target length.

A data collator is used to dynamically pad examples during batching. Since dialogue transcripts and summaries have variable lengths, padding is necessary to group multiple samples into the same batch. The general input-output flow followed by the FLAN-T5 summarization model is shown in Figure 3.4.

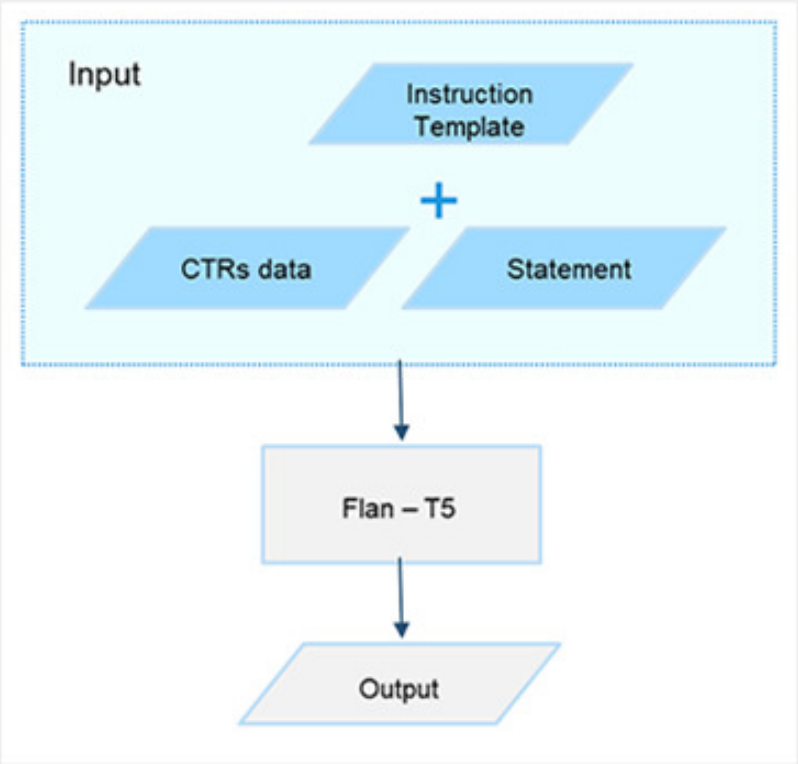


Figure 3.4: Flan T5 workflow [12]

After training, the model is evaluated on the test split by generating summaries from unseen dialogues. The generated summaries are compared with the reference summaries using ROUGE metrics, including ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-Lsum. In addition, the pipeline computes analysis indicators such as summary length, compression ratio, extractive overlap, and novel unigram ratio. These additional indicators are useful because they help determine whether the

model is merely copying words from the input or producing genuinely abstractive summaries.

Finally, the trained model and tokenizer are saved for later integration into the desktop application.

3.5.4 Architecture

The architecture of the text summarization module is based on a sequence-to-sequence transformer model, specifically FLAN-T5. In this module, the input consists of a customer service dialogue transcript, while the output is an abstractive summary generated by the model.

FLAN-T5 follows the encoder-decoder design from the T5 architecture. The encoder receives the tokenized dialogue transcript and converts it into contextual representations that capture the semantic content of the conversation. These representations allow the model to identify important elements such as the customer's issue, the agent's response, and the final outcome. The decoder then uses these encoded representations to generate the summary token by token, producing a new piece of text rather than simply extracting sentences from the original transcript.

In the implemented pipeline, the transcript is preceded by an instruction-style prompt that guides the model toward generating a customer service summary. since the model has been instruction-tuned to follow natural language commands across a variety of tasks.

At the preprocessing level, each dialogue is tokenized into a fixed maximum input length. Longer transcripts are truncated when they exceed this limit, while shorter ones are padded dynamically during batching. The target summaries are also tokenized separately and used as labels during training. This allows the model to learn the mapping between long customer service transcripts and their corresponding concise summaries.

During training, the model learns by comparing the generated token sequence with the reference summary. The optimization process adjusts the model parameters

so that the generated output becomes closer to the expected summary. Since the summaries used in this work are produced through a teacher-model approach, the architecture is trained to imitate concise dialogue summaries while adapting them to the customer service domain.

3.6 Conclusion

This chapter presented the design and methodology of the proposed system for real-time customer service call analysis. It described how raw call audio is transformed into three structured outputs: a transcript, a summary, and a dominant sentiment label.

The proposed system is organized around three main modules: speech-based sentiment analysis, speech transcription, and text summarization. A key design choice is the use of a dual-branch architecture, where the audio stream is analyzed directly for sentiment while also being converted into text for summarization. This allows the system to preserve both the emotional and factual dimensions of the call.

Overall, this chapter established the technical foundation of the proposed solution. The next chapter will focus on the implementation and evaluation of the system, using appropriate metrics for sentiment classification, transcription accuracy, and summarization quality.

Chapter 4

Implementation and Evaluation

4.1 Introduction

This chapter presents the implementation and evaluation of the proposed system. It explains how the main modules speech-based sentiment analysis, speech transcription, and text summarization were implemented, configured, and integrated into a desktop application for customer service call analysis.

The chapter also describes the development environment, tools, and libraries used during experimentation. Finally, it presents the evaluation of each module using appropriate metrics, including classification performance for sentiment analysis, transcription accuracy for ASR, and summarization quality for generated summaries.

4.2 Development Environment and Tools

The implementation of the proposed system required a set of programming languages, libraries, and development tools capable of supporting audio processing, deep learning, natural language processing, model evaluation, and desktop application development. Since the system combines three different tasks—speech sentiment analysis, speech transcription, and text summarization—the development environment had to be flexible enough to handle both audio-based and text-based

processing.

This section presents the main tools used during the implementation phase. These tools were selected according to their compatibility with machine learning workflows, their availability of pre-trained models, and their suitability for building an experimental desktop application. They contributed to the different stages of the project, including dataset preparation, model training, inference, evaluation, and final system integration.

4.2.1 Hardware and Execution Environment

The experimental development of the proposed system was mainly carried out using Kaggle Notebooks. Kaggle was selected because it provides a cloud-based execution environment with access to GPU acceleration, pre-installed machine learning libraries, and direct integration with datasets. This made it suitable for training deep learning models, testing preprocessing pipelines, and generating the pseudo-labeled summarization dataset.

Most of the training experiments and dataset generation steps were executed using Kaggle's dual NVIDIA T4 GPU accelerator. This configuration provides two T4 GPUs, allowing heavier model training and inference workloads to be executed more efficiently than on a CPU-only environment. The use of GPU acceleration was particularly important for the summarization module, where the teacher model was used to generate summaries for large collections of customer service dialogues, and for model fine-tuning tasks involving neural architectures.

In addition to GPU resources, the Kaggle environment provides a standard CPU and system memory configuration suitable for data loading, preprocessing, tokenization, batching, and evaluation. GPU-enabled Kaggle notebooks have been reported to provide increased memory and CPU resources, including 30 GB of RAM and 4 CPU cores in notebook configurations. These resources were sufficient for the experimental scope of this work, especially when combined with batch processing and model size choices adapted to the available hardware.

The use of a cloud-based notebook environment also helped maintain repro-

ducibility during experimentation. Each module could be developed and tested in isolated notebooks, with explicit dependencies, dataset paths, training parameters, and evaluation outputs. Once the models were trained or configured, the resulting checkpoints and generated files could be exported for later integration into the desktop application.

4.2.2 Kaggle

Kaggle is an online platform for data science and machine learning that provides access to datasets, notebooks, competitions, and cloud-based computational resources. It allows users to write and execute code directly in the browser, making it a practical environment for experimentation, model training, and result analysis.

In this work, Kaggle was used as the main experimental environment for training models and generating the summarization dataset. Its GPU-enabled notebooks, particularly the dual NVIDIA T4 configuration, provided the computational support required for running deep learning models and processing large datasets. Kaggle was also useful for organizing experiments in separate notebooks, storing intermediate outputs, and testing different preprocessing and training configurations before integrating the models into the final system.

4.2.3 Python

Python is a high-level programming language widely used in data science, artificial intelligence, and machine learning because of its simplicity, readability, and rich ecosystem of scientific libraries. It provides strong support for numerical computation, data manipulation, model development, and automation, making it suitable for building complete machine learning pipelines.

In this work, Python was used as the main programming language for implementing the different components of the system. It was used for dataset preparation, audio preprocessing, model training, evaluation, and inference. Python also provided access to the main libraries and frameworks used in the project, including deep learning frameworks, Hugging Face tools, audio processing libraries, and evaluation

packages.

4.2.4 Deep Learning Frameworks

Two deep learning frameworks were used in this work, each selected according to the requirements of the corresponding module. The choice of each framework was guided by its compatibility with the implemented architecture as well as the tools required for both training and inference.

- **TensorFlow/Keras:** TensorFlow is an open-source deep learning framework used for building and training neural network models, while Keras provides a higher-level interface that simplifies model design and experimentation. In this work, TensorFlow and Keras were used for the speech emotion recognition module, where the MA-CapsNet-based architecture was implemented and trained. Their model-building tools made them suitable for defining convolutional layers, pooling operations, capsule-related components, optimization procedures, and classification evaluation.
- **PyTorch:** PyTorch is an open-source deep learning framework widely used for research and experimentation because of its flexible tensor operations, GPU support, and dynamic computation graph. In this work, PyTorch was used mainly for the speech transcription and text summarization modules. Through its compatibility with the Hugging Face ecosystem, it supported the loading, fine-tuning, and inference of transformer-based models such as Moonshine Tiny Streaming for ASR and FLAN-T5 for summarization. It also provided GPU acceleration, tensor operations, and training utilities needed for experimentation.

The use of both frameworks allowed each component to be implemented with the tools most appropriate for its architecture.

4.2.5 Hugging Face Ecosystem

Hugging Face is an open-source platform and ecosystem for machine learning that provides access to pre-trained models, datasets, tokenizers, processors, and training tools. It is widely used in natural language processing and speech-related tasks because it simplifies the process of loading, adapting, and evaluating transformer-based models.

In this work, the Hugging Face ecosystem was used mainly for the speech transcription and text summarization modules. The `Transformers` library was used to load and fine-tune pre-trained models such as Moonshine Tiny Streaming for speech transcription and FLAN-T5 for text summarization. It also provided the corresponding processors and tokenizers needed to convert raw audio and textual inputs into model-compatible representations. For the summarization module, the `Seq2SeqTrainer` framework was used to manage training, validation, generation, and model saving.

The Hugging Face `Datasets` library was also used to load, process, split, and manage datasets during experimentation. It allowed the training data to be transformed into a consistent format before being passed to the models. Overall, the Hugging Face ecosystem helped reduce implementation complexity by providing ready-to-use tools for model loading, preprocessing, fine-tuning, and inference.

4.2.6 Librosa

Librosa is a Python library for audio and music signal analysis. It provides tools for loading audio files, processing waveform data, and extracting acoustic features commonly used in speech and audio-based machine learning tasks.

In this work, Librosa was used as the main audio processing library for the speech emotion recognition module. It supported audio loading and feature extraction before training and inference. More specifically, it provided the tools needed to read speech files, handle audio signals, and extract the acoustic features used as input to the sentiment analysis model.

Librosa was used to extract Mel-Frequency Cepstral Coefficients (MFCCs) and Zero-Crossing Rate (ZCR). These features were selected because they provide compact representations of the spectral and signal-level characteristics of speech. The extracted features were then normalized and prepared as input matrices for the MA-CapsNet-based speech emotion recognition model.

4.2.7 Data Processing and Evaluation Libraries

Several libraries were used to organize datasets, prepare inputs, split data, and evaluate the performance of the implemented models. These tools supported the experimental workflow across the three modules, including sentiment classification, speech transcription, and text summarization.

- **NumPy:** NumPy is a Python library for numerical computing and array manipulation. In this work, it was used for operations such as reshaping, normalization, padding, and preparing feature matrices before passing them to the models.
- **Pandas:** Pandas is a Python library for data manipulation and analysis, especially for tabular data. It was used to handle CSV files and organize datasets, transcripts, labels, and generated summaries by loading, cleaning, filtering, and restructuring them into the required format.
- **Scikit-learn:** Scikit-learn is a machine learning library that provides tools for data preprocessing, model evaluation, and traditional machine learning workflows. In this work, it was used for data splitting, label encoding, class weighting, and classification evaluation metrics.
- **JiWER:** JiWER is a Python library used to evaluate automatic speech recognition systems by computing error-based transcription metrics. It was used to calculate the Word Error Rate (WER) for the speech transcription module.
- **ROUGE:** ROUGE is a set of evaluation metrics commonly used in automatic summarization. It measures the overlap between generated summaries

and reference summaries, and was used to evaluate the quality of the text summarization module.

- **Matplotlib:** Matplotlib is a Python library for data visualization. It was used to plot experimental results, including training curves and other performance-related outputs.

4.2.8 Desktop Application Development

Several libraries and tools were used to build the desktop application and connect the different system components. These tools supported the graphical interface, audio capture, background processing, local storage, and communication between the interface and the AI modules.

- **PySide6:** PySide6 is the official Python binding for the Qt framework. It was used to build the graphical user interface of the desktop application, including windows, buttons, text areas, tables, dialogs, stacked pages, and layout managers. It allowed the application to include several pages such as live session processing, file processing, call history, and settings.
- **Qt Style Sheets (QSS):** QSS is a styling system used by Qt applications to customize the appearance of interface elements. In this project, it was used to give the application a consistent visual design, including custom buttons, rounded panels, themed cards, and a dashboard-like layout.
- **SoundCard:** SoundCard is a Python library used for audio input and output handling. It was used to access the microphone input and Windows WASAPI loopback devices, allowing the system to record both the agent's microphone stream and the customer's audio output from external call applications.
- **SQLite:** SQLite is a lightweight, file-based database engine suitable for local applications. It was used to store call-related information such as session records, audio file paths, transcripts, summaries, emotion results, customer phone numbers, and metadata.

- **SQLAlchemy:** SQLAlchemy is a Python library that provides tools for database interaction through an object-oriented interface. It was used as an abstraction layer over SQLite, making it easier to define database models and manage stored call records.
- **subprocess:** The `subprocess` module is part of the Python standard library and is used to run external processes from a Python program. In this application, it was used to separate heavy or time-sensitive tasks, such as audio recording and model inference, from the graphical interface process.
- **threading and queue:** The `threading` and `queue` modules are Python standard libraries used for concurrent execution and safe communication between tasks. They were used to manage asynchronous communication between the interface, controllers, and background workers while keeping the application responsive.
- **JSON:** JSON is a lightweight data format used for structured communication. In this project, JSON messages were used to exchange information between the main application and persistent model worker processes, making the communication simple, readable, and easier to debug.

These tools allowed the desktop application to remain modular and responsive. The interface layer handles user interaction, while recording, model inference, and database operations are managed through separate services and background workers.

4.3 Implementation and Evaluation of the Sentiment Analysis Module

This section presents the implementation and evaluation of the speech-based sentiment analysis module. The objective of this module is to classify the dominant emotional state expressed in an audio sample using acoustic features extracted directly from speech.

The experiments focus on preparing the selected emotional speech datasets, configuring the training process, evaluating the model using classification metrics, and analyzing its performance across multi-dataset conditions.

4.3.1 Dataset Description

A wide variety of datasets across languages was used during implementation. All contain annotated raw audio (.wav) files across multiple speakers for better data diversity.

- **Emo-DB:** The Berlin Emotional Speech Database contains 535 acted German utterances recorded by 10 professional speakers, including 5 male and 5 female speakers. It covers seven emotional classes: anger, boredom, disgust, fear, happiness, sadness, and neutral speech [86].
- **SAVEE:** The Surrey Audio-Visual Expressed Emotion dataset contains 480 English emotional utterances recorded by 4 male speakers. It includes seven emotion categories: anger, disgust, fear, happiness, sadness, surprise, and neutral speech [87].
- **RAVDESS:** The Ryerson Audio-Visual Database of Emotional Speech and Song contains acted emotional recordings from 24 professional actors, including 12 male and 12 female speakers. In the speech subset, the dataset includes 1,440 audio recordings covering neutral, calm, happy, sad, angry, fearful, disgust, and surprised expressions [88].
- **CASIA:** The CASIA Chinese emotional speech dataset contains Mandarin emotional speech recordings produced by 4 speakers. It includes six emotion classes: anger, fear, happiness, neutral, sadness, and surprise. This dataset was used as an additional evaluation resource because it introduces a different language and speaker setting.
- **CREMA-D:** The Crowd-sourced Emotional Multimodal Actors Dataset contains 7,442 emotional speech clips produced by 91 actors from different

demographic backgrounds. It includes six emotion classes: anger, disgust, fear, happiness, neutral, and sadness [89].

Emo-DB, SAVEE, and RAVDESS formed the main experimental corpora, while CASIA and CREMA-D were used to further examine the model's behavior on additional speech conditions.

4.3.2 Data Splitting

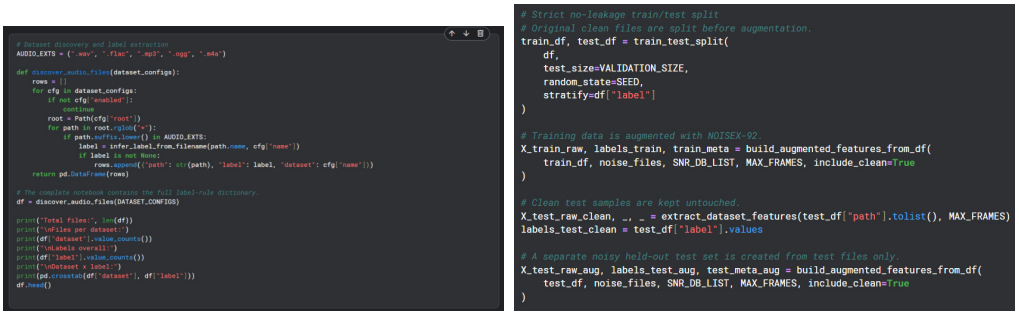
The sentiment analysis datasets were divided using an 80%–20% split, where 80% of the samples were used for training and 20% were reserved for testing. The split was applied after loading the audio files, extracting their labels from the dataset-specific naming formats, and mapping the labels into the unified emotion label set. The test subset was kept separate and used only for final evaluation. Table 4.1 summarizes the approximate number of samples assigned to the training and testing subsets for each main sentiment dataset.

Table 4.1: Approximate train-test split for the main sentiment datasets

Dataset	**Total**	**Train 80%**	**Test 20%**
Emo-DB	535	428	107
SAVEE	480	384	96
RAVDESS	1440	1152	288
CASIA	1200	960	240

For the noise-augmented experiments, the split was performed before augmentation. The original clean files were first divided into training and testing subsets. After that, NOISEX-92 noise was added only to the training files at different signal-to-noise ratio levels. The test files were not used during augmentation for training. This avoided data leakage, where different noisy versions of the same original utterance could appear in both the training and testing subsets.

The implementation outputs of the data discovery and splitting steps are shown in Figure 4.1, where the loaded datasets and the resulting train-test partitions are verified before training.



```
# Dataset discovery and label extraction
AUDIO_EXTS = {".wav", ".flac", ".mp3", ".ogg", ".m4a"}

def discover_audio_files(dataset_configs):
    rows = []
    for cfg in dataset_configs:
        if not cfg["enabled"]:
            continue
        root = Path(cfg["root"])
        for path in root.rglob("*"):
            if path.suffix.lower() in AUDIO_EXTS:
                label = infer_label_from_filename(path.name, cfg["name"])
                if label is not None:
                    rows.append({"path": str(path), "label": label, "dataset": cfg["name"]})
    return pd.DataFrame(rows)

# The complete notebook contains the full label-rule dictionary.
df = discover_audio_files(DATASET_CONFIGS)

print("Total files:", len(df))
print("\nFiles per dataset:")
print(df["dataset"].value_counts())
print("\nLabels overall:")
print(df["label"].value_counts())
print("\nDataset x label:")
print(pd.crosstab(df["dataset"], df["label"]))
df.head()
```

```
# Strict no-leakage train/test split
# Original clean files are split before augmentation.
train_df, test_df = train_test_split(
    df,
    test_size=VALIDATION_SIZE,
    random_state=SEED,
    stratify=df["label"]
)

# Training data is augmented with NOISEX-92.
X_train_raw, labels_train, train_meta = build_augmented_features_from_df(
    train_df, noise_files, SNR_DB_LIST, MAX_FRAMES, include_clean=True
)

# Clean test samples are kept untouched.
X_test_raw_clean, _, _ = extract_dataset_features(test_df["path"].tolist(), MAX_FRAMES)
labels_test_clean = test_df["label"].values

# A separate noisy held-out test set is created from test files only.
X_test_raw_aug, labels_test_aug, test_meta_aug = build_augmented_features_from_df(
    test_df, noise_files, SNR_DB_LIST, MAX_FRAMES, include_clean=True
)
```

Figure 4.1: Data discovery and splitting for the sentiment analysis model

In the final evaluation, two types of test results were considered: clean held-out test performance and noisy held-out test performance. The clean test set measured the model's performance on unseen original speech, while the noisy test set measured its robustness when unseen speech was degraded with background noise.

4.3.3 Preprocessing Phase

The preprocessing phase was applied before feature extraction in order to make the audio files consistent across the different emotional speech datasets.

Each dataset-specific label was then mapped into a unified emotion label set. This step was necessary because the datasets do not use exactly the same class names. For example, some datasets contain *calm* or *boredom*, which are not available in all other corpora. The final label mapping was used to keep the training process consistent across the selected datasets.

The audio samples were processed using the same feature extraction settings. Each recording was divided into short frames using a 25 ms window and a 10 ms shift. These values were chosen to follow the MA-CapsNet-DA feature extraction setup. After framing, the extracted features were normalized to reduce differences caused by recording conditions, speaker loudness, and dataset-specific variations.

For the noise robustness experiments, data augmentation was applied after the train-test split. Clean training samples were mixed with noise from the NOISEX-92 dataset at different signal-to-noise ratio levels. The SNR values used were -10

dB, −5 dB, 0 dB, 5 dB, and 10 dB. The test set was kept separate from this augmentation process in order to avoid data leakage.

The preprocessing phase produced a clean and standardized set of audio samples with unified emotion labels.

4.3.4 Feature Extraction Phase

In this phase, the preprocessed audio signals were transformed into acoustic descriptors using the `Librosa` library. The model did not use the raw waveform directly; instead, each audio sample was represented through Mel-Frequency Cepstral Coefficients (MFCCs) and Zero-Crossing Rate (ZCR). These features were extracted frame by frame and then combined to form the final input representation used by the sentiment analysis model. The feature extraction implementation is shown in Figure 4.2, where MFCC and ZCR features are extracted from each audio sample using Librosa.

```
# Feature extraction: 20 MFCC + 1 ZCR
def load_audio(path, sr=SR):
    y, _ = librosa.load(path, sr=sr, mono=True)
    y, _ = librosa.effects.trim(y, top_db=30)
    if len(y) == 0:
        y, _ = librosa.load(path, sr=sr, mono=True)
    return y.astype(np.float32)

def extract_llds(y, sr=SR):
    mfcc = librosa.feature.mfcc(
        y=y, sr=sr, n_mfcc=N_MFCC,
        n_fft=N_FFT, hop_length=HOP_LENGTH, win_length=N_FFT
    )
    zcr = librosa.feature.zero_crossing_rate(
        y=y, frame_length=N_FFT, hop_length=HOP_LENGTH
    )
    return np.vstack([mfcc, zcr]).T.astype(np.float32)
```

Figure 4.2: Feature extraction

MFCCs describe the short-term spectral shape of a speech signal. They are widely used in speech processing because they provide a compact representation of the frequency content of the voice while approximating human auditory perception. In this work, 20 MFCC coefficients were extracted from each frame using Librosa's MFCC extraction function:

```
librosa.feature.mfcc(y=y, sr=sr, n_mfcc=20)
```

The Zero-Crossing Rate measures how frequently the audio waveform crosses the zero-amplitude axis within a frame. For a discrete signal $x[n]$ of length N, it can be defined as:

$$ZCR = \frac{1}{2N} \sum_{n=1}^{N-1} |\text{sgn}(x[n]) - \text{sgn}(x[n-1])|$$

where $\text{sgn}(x[n])$ represents the sign of the signal at sample n. A zero crossing occurs when two consecutive samples have different signs. In this work, ZCR was extracted using the following Librosa function:

```
librosa.feature.zero_crossing_rate(y)
```

For each frame, one ZCR value was combined with the 20 MFCC coefficients, resulting in a 21-dimensional feature vector. The final feature matrices were then normalized to zero mean and unit variance before being passed to the MA-CapsNet-based model.

4.3.5 Model Learning

The sentiment analysis model was implemented and trained using TensorFlow/Keras. The learning process was based on the adapted MA-CapsNet architecture described previously, where the input of the model is a normalized acoustic feature matrix composed of MFCC and ZCR values. The output layer contains one digital capsule

for each emotion class, and the predicted class is obtained from the capsule with the highest vector length.

The model was trained using the Adam optimizer with a learning rate of 0.001. The loss function used was the capsule margin loss, following the MA-CapsNet-DA formulation. The margin parameters were set as follows:

$$m^+ = 0.9, \quad m^- = 0.1, \quad \lambda = 0.26$$

Dynamic routing was applied between the primary capsule layer and the digital capsule layer. The number of routing iterations was set to 2. This allowed the lower-level capsules to send their outputs to the most relevant emotion capsules during training.

The learning process was monitored using classification performance metrics, mainly accuracy, precision, recall, and F1-score.

4.3.6 Evaluation Metrics

The sentiment analysis module was evaluated as a multi-class classification task. The predicted emotion label for each audio sample was compared with its reference label, and the model performance was measured using accuracy, precision, recall, F1-score, and the confusion matrix.

4.3.6.1 Accuracy

Accuracy measures the proportion of correctly classified samples among all tested samples. It provides a general view of the model's overall classification performance.

$$Accuracy = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

4.3.6.2 Precision

Precision measures the proportion of correct predictions for a given class among all samples predicted as that class. In this work, it was used to evaluate how reliable the model's predicted emotion labels were.

$$Precision = \frac{TP}{TP + FP}$$

4.3.6.3 Recall

Recall measures the proportion of correctly detected samples for a given class among all samples that truly belong to that class. It was used to evaluate how well the model detected each emotion category.

$$Recall = \frac{TP}{TP + FN}$$

4.3.6.4 F1-score

The F1-score combines precision and recall into a single value using their harmonic mean. It is useful when the class distribution is not perfectly balanced.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

4.3.6.5 Confusion Matrix

The confusion matrix was used to show the distribution of correct and incorrect predictions across the emotion classes. Each row represents the true emotion class, while each column represents the predicted class. This makes it possible to identify which emotions are classified correctly and which emotions are most often confused with each other.

4.3.7 Experiments and Results

The sentiment analysis experiments were organized into three main phases. The first phase evaluated the modified MA-CapsNet-DA architecture on clean multi-dataset configurations and compared the reproduced results with the reference paper. The second phase compared several architectural variants in order to identify the most promising direction for improvement. The third phase evaluated the proposed enhanced architecture on a larger and more diverse dataset configuration, including both clean and noisy test conditions.

4.3.7.1 Phase 1: Paper-Based Architecture on Clean Multi-Dataset Configurations

The first phase evaluated the modified MA-CapsNet-DA architecture on several clean multi-dataset configurations against the paper's model. The objective was to compare the proposed model's behavior under different emotional speech corpora and to verify its performance under similar dataset combinations to those reported in the reference work.

The experiments were conducted without noise augmentation. The data was divided using an 80%–20% split. The results reported in Table 4.2 correspond to the clean held-out test sets.

Table 4.2: Phase 1 results using the paper-based MA-CapsNet-DA architecture on clean data

Configuration	**Train**	**Test**	**Accuracy**	**Macro F1**
SAVEE + Emo-DB	812	203	74.38%	72%
CASIA + SAVEE	1344	336	75.60%	75%
CASIA + Emo-DB	1388	347	78.39%	77%
CASIA + SAVEE + Emo-DB	1772	443	74.04%	72%

The best result in this phase was obtained using the CASIA + Emo-DB configuration, which included 1,388 training samples and 347 testing samples across eight emotion classes: anger, boredom, disgust, fear, happiness, neutral, sadness,

and surprise. Table 4.3 provides the detailed class-level performance for this best Phase 1 configuration.

Table 4.3: Classification report for the CASIA + Emo-DB clean test set

Class	**Precision**	**Recall**	**F1-score**	**Support**
Anger	0.80	0.94	0.87	65
Boredom	0.75	0.94	0.83	16
Disgust	0.67	0.67	0.67	9
Fear	0.67	0.69	0.68	54
Happiness	0.90	0.67	0.77	54
Neutral	0.91	0.88	0.89	56
Sadness	0.71	0.70	0.70	53
Surprise	0.76	0.78	0.77	40
Macro avg	0.77	0.78	0.77	347
Weighted avg	0.79	0.78	0.78	347

The strongest class-level results were obtained for neutral, anger, and boredom. Lower scores were observed for disgust and fear, indicating that these classes were more difficult for the model to separate under this configuration.

In contrast, the largest configuration, CASIA + SAVEE + Emo-DB, achieved an accuracy of 74.04%. This indicates that increasing the amount of data does not necessarily lead to improved performance, especially when the merged datasets introduce variations in speakers, languages, recording conditions, and class distributions.

4.3.7.2 Phase 2: Architecture Comparison

The second phase consisted of testing different architecture modifications around the modified MA-CapsNet-DA structure. The objective was to identify which architectural direction gave the best performance before selecting the final proposed model. All tested models used the same input shape of $350 \times 21 \times 1$, corresponding to 350 frames and 21 acoustic features per frame.

This phase was carried out on the merged Emo-DB + SAVEE configuration. The combined dataset contained 1,015 clean speech samples. An 80%–20% split

was used, resulting in approximately 812 samples for training and 203 samples for testing. The results reported in Table 4.4 correspond to the clean held-out test set.

Table 4.4: Phase 2 architecture comparison on the Emo-DB + SAVEE clean test set

Architecture	**Test Data**	**Accuracy**	**Macro F1**
Modified MA-CapsNet	Emo-DB + SAVEE	79.31%	78.49%
SE Attention	Emo-DB + SAVEE	76.35%	75.29%
BiLSTM Variant	Emo-DB + SAVEE	50.74%	46.62%
Multi-Scale Variant	Emo-DB + SAVEE	71.43%	71.28%
Residual Multi-Scale Variant	Emo-DB + SAVEE	71.20%	70.79%
Capsule Dropout	Emo-DB + SAVEE	79.64%	79.31%
Residual CapsDrop	Emo-DB + SAVEE	77.50%	77.25%
Hybrid CapsDrop	Emo-DB + SAVEE	72.14%	71.49%
Gated Hybrid CapsDrop	Emo-DB + SAVEE	71.79%	71.25%
Dynamic Gated Hybrid CapsDrop	Emo-DB + SAVEE	71.07%	70.67%

The results of this second phase show that adding architectural complexity did not always improve performance. Attention-based, recurrent, and multi-scale variants did not outperform the modified model. The BiLSTM-based variant gave the weakest result, suggesting that the tested additions were not suitable for this feature representation and training setup.

Among the tested modifications, the capsule dropout variant achieved the best result, with an accuracy of 79.64% and a macro F1-score of 79.31%. This result suggested that regularization inside the capsule-based architecture was more useful than adding heavier recurrent or attention-based components. Therefore, capsule dropout was retained as an important direction for the following experiments.

4.3.7.3 Phase 3: Proposed Enhanced Architecture on Multi-Dataset and Noisy Data

The third phase evaluated the proposed enhanced architecture on a larger multi-dataset configuration. The objective was to test whether the enhanced architecture

could improve classification performance while keeping the same general capsule-based design.

This experiment used the merged RAVDESS + SAVEE + Emo-DB configuration. The dataset contained 3,895 clean audio files distributed across eight emotion classes. The data was split using an 80%–20% split. In addition to clean training data, NOISEX-92 augmentation was applied to the training set using three signal-to-noise ratio levels: 0 dB, 5 dB, and 10 dB. The test set was kept separate from the augmentation process.

The proposed enhanced architecture achieved a clean held-out test accuracy of 93.45%. On the noisy held-out test set, the model obtained an accuracy of 88.84%. The detailed results are shown in Table 4.5.

Table 4.5: Phase 3 performance of the proposed enhanced architecture

Test Condition	**Samples**	**Accuracy**	**Macro F1**	**Weighted F1**
Clean held-out test set	779	93.45%	92%	93%
Noisy held-out test set	35834	88.84%	87%	89%

The clean test results show that the proposed enhanced architecture achieved strong performance across most emotion classes. High classification scores were obtained for neutral, sadness, fear, happiness, surprise, anger, and disgust. In contrast, the boredom class achieved the lowest performance, which can be attributed to its lower number of training samples compared to the other classes, leading to a class imbalance effect. The detailed class-level results of the proposed enhanced architecture on the clean test set are presented in Table 4.6.

Table 4.6: Classification report of the proposed enhanced architecture on the clean test set

Class	**Precision**	**Recall**	**F1-score**	**Support**
Anger	0.91	0.93	0.92	114
Boredom	0.80	0.75	0.77	16
Disgust	0.96	0.87	0.91	98
Fear	0.93	0.96	0.94	103
Happiness	0.93	0.94	0.94	103
Neutral	0.94	0.97	0.96	155
Sadness	0.94	0.95	0.95	101
Surprise	0.96	0.92	0.94	89
Macro avg	0.92	0.91	0.92	779
Weighted avg	0.93	0.93	0.93	779

For the noisy evaluation, the model was tested on held-out samples degraded with NOISEX-92 noise. As expected, performance was lowest at 0 dB, where the noise level is strongest, and improved as the SNR increased. The effect of noise level on model performance is illustrated in Figure 4.3.

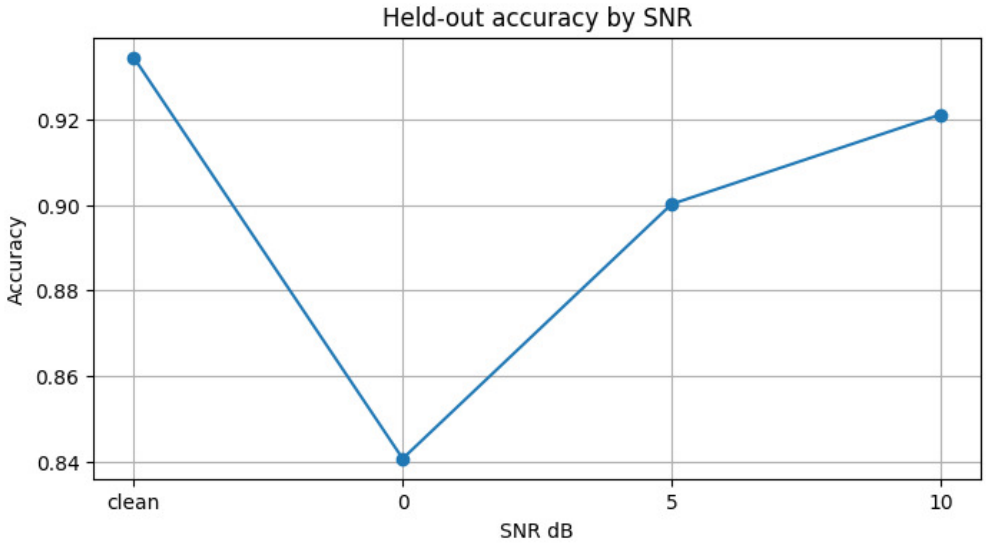


Figure 4.3: Model performance at different SNR

These results show that the proposed enhanced architecture achieved the best

overall performance among the tested sentiment analysis configurations.

4.3.8 Comparison and Discussion

The three experimental phases show the progression from reference architecture evaluation to architectural exploration and final model validation. In the first phase, the modified MA-CapsNet-DA architecture was tested on several clean multi-dataset configurations and compared with the results reported in the reference paper. This established a baseline for understanding how the architecture behaved across different dataset combinations. The comparison between the reproduced implementation and the reference paper results is summarized in Table 4.7.

Table 4.7: Comparison between the paper results and the reproduced results on clean multi-dataset configurations

Dataset Configuration	**Paper Result**	**Our Result**	**Difference**
SAVEE + Emo-DB	67.18%	74.38%	+7.20%
CASIA + Emo-DB	74.02%	78.39%	+4.37%
CASIA + SAVEE	67.74%	75.60%	+7.86%
CASIA + SAVEE + Emo-DB	66.96%	74.04%	+7.08%

The comparison shows that our reproduced implementation achieved higher results in the first phase than those reported in the reference paper across all four clean multi-dataset configurations, with an improvement of approximately 7% on most configurations.

In the second phase, several modifications were tested around the MA-CapsNet-DA architecture. The results showed that simply adding more components did not necessarily improve performance. In particular, additions such as recurrent layers or more complex hybrid branches reduced performance, suggesting that they disrupted the flow of information between the convolutional and capsule layers rather than improving the learned representation.

The capsule dropout variant produced the best result in the second phase. This result was important because it indicated that the capsule part of the architecture was the main source of overfitting. Adding dropout inside the capsule-based

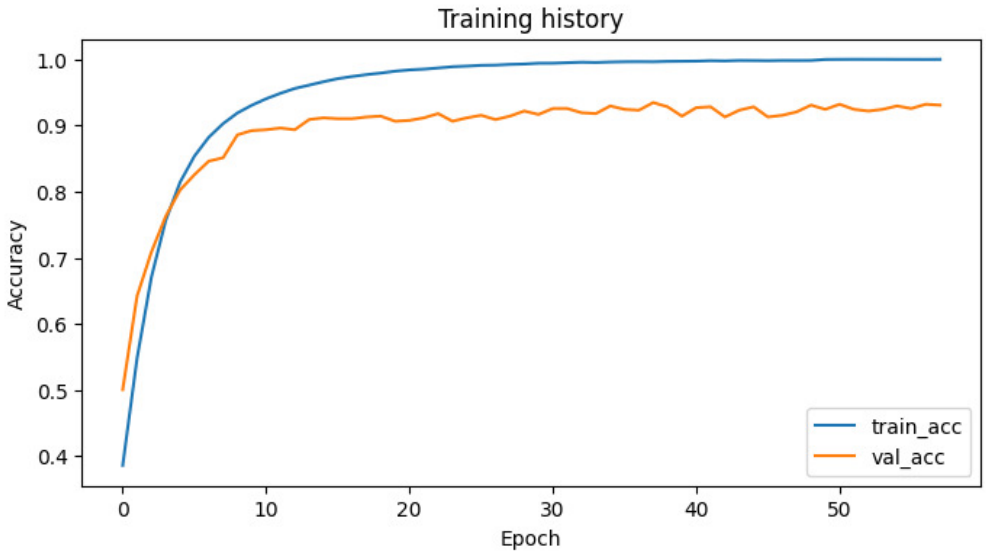
structure improved generalization, while modifying the convolutional front-end or adding heavier processing blocks did not provide the same benefit. For this reason, the final architecture kept the general MA-CapsNet-DA structure but focused on improving the capsule-based learning capacity in a controlled way.

The third phase focused on the strengths of the proposed enhanced architecture. This version is denser than the original paper-based MA-CapsNet-DA architecture, meaning that it contains more trainable parameters while preserving the same general capsule-based design. The goal was not to make the model unnecessarily large, but to increase its capacity enough to learn from more complex mixed datasets. This choice proved effective: on the RAVDESS + SAVEE + Emo-DB configuration, the proposed model achieved 93.45% accuracy on the clean held-out test set and 88.84% accuracy on the noisy held-out test set.

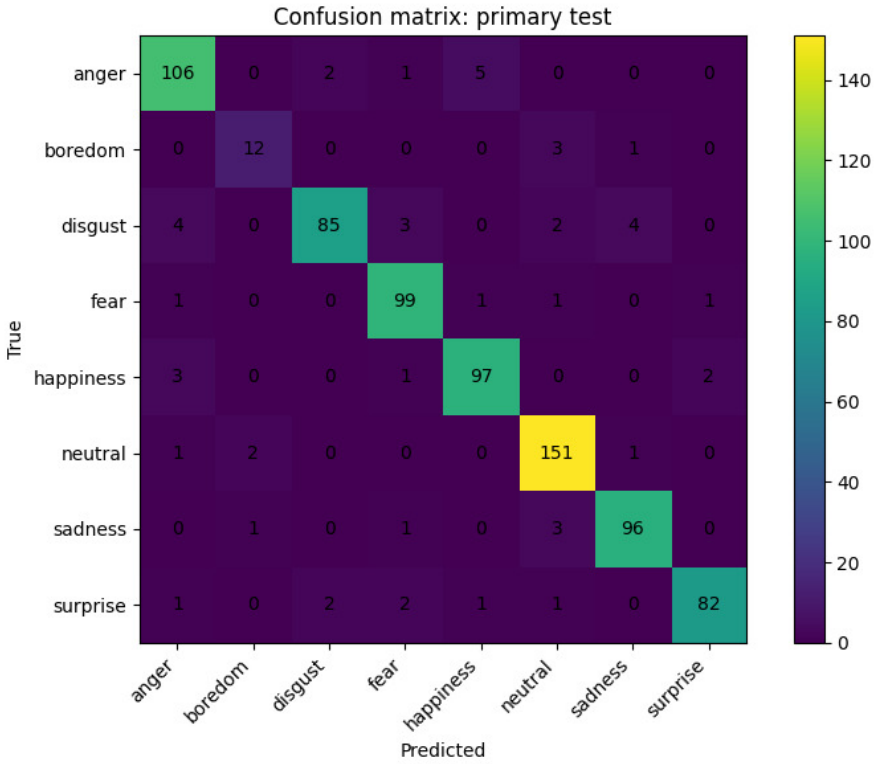
The final training behavior and confusion matrix of the proposed sentiment analysis model are shown in Figure 4.4.

The choice of replacing CASIA with RAVDESS in the final mixed configuration was also important. CASIA introduced language diversity through Mandarin Chinese speech, but RAVDESS provided greater speaker balance and a more structured emotional speech corpus. It includes both male and female speakers and a larger number of validated acted emotional recordings.

The experiments show that the final model was selected based on both performance and behavior across different conditions. The paper-based MA-CapsNet-DA architecture provided a strong starting point, but the proposed denser adaptation performed better when the training data became more varied and complex. The results also show that the capsule network component must be regularized carefully to avoid overfitting. Therefore, the final architecture represents a balance between preserving the lightweight structure of the reference model and increasing its learning capacity for multi-dataset speech emotion recognition.



(a) Validation curves



(b) Confusion Matrix

Figure 4.4: Performance of the proposed sentiment analysis model

4.4 Implementation and Evaluation of the Speech Transcription Module

This section presents the implementation and evaluation of the speech transcription module. The objective of this module is to convert speech audio into written text using the Moonshine Tiny Streaming model. In the proposed system, this module provides the transcript that is later used by the dialog summarization component. The implementation focuses on evaluating the model Word Error Rate.

4.4.1 Dataset Description

The speech transcription module was evaluated using Mozilla Common Voice. This dataset contains speech recordings paired with their corresponding textual transcriptions, making it suitable for assessing the performance of this model.

- **Mozilla Common Voice:** Mozilla Common Voice is a crowdsourced speech corpus containing audio recordings from a wide range of speakers. It includes diverse accents, speaking styles, and recording conditions, making it suitable for evaluating the robustness of automatic speech recognition (ASR) systems [92].

In this work, the Common Voice dataset was used exclusively for evaluating the transcription module. The objective was to assess the ability of the selected ASR model to generate accurate and readable transcriptions from unseen speech samples.

4.4.2 Test Data Selection

Since the Moonshine Tiny Streaming model was used in its pre-trained form, no training or validation split was created. Instead, a subset of Mozilla Common Voice was selected as the test data. Each selected sample contained the audio recording and its reference transcription.

The selected test samples were used only for inference and evaluation. The model generated a transcription for each audio sample, and the generated text was then compared using WER with the reference transcription provided by the dataset.

4.4.3 Preprocessing Phase

The preprocessing phase prepared the selected Common Voice samples for inference and evaluation. Each sample was checked to ensure that it contained both a valid audio recording and a reference transcription. Samples with missing audio or missing text were not considered for evaluation.

The mapping of Common Voice samples into the evaluation format is shown in Figure 4.5.

```
[ ]:
ACCENT_MAP = {
    "us": "United States English",
    "england": "England English",
    "canada": "Canadian English",
    "australia": "Australian English",
    "scotland": "Scottish English",
    "india": "India and South Asia English",
    "south asia": "India and South Asia English",
    "": "Unknown / not reported",
}

normalizer = Compose([
    ToLowerCase(),
    RemovePunctuation(),
    RemoveMultipleSpaces(),
    Strip()
])

def normalize_text(text):
    return normalizer(str(text))

def map_accent(accent):
    accent = str(accent).lower().strip()
    for key, value in ACCENT_MAP.items():
        if key and key in accent:
            return value
    return "Unknown / not reported"
```

Figure 4.5: Data mapping for the transcription model

When necessary, the audio was converted to a sampling rate of 16 kHz. This ensured that the input signal matched the expected format of the pre-trained model.

The reference transcriptions were also normalized before evaluation. Text normalization was applied to reduce differences caused by formatting rather than transcription quality. This included removing unnecessary spaces and applying a consistent text format before computing Word Error Rate.

4.4.4 Model Loading and Input Representation

The ASR module was implemented using PyTorch and the Hugging Face Transformers library. It did not use manual acoustic feature extraction such as MFCCs. Instead, the audio samples were processed using the Moonshine processor provided through the Hugging Face ecosystem. This processor converts the raw waveform into the input representation expected by the model.

For each audio sample, the processor prepared the audio input and converted it into model-compatible tensors. These processed inputs were then passed to the pre-trained Moonshine Tiny Streaming model for transcription generation.

The output of the model was a sequence of predicted token IDs. These token IDs were decoded by the processor into readable text, producing the final predicted transcription for each audio sample.

4.4.5 Evaluation Metrics

The speech transcription module was evaluated using Word Error Rate (WER). It is one of the most common metrics used for automatic speech recognition evaluation.

4.4.5.1 Word Error Rate

WER is computed using the number of substitutions, deletions, and insertions required to transform the predicted transcription into the reference transcription.

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

where S is the number of substituted words, D is the number of deleted words, I is the number of inserted words, and N is the total number of words in the reference transcription.

Figure 4.6 shows the implementation used to compute WER during the ASR evaluation.

```
[ ]: # Corpus-level WER and CER
     def corpus_metrics(df, label):
         ok_df = df[df["status"] == "ok"].copy()

         refs = ok_df["reference_norm"].tolist()
         preds = ok_df["prediction_norm"].tolist()

         return {
             "split": label,
             "samples": len(ok_df),
             "wer": wer(refs, preds),
             "cer": cer(refs, preds),
             "mean_sample_wer": ok_df["wer"].mean(),
             "mean_sample_cer": ok_df["cer"].mean(),
         }

     overall_metrics = pd.DataFrame([
         corpus_metrics(dev_results, "Development"),
         corpus_metrics(test_results, "Test"),
         corpus_metrics(results_df, "Development + Test")
     ])

     display(overall_metrics)
```

	split	n	wer	cer	mean_sample_wer	median_sample_wer	mean_sample_cer	median_sample_cer
0	dev	4076	0.124746	0.065174	0.129397	0.0	0.066955	0.0
1	test	3995	0.118561	0.062401	0.123981	0.0	0.065280	0.0
2	dev+test	8071	0.121675	0.063797	0.126716	0.0	0.066126	0.0

Figure 4.6: WER Implementation

A lower WER indicates better transcription performance. In this work, WER was computed using the `JiWER` library after applying normalization to both the predicted and reference transcriptions.

4.4.6 Experiments and Results

The speech transcription module was evaluated using the pre-trained Moonshine Tiny Streaming model on the Mozilla Common Voice English development and test subsets. No additional fine-tuning was performed. The objective of this experiment was to measure the transcription quality of the selected ASR model on unseen speech samples using Word Error Rate (WER) and Character Error Rate (CER).

A total of 8,071 audio samples were evaluated: 4,076 samples from the development subset and 3,995 samples from the test subset. The evaluation loop used to generate predictions and compare them with the reference transcriptions is shown in Figure 4.7.

```python
# Evaluate one dataframe split
def evaluate_dataframe(df, split_name):
    rows = []

    for _, row in tqdm(df.iterrows(), total=len(df), desc=f"Evaluating {split_name}"):
        reference = row["sentence"]
        audio_path = row["audio_path"]

        try:
            prediction = transcribe(audio_path)
            ref_norm = normalize_text(reference)
            pred_norm = normalize_text(prediction)

            rows.append({
                "split": split_name,
                "audio_path": audio_path,
                "accent_group": row["accent_group"],
                "reference": reference,
                "prediction": prediction,
                "reference_norm": ref_norm,
                "prediction_norm": pred_norm,
                "wer": wer(ref_norm, pred_norm),
                "cer": cer(ref_norm, pred_norm),
                "status": "ok"
            })
        except Exception as e:
            rows.append({
                "split": split_name,
                "audio_path": audio_path,
                "accent_group": row["accent_group"],
                "reference": reference,
                "prediction": "",
                "reference_norm": normalize_text(reference),
                "prediction_norm": "",
                "wer": np.nan,
                "cer": np.nan,
                "status": f"error: {e}"
            })

    return pd.DataFrame(rows)

dev_results = evaluate_dataframe(dev_df, "dev")
test_results = evaluate_dataframe(test_df, "test")
results_df = pd.concat([dev_results, test_results], ignore_index=True)

print("Evaluated samples:", len(results_df))
print("Successful samples:", (results_df["status"] == "ok").sum())
print("Errors:", (results_df["status"] != "ok").sum())
```

Figure 4.7: Transcription Model Evaluation Implemetation

The overall transcription results are presented in Table 4.8.

Table 4.8: Overall ASR evaluation results on Common Voice

Split	**Samples**	**WER**	**CER**	**Mean Sample WER**
Development	4076	0.1247	0.0652	0.1294
Test	3995	0.1186	0.0624	0.1240
Development + Test	8071	0.1217	0.0638	0.1267

The model obtained a WER of 0.1247 on the development subset and 0.1186 on the test subset. When both subsets were combined, the overall WER was 0.1217 and the overall CER was 0.0638. These results show that the model produced accurate transcriptions without task-specific training on the selected evaluation data.

The evaluation was also analyzed by reported speaker accent. Accent groups with very few samples were excluded from the grouped analysis to avoid unstable results. Table 4.9 presents the main accent-based results on the combined development and test subsets.

Table 4.9: ASR results by accent on Common Voice development and test subsets

Accent	**Samples**	**WER**	**CER**
Canadian English	183	0.0660	0.0295
England English	626	0.0929	0.0448
United States English	1270	0.1009	0.0497
Unknown / not reported	5383	0.1293	0.0683
Australian English	195	0.1567	0.0834
Scottish English	63	0.1719	0.0915
India and South Asia English	180	0.2229	0.1217

The variation in transcription performance across accent groups is visualized in Figure 4.8.

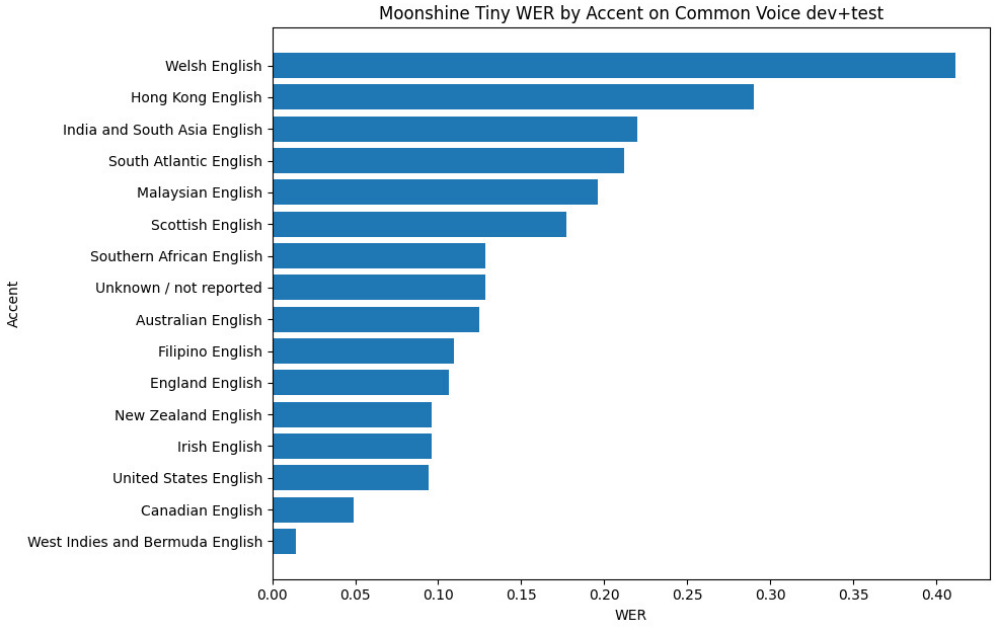


Figure 4.8: ASR evaluation results on different accents

The accent-based results show that performance varied across speaker groups. It kept accurate results on the most widely spoken accents while performing worse on some of the more rare or challenging ones.

The notebook also identified examples with perfect transcription, where the predicted text exactly matched the reference transcription after normalization. These cases show that the model can produce fully correct outputs for many short and clearly spoken samples.

The experiment shows that Moonshine Tiny Streaming provides strong transcription performance for a lightweight ASR model. Its WER of 0.1186 on the Common Voice test subset supports its use in the proposed system, especially because the model was evaluated without additional fine-tuning.

4.4.7 Discussion

The results show that Moonshine Tiny Streaming provides a strong basis for the speech transcription module. Without additional fine-tuning, the model achieved a

WER of 0.1186 on the Common Voice test subset and an overall WER of 0.1217 across the development and test subsets. This indicates that the model can generate accurate transcriptions from unseen speech samples while remaining lightweight enough for integration into the proposed desktop application.

The accent-based evaluation also shows that the model works across several English accent groups. The best results were obtained on Canadian English, England English, and United States English, with WER values of 0.0660, 0.0929, and 0.1009 respectively. Higher error rates were observed for Australian English, Scottish English, and India and South Asia English. This variation shows that accent remains an important factor in transcription quality, but the model still remained usable across different speaker groups.

Another important point is deployability. Since the model was used in its pre-trained form, no training process was required for the ASR component. This reduces implementation complexity and makes the module easier to integrate into the final system. The model can be loaded locally, process audio through its processor, and return text output that can be directly displayed or passed to the summarization module.

Its main advantage is the balance between transcription quality, model size, and deployment feasibility. However, future improvements could include fine-tuning on real customer service calls, adding stronger noise handling, and testing the model on more conversational and multi-speaker audio.

4.5 Implementation and Evaluation of the Text Summarization Module

This section presents the implementation and evaluation of the text summarization module. The objective of this module is to generate a concise summary from a customer service transcript while preserving the main issue, important actions, and final outcome of the interaction.

The experiments focus on preparing dialogue-summary pairs, preprocessing and tokenizing the text, fine-tuning FLAN-T5, and evaluating generated summaries using ROUGE metrics and additional analysis indicators.

4.5.1 Dataset Description

The summarization module was trained using dialogue-like textual data paired with summaries. Two types of data were used: existing dialogue summarization datasets and customer service datasets with teacher-generated summaries.

- **SAMSum:** SAMSum contains messenger-style dialogues paired with human-written summaries. It was used to expose the model to informal dialogue summarization [96].
- **DialogSum:** DialogSum contains real-life scenario dialogues paired with summaries. It was used to support general dialogue summarization behavior [97].
- **92k Real-World Call Center Scripts English:** This dataset contains customer service call transcripts. Since it does not provide human-written summaries, summaries were generated using the teacher model `philschmid/bart-large-cnn-samsu` [98, 100].
- **Customer Support on Twitter:** This dataset contains customer support exchanges from Twitter. It was considered for domain adaptation because it contains real customer complaints and company responses [99].

The length distribution of the call center transcripts is shown in Figure 4.9, which helps explain why long-input handling is important for this module.

```
Long transcript word counts:

> 256 words: 84827 (93.95%)

> 512 words: 52744 (58.41%)

> 768 words: 27435 (30.38%)

> 1024 words: 15136 (16.76%)

> 1500 words: 5909 (6.54%)

> 2000 words: 2555 (2.83%)

> 3000 words: 555 (0.61%)
```

Figure 4.9: Length of the 92k Real-World Call Center Scripts

The final training format used two main columns: `transcript` and `summary`. The transcript column represented the input dialogue, while the summary column represented the target output.

4.5.2 Data Splitting

For the generated customer service dataset, the data was shuffled and divided into training, validation, and test subsets using a 90%–5%–5% split. The training set was used to fine-tune the model, the validation set was used to monitor performance and select the best checkpoint, and the test set was reserved for final evaluation.

The implementation of the dataset split is shown in Figure 4.10.

```
[ ]:
# Create train, validation, and test splits
df = df.sample(frac=1.0, random_state=SEED).reset_index(drop=True)

train_df = df.iloc[:TRAIN_SIZE].copy()
val_df = df.iloc[TRAIN_SIZE:TRAIN_SIZE + VAL_SIZE].copy()
test_df = df.iloc[TRAIN_SIZE + VAL_SIZE:TRAIN_SIZE + VAL_SIZE + TEST_SIZE].copy()

raw_datasets = DatasetDict({
    "train": Dataset.from_pandas(train_df[["transcript", "summary"]], preserve_index=False),
    "validation": Dataset.from_pandas(val_df[["transcript", "summary"]], preserve_index=False),
    "test": Dataset.from_pandas(test_df[["transcript", "summary"]], preserve_index=False),
})

print("Train samples:", len(train_df))
print("Validation samples:", len(val_df))
print("Test samples:", len(test_df))
```

Figure 4.10: Data splitting implementation for the summarization model

Table 4.10 summarizes the split ratios used for training, validation, and final testing.

Table 4.10: Data split used for the text summarization module

Subset	**Ratio**	**Role**
Training set	90%	Model fine-tuning
Validation set	5%	Model selection and early stopping
Test set	5%	Final evaluation

4.5.3 Preprocessing Phase

The preprocessing phase cleaned and standardized the dialogue-summary pairs before tokenization. Samples with missing transcripts or missing summaries were removed. Very short or low-quality generated summaries were also filtered out to reduce noise in the training data.

The text was normalized by removing unnecessary spacing and formatting artifacts. Duplicate examples were removed to prevent repeated samples from affecting the learning process. After cleaning, the dataset was organized into a consistent input-output format. The filtering and cleaning steps applied to the summarization dataset are shown in Figure 4.11.

Each transcript was then preceded by an instruction-style prompt. This prompt asked the model to summarize a customer service or dialogue transcript while including the main issue, important actions, and final outcome. This format was

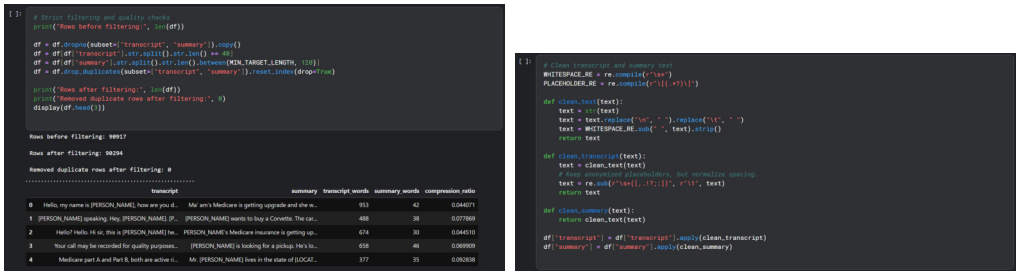


Figure 4.11: Data filtering and cleaning for the summarization model

used because FLAN-T5 is instruction-tuned and responds well to task descriptions written in natural language.

4.5.4 Tokenization Phase

The tokenization phase converted the cleaned text into model-compatible token IDs. The FLAN-T5 tokenizer was applied separately to the input transcripts and target summaries. Figure 4.12 shows the tokenizer loading step used to prepare the text data for FLAN-T5.

```
[ ]: # Load tokenizer and estimate input token lengths
tokenizer = AutoTokenizer.from_pretrained(TWITTER_CHECKPOINT_PATH)

TOKEN_DIAG_SAMPLE_SIZE = 2000
sample_df = df.sample(min(TOKEN_DIAG_SAMPLE_SIZE, len(df)), random_state=SEED)

token_lengths = sample_df["transcript"].apply(
    lambda x: len(tokenizer(str(x), truncation=False)["input_ids"])
)

print("Token length diagnostics on sample:", len(sample_df))
display(token_lengths.describe())
print("Inputs over 512 tokens:", int((token_lengths > 512).sum()))

Token length diagnostics on sample: 2000
count     2000.000000
,mean      1303.112000
,std        813.786704
,min        125.000000
,25%        783.000000
,50%       1082.000000
,75%       1551.500000
,max      10306.000000
,dtype: float64
> 512 tokens: 1901 (95.05%)

> 768 tokens: 1529 (76.45%)

> 1024 tokens: 1097 (54.85%)

> 1500 tokens: 531 (26.55%)

> 2048 tokens: 273 (13.65%)
```

Figure 4.12: Tokenizer Loading

The maximum input length was set to 1024 tokens. Transcripts longer than

this limit were truncated. The maximum target summary length was set to 128 tokens. This value was used to keep summaries concise and suitable for customer service review. The main tokenization parameters are summarized in Table 4.11.

Table 4.11: Tokenization configuration for the text summarization module

Tokenizer	FLAN-T5 tokenizer
Maximum input length	1024 tokens
Maximum target length	128 tokens

A `DataCollatorForSeq2Seq` was used during batching. It dynamically padded the input and target sequences so that examples of different lengths could be grouped into the same batch.

4.5.5 Model Learning

The summarization model was implemented using PyTorch and the Hugging Face Transformers library. The model checkpoint used in the experiment was `google/flan-t5-small`. The training process was managed using `Seq2SeqTrainer`. The loading of the FLAN-T5 model checkpoint is shown in Figure 4.13.

```
[ ]:
# Load the fine-tuned FLAN-T5 model
def clean_memory():
    gc.collect()
    if torch.cuda.is_available():
        torch.cuda.empty_cache()

clean_memory()

model = AutoModelForSeq2SeqLM.from_pretrained(TWITTER_CHECKPOINT_PATH)
model.to(device)

total_params = sum(p.numel() for p in model.parameters())
trainable_params = sum(p.numel() for p in model.parameters() if p.requires_grad)

print("Model loaded from checkpoint.")
print(f"Total parameters: {total_params:,}")
print(f"Trainable parameters: {trainable_params:,}")

Model loaded from checkpoint.

Total parameters: 76,961,152

Trainable parameters: 76,961,152
```

Figure 4.13: Model loading for the summarization model

The model was trained for a maximum of 20 epochs. Early stopping was used with a patience value of 3. The best model checkpoint was selected according to

the validation ROUGE-L score. The learning rate was set to 2×10^{-5}, and the weight decay was set to 0.01. Table 4.12 presents the main learning configuration used during fine-tuning.

Table 4.12: Learning configuration of the text summarization module

Parameter	**Value**
Model checkpoint	`google/flan-t5-small`
Maximum epochs	20
Learning rate	2×10^{-5}
Weight decay	0.01
Train batch size	4
Evaluation batch size	4
Gradient accumulation steps	2
Early stopping patience	3
Best model metric	ROUGE-L
Generation beams	4
Maximum generation length	128 tokens
Mixed precision	FP16

During evaluation, the model generated summaries using beam search with 4 beams. The generated summaries were then compared with the reference summaries using automatic evaluation metrics. The ROUGE metric implementation used for summarization evaluation is shown in Figure 4.14.

```
[ ]:
# ROUGE metric
rouge_metric = evaluate.load("rouge")

def compute_metrics(eval_pred):
    predictions, labels = eval_pred
    decoded_preds = tokenizer.batch_decode(predictions, skip_special_tokens=True)

    labels = np.where(labels != -100, labels, tokenizer.pad_token_id)
    decoded_labels = tokenizer.batch_decode(labels, skip_special_tokens=True)

    result = rouge_metric.compute(
        predictions=decoded_preds,
        references=decoded_labels,
        use_stemmer=True
    )
    result = {k: round(v, 4) for k, v in result.items()}
    result["gen_len"] = round(
        np.mean([len(pred.split()) for pred in decoded_preds]), 4
    )
    return result
```

Figure 4.14: ROUGE metrics

4.5.6 Evaluation Metrics

The text summarization module was evaluated using ROUGE metrics. These metrics compare the generated summary with the reference summary by measuring word and sequence overlap.

4.5.6.1 ROUGE-1

ROUGE-1 measures unigram overlap between the generated summary and the reference summary. It reflects how many important individual words are shared between the two texts.

4.5.6.2 ROUGE-2

ROUGE-2 measures bigram overlap. It is stricter than ROUGE-1 because it evaluates whether pairs of consecutive words are preserved.

4.5.6.3 ROUGE-L

ROUGE-L is based on the longest common subsequence between the generated summary and the reference summary. It captures sentence-level similarity without requiring exact consecutive word matches.

4.5.6.4 ROUGE-Lsum

ROUGE-Lsum is a variant of ROUGE-L commonly used for summarization evaluation. It is suitable for evaluating summaries composed of one or more sentences.

In addition to ROUGE, several analysis indicators were computed, including input length, summary length, compression ratio, extractive overlap, and novel unigram ratio. These indicators were used to examine whether the model produced concise and abstractive summaries rather than simply copying large parts of the input transcript.

4.5.7 Experiments and Results

The summarization experiments were organized as a progressive fine-tuning pipeline. The model was first adapted to general dialogue summarization using DialogSum, then further fine-tuned on SAMSum. After that, the model was adapted to customer service style using the Twitter customer support dataset, and finally fine-tuned on the generated call transcript-summary dataset.

4.5.7.1 Progressive Fine-Tuning Results

Table 4.13 presents the main evaluation results obtained at each stage of the summarization pipeline. ROUGE-L was used as the main validation metric for model selection.

Table 4.13: Summarization results across the fine-tuning stages

Stage	**Split**	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**	**ROUGE-Lsum**
DialogSum	Validation	0.4353	0.1840	0.3542	0.3546
DialogSum	Test	0.3997	0.1462	0.3199	0.3198
SAMSum	Validation	0.4775	0.2373	0.3956	0.3956
SAMSum	Test	0.4622	0.2184	0.3826	0.3822
Twitter Customer Support	Validation	0.5745	0.4053	0.5230	0.5231
Twitter Customer Support	Test	0.5712	0.4004	0.5198	0.5196
Call Transcripts	Validation	0.4225	0.2465	0.3466	0.3467
Call Transcripts	Test	0.4213	0.2446	0.3472	0.3472

The final call transcript stage improved the model's performance on the call center data compared with the previous Twitter-adapted checkpoint. Before final fine-tuning, the model obtained a validation ROUGE-L score of 0.2403 on the call transcript validation set. After fine-tuning on the generated call transcript-summary dataset, the validation ROUGE-L increased to 0.3466, while the test ROUGE-L reached 0.3472. This shows that the final fine-tuning stage helped the model adapt more effectively to the structure and vocabulary of call center transcripts.

4.5.7.2 Final Call Transcript Experiment

The final experiment used the customer call transcript dataset generated with the teacher-model approach. The model was trained on 70,000 samples, validated on 3,000 samples, and tested on 3,000 samples. The input was processed using

a head-tail strategy, where the beginning and the end of long transcripts were preserved in order to keep both the opening context and the final resolution of the call. The final call transcript fine-tuning setup is summarized in Table 4.14.

Table 4.14: Final call transcript summarization setup

Parameter	**Value**
Training samples	70,000
Validation samples	3,000
Test samples	3,000
Input mode	Head-tail
Head tokens	768
Tail tokens	640
Maximum epochs	4
Effective batch size	16
Best model metric	ROUGE-L
Best validation ROUGE-L	0.3468

The final test results of the call transcript summarization model are presented in Table 4.15.

Table 4.15: Final test results of the call transcript summarization model

Metric	**Value**
Evaluation loss	1.0675
ROUGE-1	0.4213
ROUGE-2	0.2446
ROUGE-L	0.3472
ROUGE-Lsum	0.3472
Average generated length	51.3440 tokens

Table 4.16 reported additional generation statistics that were also computed on the final test set.

Table 4.16: Generation statistics of the final summarization model

Statistic	**Value**
Average ROUGE-1	0.4583
Average ROUGE-2	0.2682
Average ROUGE-L	0.3711
Average ROUGE-Lsum	0.3711
Average prediction length	42.058 words
Average prediction compression	0.0788
Average extractive overlap ratio	0.8553
Average novel unigram ratio	0.1447
Number of truncated inputs	0
Training time	15,633.39 s

The generated summaries were generally concise and focused on the main issue and the agent response. The detailed generation results also show that the head-tail input strategy helped preserve useful parts of the call transcript without producing truncated inputs in the final evaluation. This is important because customer service calls can be long and may contain important information both at the beginning of the interaction and near the end, where the final outcome is usually discussed.

4.5.8 Discussion

The summarization results show that the progressive fine-tuning strategy helped the model gradually adapt from general dialogue summarization to customer service-oriented summarization. The first two stages, using DialogSum and SAMSum, allowed the model to learn how to summarize dialogue structure in general.

An important observation is that the fine-tuned FLAN-T5 model performed better than the teacher model `philschmid/bart-large-cnn-samsum` on the SAMSum evaluation stage, while being much smaller and less resource-demanding. The result therefore supports the choice of FLAN-T5 for the proposed system, since it provides a better balance between summarization performance, model size, and deployment feasibility. The fine-tuning behavior of the summarization model is shown in Figure 4.15.

```
[ ]:
start_time = time.time()

train_result = trainer.train()

training_time = time.time() - start_time
print("Training finished.")
print("Training time seconds:", round(training_time, 2))
print("Training time minutes:", round(training_time / 60, 2))

trainer.save_model(str(FINAL_MODEL_DIR))
tokenizer.save_pretrained(str(FINAL_MODEL_DIR))

print("Final call-transcript model saved to:", FINAL_MODEL_DIR)
```

```
Training finished.

Training time seconds: 15633.39

Training time minutes: 260.56
Final call-transcript model saved to: outputs\flan_t5_final_customer_call_transcripts\final_model
```

	stage	eval_loss	eval_rouge1	eval_rouge2	eval_rougeL	eval_rougeLsum	eval_gen_len
0	twitter_checkpoint_on_call_validation	1.599828	0.2939	0.1450	0.2403	0.2403	56.3053
1	call_finetuned_validation	1.059855	0.4225	0.2465	0.3466	0.3467	51.0217
2	call_finetuned_test	1.067549	0.4213	0.2446	0.3472	0.3472	51.3440

Figure 4.15: Fine tuning results for the summarization model

This balance is particularly relevant for the desktop application developed in this work. A smaller summarization model is easier to load, faster to execute, and more suitable for integration with the other system components.

The strongest ROUGE scores were obtained after fine-tuning on the Twitter customer support dataset. This result is expected because this dataset is closer to the customer service domain than DialogSum and SAMSum. It contains real customer complaints, company responses, and support-oriented exchanges, which helped the model learn a style closer to the target application.

After the final fine-tuning stage on call center transcripts, the model became better adapted to longer and more realistic customer service conversations. The final call transcript results were lower than the Twitter customer support results, but this is expected because the call transcript dataset is more difficult. It contains longer conversations, more repetitions, placeholders, and less structured dialogue compared with short Twitter exchanges. An example of a generated call summary is shown in Figure 4.16.

```
# Custom customer-service call test
custom_call = """
Customer: I ordered a pair of headphones two weeks ago. The tracking page says delivered,
but I never received the package.
Agent: I am sorry about that. I will open an investigation with the carrier.
Customer: How long will that take?
Agent: It usually takes two business days. If the package is not found, we will ship a replacement.
"""

start = time.time()
summary = generate_summary(custom_call)
generation_time = time.time() - start

print("Input tokens:", len(tokenizer(custom_call)["input_ids"]))
print("Generation time:", round(generation_time, 3), "sec")
print("\nGenerated summary:")
print(summary)
```

```
Input tokens: 201

Generation time: 0.357 sec


Generated summary:

Customer ordered a pair of headphones two weeks ago and the tracking page says delivered, but he never received them. The investigation
usually takes 2 business days. If the package is not found, they will ship a replacement at no extra cost.
```

Figure 4.16: Example summary

The summarization module produced stable results on the final test set and was able to generate concise summaries from long customer service transcripts. The head-tail strategy helped preserve important information from both the beginning and the end of each call, which is useful because customer service conversations usually introduce the problem early and discuss the outcome near the end. Future improvements could include using a long-context summarization model or summarizing calls in smaller chunks before generating a final global summary.

4.6 System Interface and Deployment

This section presents the desktop application developed to integrate the three main components of the proposed system: speech transcription, speech-based sentiment analysis, and dialogue summarization. This part describes how the models were connected into a usable application and how the final outputs are presented to the user.

The application was implemented as a local desktop system using Python and PySide6. Its role is to provide a graphical interface for recording or processing

customer service calls, running the AI modules, displaying the generated transcript, summary, and sentiment results, and saving completed sessions for later review. The dashboard and settings pages of the desktop application are shown in Figure 4.17.

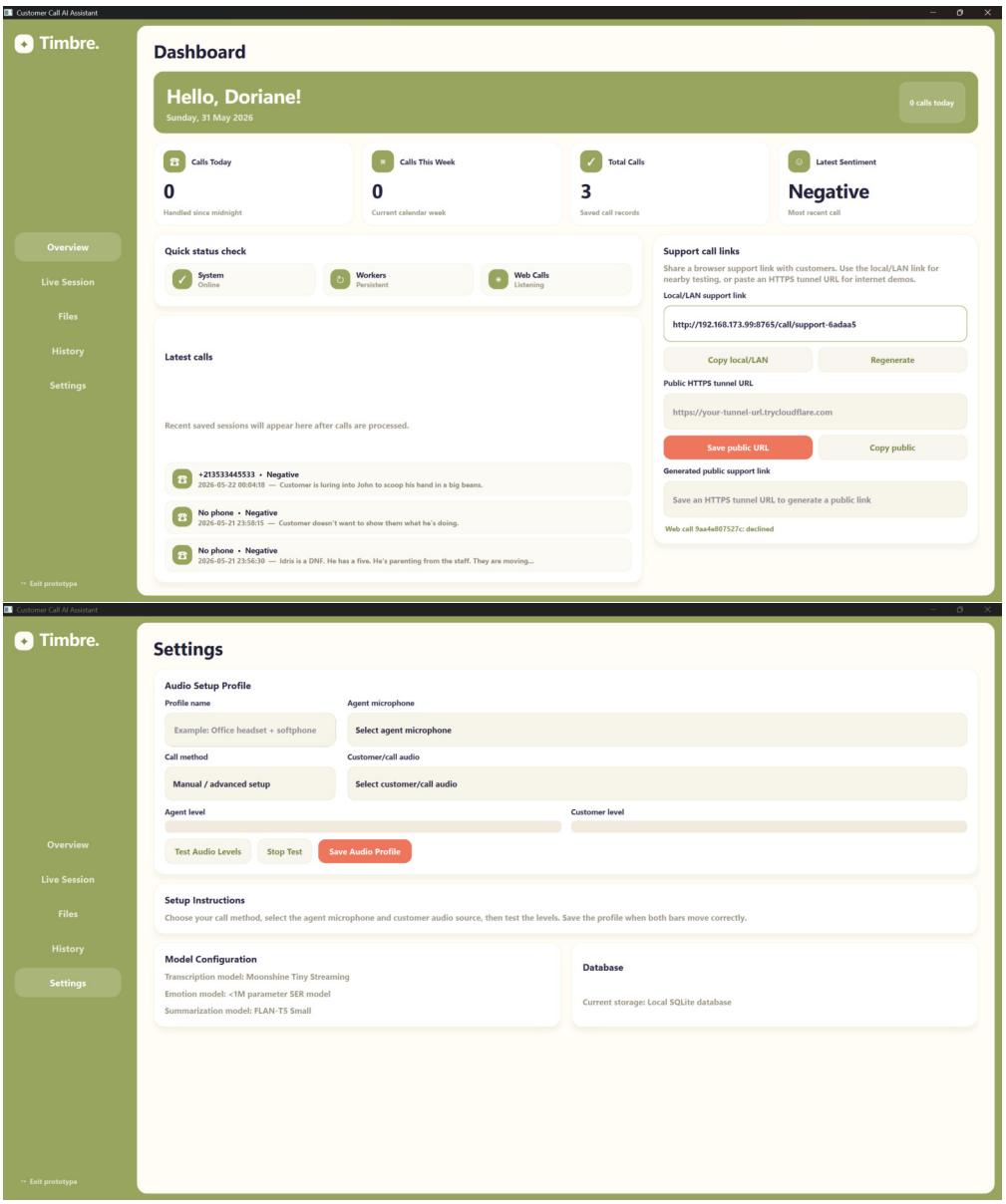


Figure 4.17: Application Dashboard and Settings

4.6.1 Desktop Application Architecture

The desktop application follows a modular architecture composed of four main layers: the user interface layer, the controller layer, the service layer, and the worker/process layer. This separation was used to keep the interface organized and to prevent heavy processing tasks from blocking the graphical interface. The main classes involved in the desktop application implementation are presented in Figure 4.18.

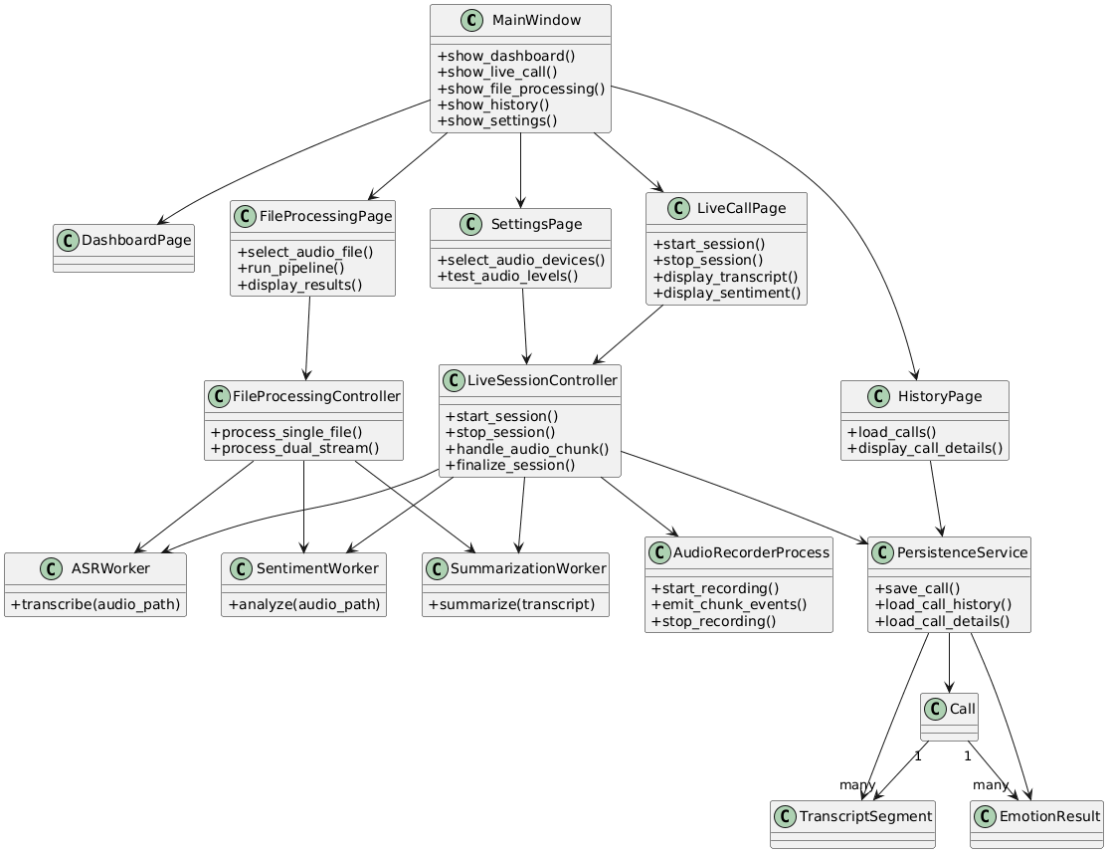


Figure 4.18: Class diagram of the desktop application implementation

The user interface layer was implemented with PySide6. It contains the main window and the different application pages, including the dashboard page, live call page, file processing page, call history page, and settings page. The interface allows the user to configure audio devices, start and stop live sessions, process recorded

files, and review previous calls.

The controller layer connects the interface with the rest of the system. It includes controllers for recording, live transcription, live emotion analysis, live summarization, file processing, audio setup, and session persistence. These controllers receive user actions from the interface and forward the required tasks to the appropriate services or background workers.

The worker/process layer is responsible for executing bigger tasks outside the main graphical interface. Audio recording is handled by a separate recorder process, while the transcription, emotion recognition, and summarization models are accessed through persistent AI server processes. Communication with these processes is done through JSON messages sent through standard input and output streams. The interaction between the interface, controllers, workers, models, and database is summarized in Figure 4.19.

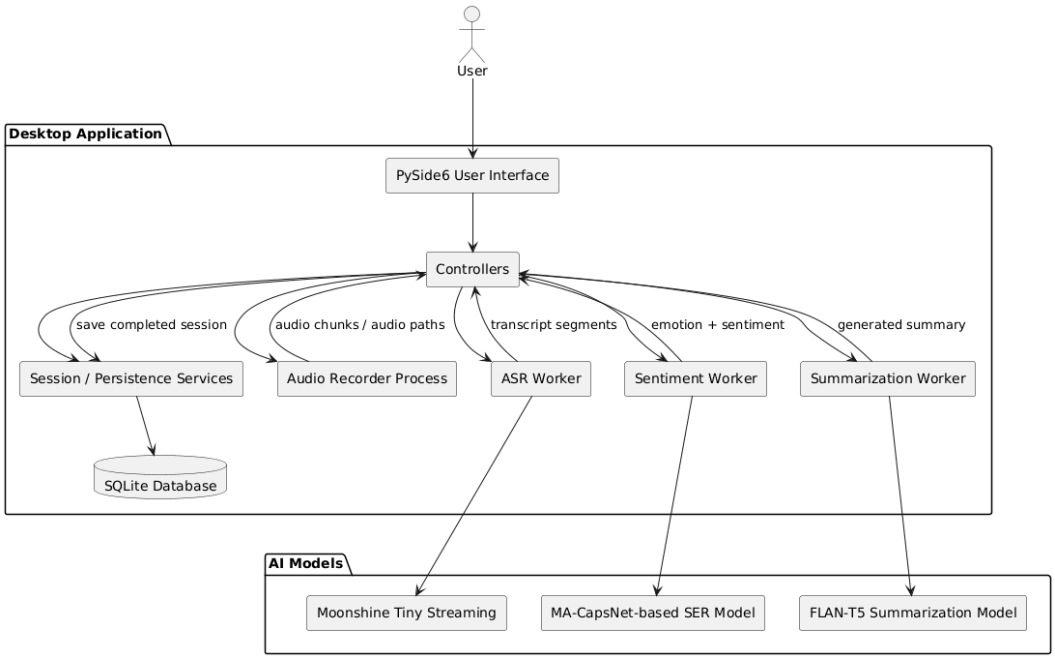


Figure 4.19: Component diagram of the desktop application architecture

Table 4.17 summarizes the main architectural layers of the application and the

role of each layer.

Table 4.17: Main architectural layers of the desktop application

Layer	**Role**
User interface	Displays pages, buttons, inputs, results, and call history
Controllers	Connect user actions to recording, inference, and storage logic
Services	Manage call sessions, persistence, transcription, emotion, and summaries
Workers / subprocesses	Execute recording and AI inference outside the GUI thread
Database	Stores completed calls, transcripts, summaries, and emotion results

This architecture keeps the application responsive during live recording and model inference. The interface remains responsible for interaction and display, while audio capture and AI processing are executed separately in background threads or subprocesses.

4.6.2 Audio Capture and Processing Flow

The application supports dual-stream audio recording. This means that the agent audio and the customer or call audio are recorded separately. The agent stream is captured from a selected microphone input, while the customer stream can be captured from a selected customer audio source, including a Windows WASAPI loopback device when the call audio comes from a browser or desktop softphone.

Recording is handled by a dedicated subprocess using the `SoundCard` library. The recorder writes two audio files: one for the agent stream and one for the customer stream at a default sampling rate of 48 kHz. The live call analysis interface is shown in Figure 4.20.

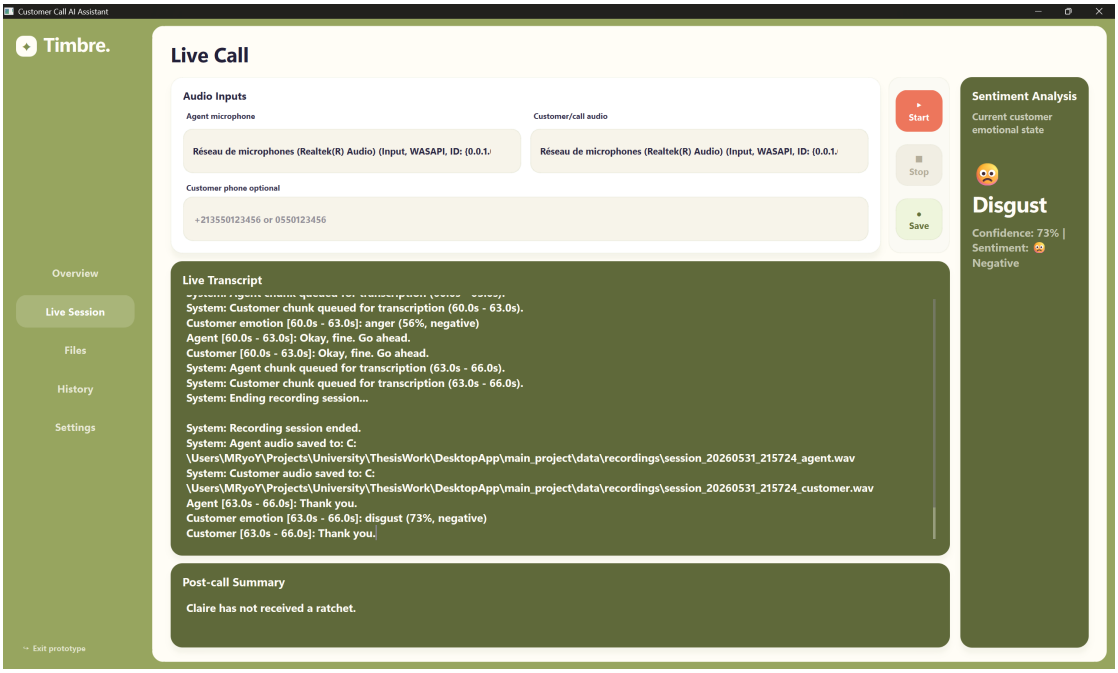


Figure 4.20: Application Live Call Analysis

During live sessions, the recorder also produces short audio chunks. The chunk duration is set to 3 seconds. Each chunk is saved as a WAV file and reported back to the main application through a JSON event. These chunk events are then forwarded to the live transcription and emotion analysis controllers. The main audio capture parameters used by the application are summarized in Table 4.18.

Table 4.18: Audio capture configuration used by the desktop application

Parameter	**Value / Description**
Recording mode	Dual-stream recording
Agent source	Selected microphone input
Customer source	Selected input or WASAPI loopback source
Agent channels	1 channel
Customer channels	Up to 2 channels
Capture sampling rate	48 kHz
Live chunk duration	3 seconds
Audio format	WAV, PCM 16-bit

The general live processing flow is illustrated in Figure 4.21.

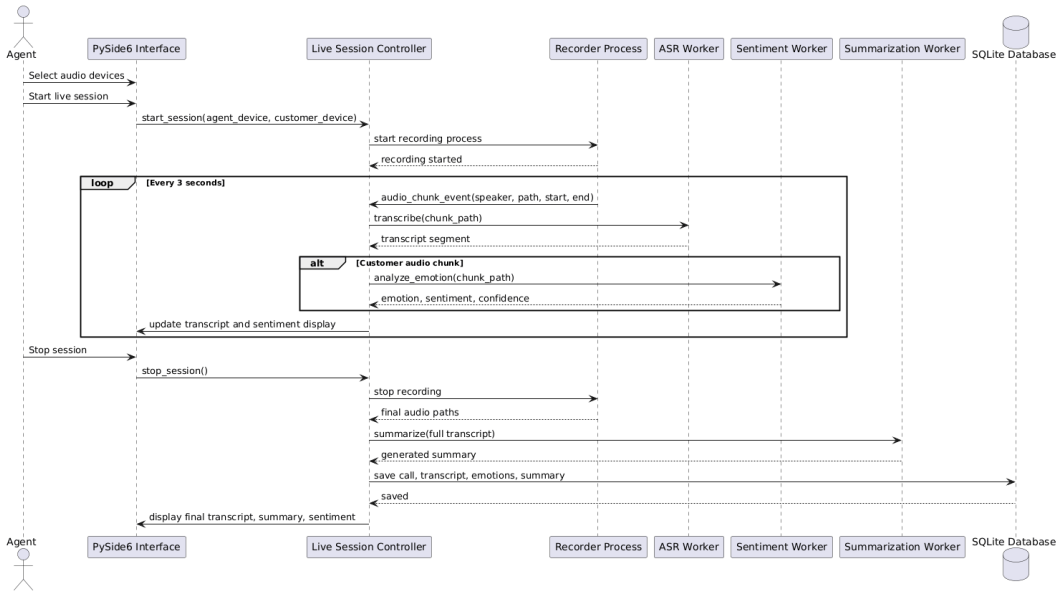


Figure 4.21: Sequence diagram of the live call processing flow

This flow allows the application to process the call progressively while preserving complete audio files for later review.

4.6.3 Model Integration

The three AI modules are integrated as separate model workers, each model is accessed through a dedicated controller that starts a persistent server process when needed and sends inference requests using JSON messages.

Audio chunks or complete audio files are passed to the transcription worker, which loads the model and processor, resamples the audio to 16 kHz, generates token predictions, and decodes them into text. During live calls, both agent and customer chunks can be transcribed and displayed.

The emotion worker receives a customer audio chunk, extracts the required acoustic features, applies the trained model, and returns the predicted emotion, sentiment label, and confidence score.

Summarization is applied after the call. The live transcript segments are first converted into a dialogue format, where each line is associated with a speaker label such as *Agent* or *Customer*. This dialogue is then passed to the summarization worker, which generates a concise summary of the interaction.

Table 4.19 summarizes the inputs and outputs of the three integrated AI modules.

Table 4.19: Integrated AI modules in the desktop application

Module	**Input**	**Output**
Transcription	Agent/customer audio chunks or files	Text transcript
Emotion analysis	Customer audio chunks or files	Emotion, sentiment, confidence
Summarization	Transcript segments	Call summary

The application also includes a file processing mode. In this mode, the user can process existing recordings instead of recording a live call. The file processing page supports two modes: dual-stream recording, where agent and customer files are provided separately, and single mixed recording, where one audio file contains the full call.

4.6.4 Data Storage and Call History

Completed live sessions are saved locally using a SQLite database managed through SQLAlchemy. The database stores the main call information, transcript segments, and emotion results. This allows the application to provide a call history page where previous sessions can be reviewed after processing. The call history page, where previous sessions can be reviewed, is shown in Figure 4.22.

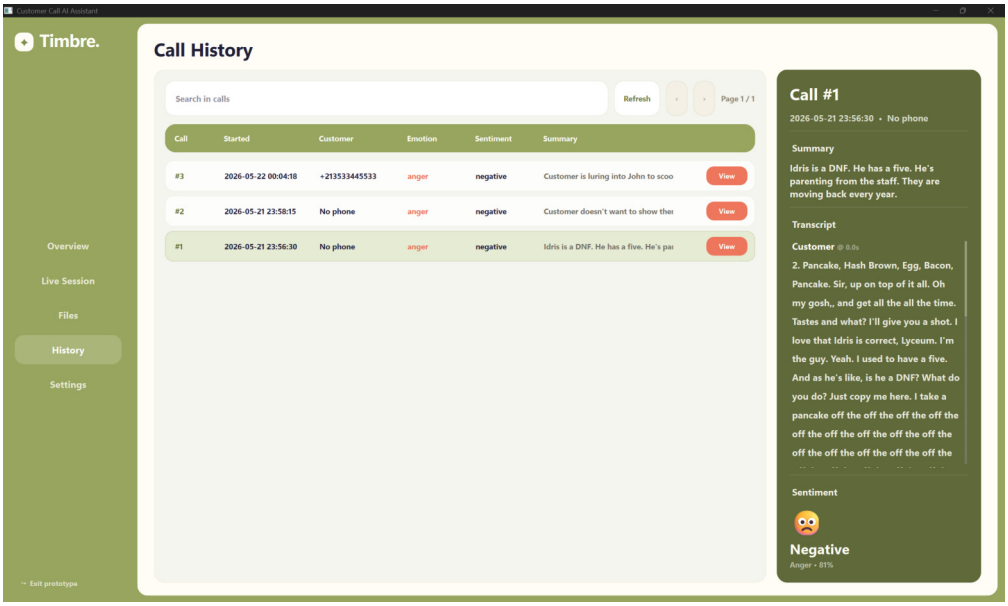


Figure 4.22: Application Call History Page

The main database table is the `calls` table. It stores general information about each session, including the session identifier, start and end times, call method, customer phone number, audio file paths, generated summary, overall emotion, and overall sentiment. The overall emotion and sentiment are selected from the emotion timeline, using the result with the strongest confidence score.

The transcript is stored in a separate `transcript_segments` table. Each segment includes the speaker label, start time, end time, text content, and source. This structure is useful because live transcription produces multiple timestamped segments rather than one single text block.

Emotion predictions are stored in the `emotion_results` table. Each record contains the predicted emotion, sentiment label, confidence score, and the time interval associated with the analyzed audio chunk. This makes it possible to reconstruct an emotion timeline for the customer stream. The database structure used to store call records, transcript segments, and emotion results is shown in Figure 4.23.

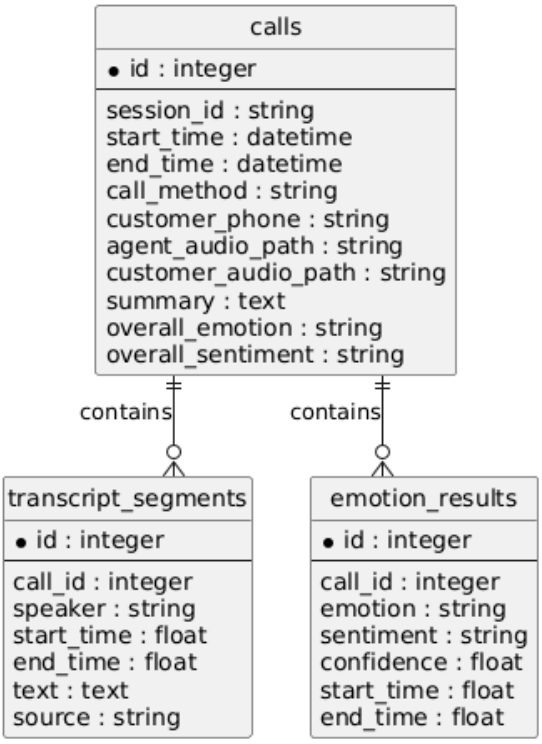


Figure 4.23: Database structure used for call history storage

Table 4.20 summarizes the main information stored for each completed call session.

Table 4.20: Stored data for completed call sessions

Stored Element	**Description**
Session information	Session ID, start time, end time, and call method
Customer information	Customer phone number when provided
Audio paths	Agent and customer recording file paths
Transcript segments	Speaker, timestamp, and transcribed text
Emotion results	Emotion, sentiment, confidence, and time interval
Summary	Generated post-call summary
Overall sentiment	Dominant sentiment selected from the emotion results

4.6.5 User Interface and Final Output Display

The desktop interface is organized into several pages. The dashboard page provides a general status view of the system. The settings page allows the user to configure the audio profile, select the agent microphone, select the customer audio source, test audio levels, and view the configured models and local database storage. The file processing interface is shown in Figure 4.24.

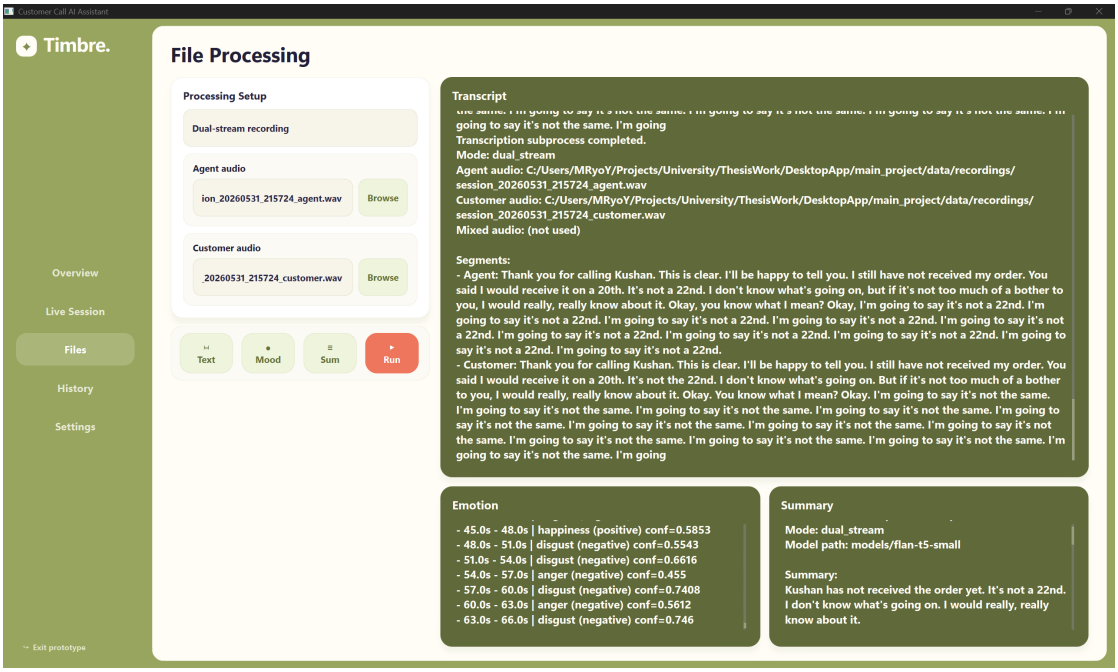


Figure 4.24: Application Call Processing

The live call page is used to run a real-time session. It includes controls for starting and stopping recording, selecting devices, entering call information, and viewing live processing results. During a session, transcript segments are displayed as they are produced, and customer emotion results are updated from the emotion module.

The file processing page allows the user to process existing recordings depending on the user's data and needs.

The history page is used to review saved sessions. It displays previous calls and

allows the user to inspect the generated outputs, and saved audio paths.

The main pages of the desktop application and their functions are summarized in Table 4.21.

Table 4.21: Main pages of the desktop application

Page	**Main Function**
Dashboard	Shows general system and worker status
Live call	Records and processes a customer service call
File processing	Runs the AI pipeline on existing recordings
Call history	Displays saved calls, transcripts, summaries, and emotions
Settings	Configures audio devices, profiles, models, and storage

The final output of the application is composed of three main elements: the call transcript, the generated summary, and the detected customer sentiment. Together, these outputs transform a raw call recording into structured information that can be reviewed more easily by agents or supervisors. The customer and agent displays during a call are shown in Figure 4.25.

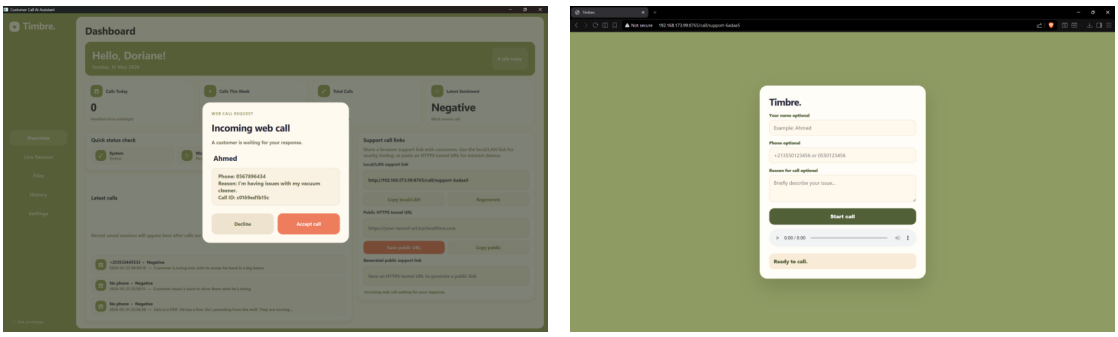


Figure 4.25: Customer and Agent display during a call

4.7 Conclusion

The chapter evaluated the three main modules of the system. For sentiment analysis, the experiments showed that the proposed enhanced MA-CapsNet-based architecture achieved strong performance on mixed emotional speech datasets and maintained good robustness under noisy conditions. For speech transcription, the

Moonshine Tiny Streaming model provided a deployable and lightweight ASR basis, achieving low WER values on Common Voice without additional fine-tuning. For dialogue summarization, the FLAN-T5-based model showed the ability to generate concise summaries from customer service transcripts, while also offering a better balance between performance and deployment cost than larger alternatives.

Finally, the chapter described the integration of these modules into a desktop application. Overall, the implementation demonstrates that the proposed system can transform raw customer service audio into structured outputs composed of a transcript, a summary, and a sentiment label, forming a practical basis for assisted call review and customer interaction analysis.

General Conclusion

In conclusion, this thesis presented a complete system for the automatic analysis of customer service calls using Artificial Intelligence, Machine Learning, and Deep Learning techniques. The developed system combines speech transcription, speech-based sentiment analysis, and dialogue summarization in order to transform raw call audio into structured and useful information. Through this combination, the system aims to support the understanding of both the factual content of a conversation and the emotional state expressed during the interaction.

By automating the analysis of customer service calls, the proposed system helps reduce the limitations of manual review and makes it possible to process interactions more efficiently. Instead of relying only on human agents or supervisors to listen to recordings and extract important information, the system provides a transcript, a concise summary, and a sentiment indication that can be reviewed more easily. This contributes to better organization of call data and supports faster interpretation of customer needs, complaints, and outcomes.

The sentiment analysis module was designed to detect emotional information directly from the audio signal. This choice is important because speech carries vocal cues such as tone, rhythm, intensity, and variation, which may not be fully preserved in text. The transcription module converts the spoken conversation into written form, making the content accessible for further analysis and storage. The summarization module then condenses the transcript into a shorter representation of the interaction, focusing on the main issue, the actions taken, and the final result. Together, these modules provide a more complete view of the customer service exchange.

The integration of these components into a desktop application demonstrates the practical potential of the proposed approach. The application allows users to process live or recorded calls, view the generated transcript, inspect the detected sentiment, read the produced summary, and access previous call records through a local history system. This makes the work not only an experimental study of separate models, but also a functional prototype that reflects a realistic customer service use case.

One of the main advantages of the proposed system is its ability to convert large amounts of unstructured audio data into structured information. Customer service calls often contain valuable insights, but these insights are difficult to exploit when they remain stored only as raw recordings. By extracting transcripts, summaries, and emotional indicators, the system helps make this information more accessible for agents, supervisors, and organizations. This can support follow-up, quality monitoring, training, and the identification of recurring customer problems.

Furthermore, the proposed system can contribute to improving the customer experience. By making important call information easier to review, agents can better understand previous interactions and avoid asking customers to repeat the same details. Supervisors can also gain a clearer view of service quality and customer concerns. When combined with sentiment analysis, call summaries provide not only what happened during the interaction, but also an indication of how the customer felt, which is essential for more responsive and human-centered support.

Despite the promising results, several limitations remain. The sentiment analysis model was developed using emotional speech datasets that may not fully represent the subtle emotions found in real customer service calls. The transcription model may also be affected by noise, accents, overlapping speech, and domain-specific vocabulary. In addition, long transcripts can create challenges for the summarization module, especially when important information is distributed across different parts of the conversation. These limitations show that further refinement is still needed before deployment in a real operational environment.

Future work can focus on improving the system by using real customer service

call recordings, adding stronger speaker diarization, improving noise robustness, and exploring long-context summarization models. The desktop application could also be extended with more advanced dashboards, CRM integration, multilingual support, and real-time alerting when negative sentiment or urgent situations are detected. These improvements would make the system more adaptable to real business environments and more useful for daily customer service operations.

Bibliography

[1] Emma Collins. Omnichannel customer service: Implementation and best practice tips. https://www.helpwire.app/blog/omnichannel-customer-service/, 2024. Published: 2023-06-19; updated: 2024-04-19. Accessed: 2026-05-11.

[2] Collidu. Evolution of customer service. https://www.collidu.com/presentation-evolution-of-customer-service, n.d. Accessed: 2026-05-11.

[3] RingQ. What's a pbx? the different types of pbxs and what to look for. https://www.ringq.com/call-center-faq/pbx/, n.d. Accessed: 2026-04-14.

[4] AiGuideCentral. Ultimate ai glossary: Terms you must know. https://aiguidecentral.com/ultimate-ai-glossary/, n.d. Accessed: 2026-05-11.

[5] Vishakha Joseph, Chandra Prakash Lora, and Narmadha T. Exploring the application of natural language processing for social media sentiment analysis. In *2024 3rd International Conference for Innovation in Technology (INOCON)*, pages 1–6. IEEE, 2024.

[6] Debasish Kalita. A comprehensive overview of sentiment analysis. https://www.analyticsvidhya.com/blog/2022/04/a-comprehensive-overview-of-sentiment-analysis/, 2022. Accessed: 2026-05-11.

[7] Yifan Hu, Xi Huang, Xianbing Wang, Hai Lin, and Rong Zhang. Transformer-based adaptive contrastive learning for multimodal sentiment analysis. *Multimedia Tools and Applications*, 84:1385–1402, 2024.

[8] Hami Ismail. Project: Speech emotion recognition using cnn with crema-d dataset using google colab. `https://hami-asmai.medium.com/project-speech-emotion-recognition-using-cnn-with-crema-d-dataset-using-google-colab-8cb4fdbd5044`, 2023. Accessed: 2026-05-11.

[9] BotPenguin. Speech processing: Benefits and challenges. `https://botpenguin.com/glossary/speech-processing`, n.d. Accessed: 2026-05-11.

[10] Wang Chen, Piji Li, Hou Chan, and Irwin King. Dialogue summarization with supporting utterance flow modeling and fact regularization. *arXiv preprint arXiv:2108.01268*, 2021.

[11] Turbolab Technologies. Types of text summarization: Extractive and abstractive summarization basics. `https://turbolab.in/types-of-text-summarization-extractive-and-abstractive-summarization-basics/`, n.d. Accessed: 2026-05-11.

[12] Kamal Raj Kanakarajan and Malaikannan Sankarasubbu. Saama ai research at semeval-2023 task 7: Exploring the capabilities of flan-t5 for multi-evidence natural language inference in clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 995–1003, 2023.

[13] I Gede Juanamasta, Ni Made Nopita Wati, Erna Hendrawati, Wiwin Wahyuni, Mira Pramudianti, Nugrahini Susantinah Wisnujati, Anindya Prastiwi Setiawati, Susetyorini Susetyorini, Umi Elan, Rusdiyanto Rusdiyanto, et al. The role of customer service through customer relationship management (crm) to increase customer loyalty and good image. *International Journal of Scientific and Technology Research*, 8(10):2004–2007, 2019.

[14] Kenneth Alan Grossberg. *The Origins of Customer Service as Concept and Strategy*. PhD thesis, Waseda University, 2012.

[15] Jukka Majava and Ville Isoherranen. Business model evolution of customer care services. *Journal of Industrial Engineering and Management (JIEM)*, 12(1):1–12, 2019.

[16] Sai Mounika Inavolu. Exploring ai-driven customer service: Evolution, architectures, opportunities, challenges and future directions. *International Journal of Engineering and Advanced Technology*, 13(3):156–163, 2024.

[17] Noah Gans, Ger Koole, and Avishai Mandelbaum. Telephone call centers: Tutorial, review, and research prospects. *Manufacturing & service operations management*, 5(2):79–141, 2003.

[18] Julieo S Thompson. *The Evolution of Technology in Call Centers*. PhD thesis, Nova Southeastern University, 2019.

[19] Yair Meidan, Michael Bohadana, Yisroel Mathov, Yisroel Mirsky, Asaf Shabtai, Dominik Breitenbacher, and Yuval Elovici. A new approach for detecting unknown malicious code in cloud-based call centers. *IEEE Access*, 5:17249–17264, 2017.

[20] Ming-Hui Huang and Roland T Rust. Artificial intelligence in service. *Journal of service research*, 21(2):155–172, 2018.

[21] Erik Cambria, Dipankar Das, Sivaji Bandyopadhyay, and Antonio Feraco. Affective computing and sentiment analysis. In *A practical guide to sentiment analysis*, pages 1–10. Springer, 2017.

[22] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, 4th edition, 2021.

[23] Daniel Jurafsky and James H. Martin. *Speech and Language Processing*. 3rd edition, 2023. Draft version.

[24] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine*, 13(3):55–75, 2018.

[25] Bing Liu. *Sentiment Analysis and Opinion Mining*. Springer Nature, 2022.

[26] Erik Cambria, Björn Schuller, Yunqing Xia, and Catherine Havasi. New avenues in opinion mining and sentiment analysis. *IEEE Intelligent systems*, 28(2):15–21, 2013.

[27] Soujanya Poria, Erik Cambria, Rajiv Bajpai, and Amir Hussain. A review of affective computing: From unimodal analysis to multimodal fusion. *Information Fusion*, 37:98–125, 2017.

[28] James A Russell. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161, 1980.

[29] Maria Pontiki, Dimitrios Galanis, Harris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 19–30, 2016.

[30] Björn W. Schuller. Speech emotion recognition: Two decades in a nutshell, benchmarks, and ongoing trends. *Communications of the ACM*, 61(5):90–99, 2018.

[31] Zhihong Zeng, Maja Pantic, Glenn I. Roisman, and Thomas S. Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):39–58, 2009.

[32] Nikolay Archak, Anindya Ghose, and Panagiotis G Ipeirotis. Deriving the pricing power of product features by mining consumer reviews. *Management science*, 57(8):1485–1509, 2011.

[33] Andranik Tumasjan, Timm Sprenger, Philipp Sandner, and Isabell Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *Proceedings of the international AAAI conference on web and social media*, volume 4, pages 178–185, 2010.

[34] Wu He, Shenghua Zha, and Ling Li. Social media competitive analysis and text mining: A case study in the pizza industry. *International journal of information management*, 33(3):464–472, 2013.

[35] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, 5(4):1093–1113, 2014.

[36] Björn Schuller, Anton Batliner, Stefan Steidl, and Dino Seppi. Recognising realistic emotions and affect in speech: State of the art and lessons learnt from the first challenge. *Speech communication*, 53(9-10):1062–1087, 2011.

[37] Rainer Banse and Klaus R. Scherer. Acoustic profiles in vocal emotion expression. *Journal of Personality and Social Psychology*, 70(3):614–636, 1996.

[38] Klaus R Scherer. Vocal communication of emotion: A review of research paradigms. *Speech communication*, 40(1-2):227–256, 2003.

[39] Johan Sundberg. *The Science of the Singing Voice*. Northern Illinois University Press, 2013.

[40] Björn Schuller, Stefan Steidl, Anton Batliner, Felix Burkhardt, Laurence Devillers, Christian MüLler, and Shrikanth Narayanan. Paralinguistics in speech and language—state-of-the-art and the challenge. *Computer Speech & Language*, 27(1):4–39, 2013.

[41] Nicholas Cummins, Stefan Scherer, Jarek Krajewski, Sebastian Schnieder, Julien Epps, and Thomas F. Quatieri. A review of depression and suicide risk assessment using speech analysis. *Speech Communication*, 71:10–49, 2015.

[42] Sidney D'Mello and Jacqueline Kory. Speech emotion recognition in educational contexts: A review. *Educational Psychologist*, 53(2):83–107, 2018.

[43] Moataz El Ayadi, Mohamed S. Kamel, and Fakhri Karray. Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*, 44(3):572–587, 2011.

[44] Peter C. Verhoef, Katherine N. Lemon, A. Parasuraman, Anne Roggeveen, Michael Tsiros, and Leonard A. Schlesinger. Customer experience creation: Determinants, dynamics and management strategies. *Journal of Retailing*, 85(1):31–41, 2009.

[45] Dagmar M Schuller and Björn W Schuller. A review on five recent and near-future developments in computational processing of emotion in the human voice. *Emotion Review*, 13(1):44–50, 2021.

[46] Jinyu Li, Li Deng, Yifan Gong, and Reinhold Haeb-Umbach. An overview of noise-robust automatic speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(4):745–777, 2014.

[47] Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, and Rada Mihalcea. Emotion recognition in conversation: Research challenges, datasets, and recent advances. *IEEE Access*, 7:100943–100953, 2019.

[48] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021.

[49] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Yao, Jinyu Zhang, Furu Wei, et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518, 2022.

[50] Christoph Schuhmann, Robert Kaczmarczyk, Gollam Rabby, Felix Friedrich, Maurice Kraus, Kourosh Nadi, Huu Nguyen, Kristian Kersting, and Sören Auer. Emonet-voice: A fine-grained, expert-verified benchmark for speech emotion detection. *arXiv preprint arXiv:2506.09827*, 2025.

[51] Yonghong Fan, Heming Huang, Huiyun Zhang, et al. Emo-stcapsnet: A spatio-temporal modeling approach with enhanced capsule network for speech emotion recognition. *Knowledge-Based Systems*, 2026.

[52] Lawrence Rabiner and Biing-Hwang Juang. *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[53] Dong Yu and Li Deng. *Automatic Speech Recognition: A Deep Learning Approach*. Springer, 2016.

[54] Xavier Anguera, Simon Bozonnet, Nicholas Evans, Corinne Fredouille, Gerald Friedland, and Oriol Vinyals. Speaker diarization: A review of recent research. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(2):356–370, 2012.

[55] Mishaim Malik, Muhammad Kamran Malik, Khawar Mehmood, and Imran Makhdoom. Automatic speech recognition: a survey. *Multimedia Tools and Applications*, 80(6):9411–9457, 2021.

[56] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR, 2023.

[57] Monica Sekoyan, Nithin Rao Koluguri, Nune Tadevosyan, Piotr Zelasko, Travis Bartley, Nick Karpov, Jagadeesh Balam, and Boris Ginsburg. Canary-1b-v2 & parakeet-tdt-0.6b-v3: Efficient and high-performance models for multilingual asr and ast. *arXiv preprint arXiv:2509.14128*, 2025.

[58] Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, et al. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*, 2024.

[59] Loïc Barrault et al. Seamlessm4t—massively multilingual and multimodal machine translation. *arXiv preprint arXiv:2308.11596*, 2023.

[60] Ani Nenkova and Kathleen McKeown. Automatic summarization. *Foundations and Trends® in Information Retrieval*, 5(2–3):103–233, 2011.

[61] Yusen Zhang, Hai Zhao, Chengqing Li, Lidong Liu, and Min Zhang. A survey of dialogue summarization. *arXiv preprint arXiv:2108.13629*, 2021.

[62] Gabriel Murray, Steve Renals, and Jean Carletta. Extractive summarization of meeting recordings. In *Proceedings of Interspeech*, 2005.

[63] Gabriel Murray, Jean Carletta, and Steve Renals. Extractive summarization of meeting recordings. *ACM Transactions on Speech and Language Processing*, 6(3):1–21, 2009.

[64] Dingding Wang and Tao Li. A system for real-time meeting summarization using speaker and content information. In *Proceedings of the ACL-HLT 2011 System Demonstrations*, pages 91–96, 2011.

[65] Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of NAACL-HLT*, pages 615–621, 2018.

[66] Filippo Galgani, Paul Compton, and Achim Hoffmann. Automatic keyphrase extraction from legal documents. *Artificial Intelligence and Law*, 20(3):239–273, 2012.

[67] Dragomir R. Radev, Hongyan Jing, Malgorzata Styś, and Daniel Tam. Centroid-based summarization of multiple documents: Sentence extraction, utility-based evaluation, and user studies. *Information Processing & Management*, 40(6):919–938, 2004.

[68] Stergos D. Afantenos, Vangelis Karkaletsis, and Panagiotis Stamatopoulos. Summarization from medical documents: A survey. *Artificial Intelligence in Medicine*, 33(2):157–177, 2005.

[69] Udo Hahn and Inderjeet Mani. The challenges of automatic summarization for financial documents. In *Proceedings of the ACL Workshop on Automatic Summarization*, 2000.

[70] Hans Peter Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165, 1958.

[71] H. P. Edmundson. New methods in automatic extracting. *Journal of the ACM*, 16(2):264–285, 1969.

[72] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411, 2004.

[73] Günes Erkan and Dragomir R Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479, 2004.

[74] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1073–1083, 2017.

[75] Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, 2015.

[76] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[77] Tetsuya Nasukawa and Jeonghee Yi. Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture*, pages 70–77, 2003.

[78] Angela Fan, David Grangier, and Michael Auli. Controllable abstractive summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 45–54, 2018.

[79] Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, 2020.

[80] Erlend Antonsen et al. Customer service analytics: A survey of recent developments. *Service Science*, 2018.

[81] Meta AI. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation. `https://ai.meta.com/blog/llama-4-multimodal-intelligence/`, 2025. Accessed: 2026-06-01.

[82] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[83] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 7871–7880, 2020.

[84] Aditya Gotmare et al. An adaptive fusion-based data augmentation method for abstractive dialogue summarization. *PeerJ Computer Science*, 2026.

[85] OpenAI, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

[86] Felix Burkhardt, Astrid Paeschke, Miriam Rolfes, Walter F. Sendlmeier, and Benjamin Weiss. A database of german emotional speech. In *Proceedings of Interspeech*, pages 1517–1520, 2005.

[87] S. Haq and P. J. B. Jackson. Audio-visual emotion recognition using the surrey audio-visual expressed emotion (savee) database. In *Proceedings of the International Conference on Auditory-Visual Speech Processing*, pages 53–58, 2009.

[88] Steven R. Livingstone and Frank A. Russo. The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PLOS ONE*, 13(5):e0196391, 2018.

[89] Houwei Cao, David G. Cooper, Michael K. Keutmann, Ruben C. Gur, Ani Nenkova, and Ragini Verma. Crema-d: Crowd-sourced emotional multimodal

actors dataset. *IEEE Transactions on Affective Computing*, 5(4):377–390, 2014.

[90] Heming Huang, Yonghong Fan, Huiyun Zhang, et al. Ma-capsnet-da: Speech emotion recognition based on ma-capsnet using data augmentation. *Expert Systems with Applications*, 234:121023, 2023.

[91] Andrew Varga and Herman J. M. Steeneken. Assessment for automatic speech recognition: Ii. noisex-92: A database and an experiment to study the effect of additive noise on speech recognition systems. *Speech Communication*, 12(3):247–251, 1993.

[92] Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber. Common voice: A massively-multilingual speech corpus. *arXiv preprint arXiv:1912.06670*, 2019.

[93] Jean Carletta. Unleashing the killer corpus: Experiences in creating the multi-everything ami meeting corpus. *Language Resources and Evaluation*, 41(2):181–190, 2007.

[94] Nat Jeffries, Evan King, Manjunath Kudlur, Guy Nicholson, James Wang, and Pete Warden. Moonshine: Speech recognition for live transcription and voice commands. *arXiv preprint arXiv:2410.15608*, 2024.

[95] Useful Sensors. Usefulsensors/moonshine-streaming-tiny. `https://huggingface.co/UsefulSensors/moonshine-streaming-tiny`, 2026. Accessed: 2026-05-09.

[96] Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, 2019.

[97] Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. Dialogsum: A real-life scenario dialogue summarization dataset. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5062–5074, 2021.

[98] AIxBlock. 92k real-world call center scripts english. https://huggingface.co/datasets/AIxBlock/92k-real-world-call-center-scripts-english, 2025. Accessed: 2026-05-09.

[99] ThoughtVector. Customer support on twitter. https://www.kaggle.com/datasets/thoughtvector/customer-support-on-twitter, 2017. Accessed: 2026-05-09.

[100] Philipp Schmid. Bart large cnn samsum. https://huggingface.co/philschmid/bart-large-cnn-samsum, 2021. Accessed: 2026-05-09.

[101] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة محمد خيضر بسكرة

كلية العلوم الدقيقة وعلوم الطبيعة والحياة

عنوان المشروع

تطبيق مكتبي ذكي لتحليل مكالمات خدمة العملاء باستعمال الذكاء الاصطناعي

مشروع لنيل شهادة مؤسسة ناشئة في إطار القرار الوزاري **1275**

صورة العلامة التجارية



الاسم التجاري

Timbre

السنة الجامعية: **2025/2026**

بطاقة معلومات:
حول فريق الإشراف وفريق العمل

1- فريق الإشراف:

فريق الإشراف	
التخصص: إعلام آلي	المشرف الرئيسي: نور الهدى زرارقة

2- فريق العمل:

الكلية	**التخصص**	**فريق المشروع**
كلية العلوم الدقيقة وعلوم الطبيعة والحياة	AI	الطالب: الوافي إيلاف دريان

المحتويات

مقدمة 5

1 المحور الأول: تقديم المشروع 6
1 فكرة المشروع والحل المقترح 6
2 القيم المقترحة 7
3 فريق العمل 8
4 أهداف المشروع 8
5 الجدول الزمني لتحقيق المشروع 9

2 المحور الثاني: الجوانب الابتكارية 11
1 طبيعة الابتكارات 11
2 مجالات الابتكار 11

3 المحور الثالث: التحليل الاستراتيجي للسوق 13
1 عرض القطاع السوقي 13
2 تحليل المتغيرات الكلية PESTEL 14
3 تحليل القوى التنافسية PORTER 15
4 تحليل SWOT 16
5 المزيج التسويقي 17
6 الاستراتيجيات التسويقية 18

4 المحور الرابع: خطة الإنتاج والتنظيم 20
1 خطة الإنتاج 20
2 خطة التنظيم 22
3 الموردون والشركاء الرئيسيون 23
4 الشراكات الرئيسية 24
5 المهام الرئيسية 24
6 الشكل التنظيمي المقترح 25

5 المحور الخامس: الخطة المالية 27
1 التكاليف والأعباء 27
2 رقم الأعمال 29
3 خطة الخزينة 31
4 جدول حسابات النتائج المتوقع 31
5 نقطة التعادل 32
6 خلاصة الخطة المالية 33

6 المحور السادس: النموذج الأولي التجريبي 34
1 تقديم النموذج الأولي 34
2 وصف التطبيق 35
3 مكونات النموذج الأولي 37
4 طريقة عمل النموذج الأولي 38
5 الواجهات الأساسية للتطبيق 40
6 نتائج النموذج الأولي 41
7 حدود النموذج الأولي 42
8 آفاق تطوير النموذج الأولي 43
9 خلاصة المحور 43

مقدمة

أصبحت الرقمنة والذكاء الاصطناعي من أهم الوسائل التي تساعد المؤسسات على تحسين خدماتها واستغلال بياناتها بشكل أفضل. ومن بين هذه البيانات نجد مكالمات خدمة العملاء، التي تحتوي على معلومات مهمة حول احتياجات الزبائن، شكاويهم، ومستوى رضاهم، غير أنها غالبا ما تبقى محفوظة كتسجيلات صوتية يصعب تحليلها أو البحث داخلها يدويا.

من هذا المنطلق، جاء مشروعنا المتمثل في تطبيق مكتبي ذكي لتحليل مكالمات خدمة العملاء باستعمال تقنيات الذكاء الاصطناعي. يهدف التطبيق إلى تحويل المكالمات الصوتية إلى نصوص مكتوبة، تقدير الحالة الشعورية العامة للعميل، ثم توليد ملخص مختصر للمكالمة وحفظ النتائج في قاعدة بيانات يمكن الرجوع إليها لاحقا.

يساعد هذا الحل المؤسسات على تقليل الوقت والجهد اللازمين لمراجعة التسجيلات، تحسين متابعة جودة خدمة العملاء، وتسهيل الوصول إلى المعلومات المهمة داخل المكالمات. كما يتميز المشروع بإمكانية المعالجة المحلية داخل أجهزة المؤسسة، مما يساهم في حماية خصوصية البيانات وتقليل تكاليف التخزين والبنية التحتية.

تندرج هذه الدراسة ضمن إطار إعداد مشروع مؤسسة ناشئة وفق القرار الوزاري 1275، وتهدف إلى عرض فكرة المشروع، جوانبه الابتكارية، تحليله السوقي، خطته التنظيمية والمالية، إضافة إلى تقديم النموذج الأولي التجريبي للتطبيق.

المحور 1

المحور الأول: تقديم المشروع

1 فكرة المشروع والحل المقترح

- مشروعنا هو حل رقمي ذكي يعتمد على خدمة العملاء، يهدف إلى تحويل المكالمات الهاتفية من تسجيلات صوتية صعبة الاستغلال إلى نصوص وملخصات ومؤشرات تساعد على فهم حالة العميل وتحسين جودة الخدمة.

- بدأت فكرة المشروع من المشاكل التي تواجهها المؤسسات التي تعتمد على المكالمات الهاتفية للتواصل مع الزبائن، حيث يتم يوميا تسجيل عدد كبير من المكالمات، لكن أغلبها لا يتم تحليله أو استغلاله بطريقة فعالة. وغالبا ما تبقى هذه المكالمات محفوظة على شكل ملفات صوتية يصعب البحث فيها أو استخراج المعلومات المهمة منها.

- تعاني المؤسسات من ضياع معلومات مهمة داخل المكالمات غير المعالجة، مثل شكاوى الزبائن، أسباب عدم الرضا، المشاكل المتكررة، وطريقة تعامل أعوان خدمة العملاء مع هذه الحالات. كما أن مراجعة المكالمات يدويا تتطلب وقتا وجهدا وتكلفة معتبرة، خاصة بالنسبة للمؤسسات التي تستقبل عددا كبيرا من الاتصالات يوميا.

- من جهة أخرى، فإن تخزين التسجيلات الصوتية لمدة طويلة يتطلب مساحة تخزين كبيرة، بينما يمكن تحويل محتوى المكالمات إلى نصوص وملخصات صغيرة الحجم وسهلة البحث، مما يقلل من تكلفة التخزين ويسهل استغلال البيانات لاحقا.

- لذلك جاء مشروعنا كحل رقمي يتمثل في تطبيق مكتبي ذكي يسمح للمؤسسات بـ:

1. تفريغ مكالمات خدمة العملاء وتحويلها إلى نصوص مكتوبة.
2. تقدير الحالة الشعورية العامة للعميل أثناء المكالمة، سواء كانت إيجابية، سلبية، أو محايدة.
3. تلخيص المكالمة بعد انتهائها، مع إبراز أهم النقاط مثل مشكلة العميل، الإجراءات المقترحة، والنتيجة النهائية.
4. حفظ نتائج المكالمات في قاعدة بيانات محلية أو قاعدة بيانات خاصة بالمؤسسة.
5. البحث في المكالمات السابقة حسب الكلمات المفتاحية أو محتوى المكالمة أو الحالة الشعورية العامة.

6. معالجة مكالمات سابقة مسجلة، وليس فقط المكالمات المباشرة.

- يعتمد المشروع على فكرة المعالجة المحلية داخل أجهزة المؤسسة، حيث يمكن لكل حاسوب خاص بعون خدمة العملاء أن يعالج المكالمات الخاصة به، ثم يتم جمع النتائج لاحقا داخل قاعدة بيانات موحدة. وهذا يسمح بتقليل تكاليف البنية التحتية، لأن المؤسسة تستفيد من الحواسيب الموجودة لديها بدلا من شراء خوادم قوية ومكلفة لمعالجة كل المكالمات في مكان واحد.
- كما أن إبقاء المعالجة والبيانات داخل المؤسسة يساهم في حماية خصوصية الزبائن، لأن المكالمات لا تحتاج إلى إرسالها إلى خدمات خارجية من أجل تحليلها. وهذا يعتبر نقطة مهمة بالنسبة للمؤسسات التي تتعامل مع معلومات حساسة أو بيانات خاصة بالعملاء.
- يمكن تقديم المنتج المقترح في شكلين أساسيين:

1. بالنسبة للمؤسسات الكبيرة: يتم تقديم نسخة مخصصة من التطبيق حسب احتياجات المؤسسة ونظام الاتصال الخاص بها، من خلال عقد تطوير أو تخصيص، مثل شركات الاتصالات، البنوك، شركات التأمين، ومراكز الاتصال.
2. بالنسبة للمؤسسات الصغيرة والمصغرة: يتم تقديم نسخة مكتبية جاهزة يمكن استعمالها برخص قصيرة المدى أو باشتراك محدود، مع إمكانية توفير نسخة مجانية محدودة الخصائص للتجربة.

2 القيم المقترحة

تتوافق القيم المقترحة في هذا المشروع مع العناصر التالية:

1. الحداثة: يقدم المشروع حلا حديثا يعتمد على الذكاء الاصطناعي لتحليل مكالمات خدمة العملاء، من خلال الجمع بين التفريغ الآلي، تقدير الحالة الشعورية، والتلخيص الآلي في تطبيق واحد.
2. سرعة التنفيذ: يسمح التطبيق بمعالجة المكالمات بسرعة، سواء أثناء المكالمة أو بعد انتهائها، مما يساعد على الحصول على النص والملخص والمؤشر الشعوري في وقت قصير.
3. الأداء: يساعد التطبيق أعوان خدمة العملاء والمشرفين على الوصول إلى المعلومات المهمة دون الحاجة إلى الاستماع الكامل لكل التسجيلات.
4. خفض التكلفة: يساهم المشروع في تقليل تكاليف مراجعة المكالمات يدويا، وتقليل الحاجة إلى تخزين كميات كبيرة من الملفات الصوتية، بالاعتماد على النصوص والملخصات.
5. استغلال الأجهزة الموجودة: يمكن تشغيل التطبيق على حواسيب أعوان خدمة العملاء الموجودة أصلا داخل المؤسسة، مما يسمح بمعالجة المكالمات من المصدر دون الحاجة إلى بنية تحتية مكلفة في المرحلة الأولى.
6. حماية البيانات: تتم المعالجة بشكل محلي داخل المؤسسة، مما يقلل من خطر تسريب بيانات الزبائن ويحافظ على سرية المكالمات والمعلومات الحساسة.

7. سهولة الوصول إلى المعلومات: يوفر التطبيق قاعدة بيانات قابلة للبحث، مما يسمح بالوصول إلى المكالمات السابقة ومعرفة المشاكل المتكررة بسرعة.
8. تحسين جودة الخدمة: يساعد التطبيق على إعطاء صورة أوضح حول حالة العميل أثناء المكالمة، وتحديد المكالمات التي تحتاج إلى متابعة أو تحسين في طريقة التعامل.
9. المرونة: يمكن تكييف التطبيق حسب حجم المؤسسة، سواء من خلال نسخة بسيطة للمؤسسات الصغيرة أو نسخة مخصصة للمؤسسات الكبيرة.
10. سهولة الاستخدام: يتم تطوير التطبيق كبرنامج مكتبي بواجهة واضحة تسمح ببدء المعالجة، عرض النص، مشاهدة الحالة الشعورية، توليد الملخص، وحفظ النتائج بطريقة بسيطة.

من خلال هذه القيم، يهدف المشروع إلى تقديم حل رقمي عملي يساعد المؤسسات على تحويل مكالمات خدمة العملاء من تسجيلات صوتية صعبة الاستغلال إلى بيانات منظمة وقابلة للبحث والتحليل.

3 فريق العمل

يتكون فريق العمل في المرحلة الحالية من صاحب المشروع، باعتباره المطور الرئيسي والمسؤول عن إنجاز النموذج الأولي للتطبيق، ودراسة الجوانب التقنية والتجارية للمشروع.

الاسم واللقب	المؤهلات والمهارات	المهام
الوافي إيلاف دريان	-ليسانس إعلام آلي، تخصص نظم معلوماتية. -طالب ماستر 2 في الذكاء الاصطناعي. -معارف إضافية في تطوير البرمجيات وتطوير تطبيقات الويب.	-تصميم وبرمجة التطبيق. -تطوير واجهة الاستخدام. -دمج نماذج الذكاء الاصطناعي. -إعداد قاعدة البيانات-. -اختبار النموذج الأولي. -إعداد الدراسة التجارية للمشروع.

ومع تطور المشروع مستقبلا، يمكن توسيع الفريق بإضافة مختص في التسويق، ومختص في الدعم التقني، ومختص في أمن المعلومات، خاصة عند التوجه نحو المؤسسات الكبيرة أو عند توقيع عقود مخصصة.

4 أهداف المشروع

تشمل الأهداف العامة للمشروع ما يلي:

- المساهمة في رقمنة خدمات المؤسسات وتحسين طريقة استغلال البيانات الناتجة عن تواصلها مع الزبائن.

- مساعدة المؤسسات على تحسين مراقبة جودة خدمة العملاء من خلال توفير رؤية أوضح حول مضمون المكالمات وطبيعة تفاعل الزبائن.
- دعم عملية اتخاذ القرار داخل المؤسسة من خلال تحويل تفاعلات العملاء إلى مؤشرات ومعلومات قابلة للاستعمال.
- تحسين تجربة الزبون من خلال تمكين المؤسسة من فهم احتياجاته ومشاكله المتكررة بشكل أفضل.
- دعم المؤسسات في بناء أرشيف رقمي منظم لمكالمات العملاء، يمكن الرجوع إليه عند الحاجة ومتابعة المشاكل المتكررة من خلاله.
- تشجيع استعمال حلول الذكاء الاصطناعي المحلية في المؤسسات الجزائرية، خاصة في المجالات المرتبطة بخدمة العملاء والتحول الرقمي.
- تقديم حل عملي يساعد المؤسسات على رفع كفاءة العمل وتقليل الوقت الضائع في المهام المتكررة.
- تطوير مشروع قابل للنمو من نموذج أولي بسيط إلى منتج تجاري يمكن تسويقه وتكييفه مع احتياجات قطاعات مختلفة.
- خلق فرصة لمؤسسة ناشئة محلية تقدم خدمة رقمية ذات قيمة مضافة في مجال تحليل تواصل الزبائن.
- بناء نموذج اقتصادي مستدام يعتمد على الرخص، الاشتراكات، العقود المخصصة، وخدمات الدعم والصيانة.

5 الجدول الزمني لتحقيق المشروع

يوضح الجدول التالي المراحل الأساسية المقترحة لإنجاز المشروع وتطويره:

المرحلة	النشاط	المدة الزمنية
الدراسة والتخطيط	دراسة مشاكل المؤسسات في معالجة مكالمات خدمة العملاء، وتحديد الفئة المستهدفة.	أسبوعان
تحديد المتطلبات	تحديد الوظائف الأساسية للتطبيق، مثل التفريغ الآلي، تقدير الحالة الشعورية، التلخيص، التخزين، والبحث.	أسبوعان
التصميم	تصميم واجهة التطبيق وقاعدة البيانات وطريقة عرض نتائج المكالمات.	3 أسابيع
البرمجة والتطوير	برمجة التطبيق المكتبي ودمج نماذج الذكاء الاصطناعي وربطها بقاعدة البيانات.	من شهرين إلى ثلاثة أشهر
الاختبار والتجربة	تجربة التطبيق على مكالمات مباشرة ومكالمات مسجلة سابقا، وتصحيح الأخطاء.	من 3 إلى 4 أسابيع
تحسين النموذج الأولي	تحسين الواجهة والأداء وطريقة عرض النتائج بناء على التجربة.	شهر واحد
الإطلاق الأولي	تقديم نسخة تجريبية محدودة للمؤسسات الصغيرة أو للمستخدمين الأوائل.	أسبوعان
التسويق	إعداد عرض تجاري للمشروع، والتواصل مع المؤسسات التي تعتمد على خدمة العملاء.	شهر واحد
المتابعة والتطوير	جمع الملاحظات، تحسين التطبيق، وإضافة خصائص جديدة حسب احتياجات المستخدمين.	مستمر

المحور 2

المحور الثاني: الجوانب الابتكارية

1 طبيعة الابتكارات

يمكن تحديد طبيعة الابتكارات المعتمدة في المشروع من خلال النقاط التالية:

- ابتكار في استغلال مكالمات خدمة العملاء: يعتمد المشروع على تحويل المكالمات من تسجيلات صوتية صعبة الاستغلال إلى بيانات منظمة، مثل النصوص المكتوبة، الملخصات، والمؤشرات العامة حول حالة العميل، مما يجعلها قابلة للبحث والتحليل.
- ابتكار تكنولوجي: يوظف المشروع تقنيات الذكاء الاصطناعي في تطبيق واحد يجمع بين التفريغ الآلي للمكالمات، تقدير الحالة الشعورية العامة للعميل، والتلخيص الآلي لمحتوى المكالمة.
- ابتكار في طريقة المعالجة: يعتمد التطبيق على مبدأ المعالجة المحلية داخل أجهزة المؤسسة، مما يقلل الحاجة إلى إرسال البيانات إلى خدمات خارجية، ويسمح بالاستفادة من الحواسيب الموجودة مسبقا.
- ابتكار في تخفيض التكاليف: يساهم المشروع في تقليل تكلفة مراجعة المكالمات يدويا، وتقليل مساحة التخزين المطلوبة، لأن النصوص والملخصات تحتاج إلى مساحة أقل مقارنة بالتسجيلات الصوتية.
- ابتكار في دعم اتخاذ القرار: يساعد التطبيق المؤسسة على استخراج معلومات عملية من المكالمات، مثل تحديد المكالمات ذات الطابع السلبي، معرفة المشاكل المتكررة، ومتابعة جودة تعامل أعوان خدمة العملاء.
- ابتكار تدريجي قابل للتطوير: يمكن تطوير المشروع مستقبلا بإضافة لوحة تحكم للمشرفين، إحصائيات حول حالة العملاء، تصنيف المكالمات حسب الموضوع، أو ربط التطبيق بأنظمة إدارة علاقات العملاء.

بناء على ذلك، يمكن اعتبار المشروع ابتكارا تكنولوجيا وخدماتيا في نفس الوقت، لأنه يستعمل تقنيات الذكاء الاصطناعي ويقترح طريقة عملية لتحسين خدمة العملاء وتقليل تكاليف معالجة المكالمات.

2 مجالات الابتكار

تظهر مجالات الابتكار في هذا المشروع من خلال عدة جوانب أساسية، أهمها:

- تحليل مكالمات خدمة العملاء: يقدم التطبيق طريقة جديدة لاستغلال المكالمات، من خلال التعامل معها كمصدر للمعلومات وليس كتسجيلات صوتية فقط.

- تقدير الحالة الشعورية للعميل: يوفر التطبيق مؤشرا عاما حول حالة العميل أثناء المكالمة، مثل إيجابية، سلبية، أو محايدة، مما يساعد على تحديد المكالمات التي تحتاج إلى متابعة.
- التفريغ الآلي للمكالمات: يحول التطبيق الكلام المنطوق إلى نص مكتوب، مما يسهل حفظ محتوى المكالمة والرجوع إليه لاحقا.
- التلخيص الآلي: يقوم التطبيق بإنشاء ملخص مختصر لأهم نقاط المكالمة، مثل سبب الاتصال، المشكلة المطروحة، والإجراء المتخذ.
- الأرشفة والبحث: يسمح المشروع بحفظ نتائج المكالمات في قاعدة بيانات قابلة للبحث حسب الكلمات المفتاحية أو محتوى المكالمة أو الحالة الشعورية.
- حماية خصوصية البيانات: تتم المعالجة محليا داخل المؤسسة، مما يقلل الحاجة إلى إرسال بيانات العملاء إلى منصات خارجية.
- الاستفادة من الأجهزة المتوفرة: يمكن تشغيل التطبيق على حواسيب أعوان خدمة العملاء، مما يوزع عملية المعالجة ويخفض تكلفة الانطلاق.
- معالجة المكالمات السابقة: يسمح التطبيق بمعالجة تسجيلات قديمة، مما يساعد المؤسسة على الاستفادة من أرشيفها الصوتي.
- قابلية التخصيص: يمكن تكييف التطبيق حسب طبيعة المؤسسة واحتياجاتها، سواء من حيث العرض، التخزين، أو البحث.
- نموذج تجاري مرن: يمكن تقديم نسخة مخصصة للمؤسسات الكبيرة، ونسخة أبسط للمؤسسات الصغيرة عن طريق رخص أو اشتراكات محدودة.

من خلال هذه المجالات، يسعى المشروع إلى تقديم حل عملي يحول مكالمات خدمة العملاء من عبء تخزيني وتشغيلي إلى مصدر منظم للمعلومات، يساعد المؤسسة على تحسين جودة خدماتها وفهم زبائنها بشكل أفضل.

المحور 3

المحور الثالث: التحليل الاستراتيجي للسوق

1 عرض القطاع السوقي

ينتمي مشروعنا إلى قطاع الخدمات الرقمية الموجهة للمؤسسات، وبشكل خاص إلى مجال حلول الذكاء الاصطناعي الخاصة بتحسين خدمة العملاء وتحليل المكالمات الهاتفية. ويعرف هذا المجال اهتماما متزايدا، بسبب اعتماد المؤسسات على المكالمات كوسيلة مهمة للتواصل مع الزبائن في الاستفسارات، الدعم، الشكاوى، وخدمات ما بعد البيع.

يستهدف المشروع في نسخته الأولية الأسواق الناطقة باللغة الإنجليزية، وذلك لأن النماذج والبيانات المتوفرة حاليا في مجال التفريغ الآلي، تحليل المشاعر، والتلخيص تكون غالبا أكثر تطورا ودقة باللغة الإنجليزية. ويسمح هذا التوجه باختبار النموذج الأولي في بيئة لغوية أكثر استقرارا قبل الانتقال إلى أسواق أخرى.

أما في مرحلة لاحقة، فيهدف المشروع إلى التوجه نحو السوق الجزائرية من خلال تطوير دعمه للغات المستعملة محليا، خاصة العربية، الفرنسية، والدارجة الجزائرية. ويعتبر هذا التطوير مهما لأن السوق الجزائرية تحتاج إلى حلول محلية تراعي خصوصية اللغة وطريقة تواصل الزبائن مع المؤسسات.

يمكن تقسيم السوق المستهدف إلى الفئات التالية:

- المؤسسات الكبيرة: مثل شركات الاتصالات، البنوك، شركات التأمين، مؤسسات النقل، ومؤسسات الخدمات التي تملك مراكز اتصال أو أقساما خاصة بخدمة العملاء.
- المؤسسات المتوسطة: مثل شركات التجارة الإلكترونية، شركات التوصيل، العيادات الخاصة، المؤسسات التعليمية الخاصة، ومراكز الدعم التقني.
- المؤسسات الصغيرة والمصغرة: وهي المؤسسات التي لا تملك مراكز اتصال كبيرة، لكنها تتعامل مع الزبائن عبر الهاتف بشكل منتظم وتحتاج إلى تنظيم مكالماتها.
- أصحاب الأنشطة الحرة: مثل مكاتب الخدمات، الوكالات، أو الفرق الصغيرة التي تحتاج إلى متابعة تواصلها مع الزبائن دون الاستثمار في أنظمة مكلفة.

وتكمن فرصة المشروع في أن عددا كبيرا من المؤسسات يقوم بتسجيل المكالمات أو استقبالها يوميا، لكن دون وجود أداة بسيطة تساعد على تحويل هذه التسجيلات إلى بيانات قابلة للبحث والتحليل. ومن هنا يقترح المشروع تطبيقا مكتبيا يمكن استعماله داخل المؤسسة، مع الحفاظ على خصوصية البيانات وتقليل تكلفة المعالجة والتخزين.

2 تحليل المتغيرات الكلية PESTEL

يستخدم تحليل PESTEL لدراسة العوامل الخارجية التي يمكن أن تؤثر على المشروع، سواء بطريقة إيجابية أو سلبية. ويمكن تلخيص هذه العوامل كما يلي:

العامل	تأثيره على المشروع
العوامل السياسية	توجه الدولة نحو رقمنة القطاعات وتشجيع المؤسسات الناشئة يمثل عاملا إيجابيا للمشروع، خاصة أنه يدخل ضمن مجال التحول الرقمي واستعمال الذكاء الاصطناعي في تحسين الخدمات. كما أن دعم المشاريع المبتكرة في إطار القرار الوزاري 1275 يمكن أن يساعد على تطوير المشروع في مراحله الأولى.
العوامل الاقتصادية	يمكن أن تستفيد المؤسسات من هذا النوع من الحلول لأنه يساعد على تقليل تكاليف مراجعة المكالمات يدويا، وتقليل تكاليف التخزين والمعالجة. لكن القدرة المالية لبعض المؤسسات الصغيرة قد تؤثر على سرعة تبنيها للتطبيق، لذلك يجب توفير نسخة محدودة التكلفة أو رخص قصيرة المدى.
العوامل الاجتماعية	يزداد اعتماد الزبائن على الهاتف ووسائل الاتصال عن بعد للتواصل مع المؤسسات. كما أن رضا الزبون أصبح عاملا مهما في سمعة المؤسسة واستمراريتها، لذلك فإن أي أداة تساعد على فهم مشاكل الزبائن وتحسين جودة الخدمة يمكن أن تكون مفيدة ومقبولة لدى المؤسسات.
العوامل التكنولوجية	تطور تقنيات الذكاء الاصطناعي، معالجة الصوت، التفريغ الآلي، وتحليل النصوص يساعد على إنجاز المشروع وتحسينه تدريجيا. كما أن توفر الحواسيب داخل المؤسسات يسمح بتشغيل التطبيق محليا دون الحاجة إلى تجهيزات معقدة في البداية.
العوامل البيئية	تحويل المكالمات إلى نصوص وملخصات يمكن أن يساهم في تقليل الحاجة إلى تخزين كميات كبيرة من الملفات الصوتية، مما يقلل من استهلاك موارد التخزين والطاقة على المدى الطويل، خاصة في المؤسسات ذات الحجم الكبير.

العوامل القانونية	يتعامل المشروع مع بيانات حساسة مرتبطة بمكالمات الزبائن، لذلك يجب احترام قواعد حماية البيانات والخصوصية. وتعتبر المعالجة المحلية داخل المؤسسة نقطة قوة لأنها تقلل من إرسال البيانات إلى أطراف خارجية، لكنها تتطلب وضع سياسات واضحة لحفظ البيانات واستعمالها.

3 تحليل القوى التنافسية PORTER

يساعد تحليل القوى التنافسية على فهم وضع المشروع داخل السوق ومدى جاذبية هذا القطاع. ويمكن عرض القوى الخمس كما يلي:

القوة التنافسية	التحليل
حدة المنافسة	تعتبر المنافسة المباشرة في السوق المحلي ضعيفة إلى متوسطة، لأنه لا توجد حلول محلية منتشرة، حسب حدود الدراسة، تقدم نفس الجمع بين التفريغ الآلي، التلخيص، تقدير الحالة الشعورية، والمعالجة المحلية لمكالمات خدمة العملاء. لكن توجد حلول دولية قريبة، خاصة في مجال تفريغ الاجتماعات أو منصات تحليل مراكز الاتصال.
قوة العملاء	تعتبر قوة العملاء متوسطة، لأن المؤسسات تبحث دائما عن حلول فعالة وبأسعار مناسبة. فإذا كان التطبيق سهل الاستعمال ويوفر قيمة واضحة مثل تقليل الوقت والتكلفة، يمكن أن يقنع العملاء بتجربته. لكن المؤسسات الكبيرة قد تطلب تخصيصات تقنية ودعما مستمرا قبل الاعتماد عليه.
قوة الموردين	في المرحلة الأولى، لا يعتمد المشروع على موردين كثر، لأنه يقوم أساسا على تطوير برمجي ونماذج ذكاء اصطناعي وأجهزة موجودة لدى المستخدمين. لكن قد تظهر الحاجة لاحقا إلى خدمات استضافة، معدات أقوى، أو أدوات تطوير مدفوعة، مما يجعل قوة الموردين متوسطة.

تهديد الخدمات البديلة	توجد بدائل جزئية مثل التسجيل الصوتي التقليدي، المراجعة اليدوية، برامج تحويل الصوت إلى نص، أو برامج إدارة علاقات العملاء. لكن هذه البدائل لا تقدم غالبا نفس الحل المتكامل المقترح في مشروعنا، خاصة من ناحية الجمع بين التفريغ، التلخيص، تقدير الحالة الشعورية، والبحث داخل قاعدة بيانات محلية.
تهديد دخول منافسين جدد	يعتبر هذا التهديد متوسطا، لأن تطوير تطبيق مشابه أصبح ممكنا مع توفر أدوات الذكاء الاصطناعي، لكن بناء حل عملي موجه للمؤسسات، قابل للتشغيل محليا، ويحافظ على الخصوصية، يحتاج إلى معرفة تقنية وتجربة ميدانية وفهم لاحتياجات خدمة العملاء.

بناء على هذا التحليل، يمكن القول إن المشروع يملك فرصة جيدة إذا تم تقديمه كحل بسيط، عملي، منخفض التكلفة، ومناسب لاحتياجات المؤسسات التي تعتمد على مكالمات خدمة العملاء.

4 تحليل SWOT

يساعد تحليل SWOT على تحديد نقاط القوة والضعف داخل المشروع، إضافة إلى الفرص والتهديدات الموجودة في البيئة الخارجية.

T التهديدات	O الفرص	W نقاط الضعف	S نقاط القوة
- دخول منافسين دوليين أو محليين بموارد أكبر. - تخوف بعض المؤسسات من استعمال الذكاء الاصطناعي في معالجة بيانات العملاء. - تغير القوانين المتعلقة بحماية البيانات والخصوصية. - التطور السريع للتكنولوجيا قد يتطلب تحديثات مستمرة. - صعوبة إقناع المؤسسات الكبيرة دون تجارب ميدانية قوية.	- تزايد حاجة المؤسسات إلى حلول رقمية لتحسين خدمة العملاء. - الاهتمام المتزايد بحلول الذكاء الاصطناعي والتحول الرقمي. - ضعف توفر حلول محلية متخصصة في هذا المجال. - إمكانية عقد شراكات مع مؤسسات تملك مراكز اتصال. - إمكانية تطوير التطبيق لدعم لغات ولهجات محلية.	- المشروع في بدايته ويحتاج إلى تجارب ميدانية أكثر. - دقة التفريغ والتحليل قد تتأثر بجودة الصوت واللهجات والضجيج. - محدودية الموارد المالية والبشرية في المرحلة الأولى. - الحاجة إلى بناء ثقة المؤسسات في استعمال حل جديد. - الحاجة إلى دعم تقني مستمر عند التوجه نحو المؤسسات الكبيرة.	- الجمع بين التفريغ الآلي والتلخيص وتقدير الحالة الشعورية في تطبيق واحد. - المعالجة المحلية داخل المؤسسة مما يعزز الخصوصية. - تقليل تكلفة التخزين بالاعتماد على النصوص والملخصات. - إمكانية تشغيل التطبيق على أجهزة موجودة مسبقا. - قابلية التخصيص حسب حجم المؤسسة واحتياجاتها.

من خلال هذا التحليل، يتضح أن المشروع يمتلك نقاط قوة مهمة، خاصة من حيث الخصوصية والمعالجة المحلية وتخفيض التكلفة. وفي المقابل، يجب التعامل بواقعية مع نقاط الضعف، خصوصا الحاجة إلى تجارب ميدانية وتحسين أداء التطبيق في ظروف صوتية مختلفة.

5 المزيج التسويقي

يعتمد المزيج التسويقي للمشروع على أربعة عناصر أساسية، وهي المنتج، السعر، التوزيع، والترويج.

العنصر	التوضيح

المنتج	تطبيق مكتبي ذكي لتحليل مكالمات خدمة العملاء، يسمح بتفريغ المكالمات، تقدير الحالة الشعورية العامة للعميل، تلخيص المكالمة، حفظ النتائج، والبحث داخل قاعدة البيانات. ويمكن تقديم نسخة بسيطة للمؤسسات الصغيرة، ونسخة مخصصة للمؤسسات الكبيرة.
السعر	يعتمد المشروع على سياسة سعرية مرنة حسب نوع العميل. فالمؤسسات الكبيرة يمكن التعامل معها من خلال عقود تطوير ونشر وصيانة سنوية، بينما تستفيد المؤسسات الصغيرة من رخص قصيرة المدى أو اشتراكات شهرية أو سنوية، مع إمكانية توفير نسخة تجريبية محدودة الخصائص.
التوزيع	يتم توزيع التطبيق مباشرة على المؤسسات من خلال التواصل التجاري، العروض التجريبية، والموقع الإلكتروني الخاص بالمشروع مستقبلا. وبالنسبة للمؤسسات الكبيرة، يتم الاعتماد على تثبيت مخصص ودعم تقني مباشر.
الترويج	يعتمد الترويج في البداية على التسويق الرقمي، العروض التوضيحية، التواصل مع المؤسسات، والمشاركة في فعاليات الابتكار والمقاولاتية. وتركز الرسائل التسويقية على تقليل التكلفة، حماية البيانات، سهولة البحث في المكالمات، وتحسين جودة خدمة العملاء.

6 الاستراتيجيات التسويقية

تعتمد الاستراتيجية التسويقية للمشروع على عدة محاور:

- استراتيجية التميز: يتم التركيز على تمييز المنتج من خلال كونه يجمع بين عدة خصائص في تطبيق واحد، وهي التفريغ الآلي، التلخيص، تقدير الحالة الشعورية، البحث في قاعدة البيانات، والمعالجة المحلية.
- استراتيجية التركيز: في المرحلة الأولى، يتم التركيز على المؤسسات التي تعتمد على المكالمات بشكل واضح، مثل شركات الاتصالات، البنوك، شركات التأمين، التجارة الإلكترونية، ومراكز الدعم التقني. وهذا يسمح بتوجيه الجهود نحو فئة تحتاج فعلا إلى الحل المقترح.
- استراتيجية الثقة والخصوصية: يتم إبراز أن التطبيق يعالج البيانات داخل المؤسسة، مما يساعد على حماية خصوصية الزبائن وتقليل الاعتماد على خدمات خارجية. هذه النقطة مهمة خاصة بالنسبة للمؤسسات التي تتعامل مع بيانات حساسة.

- استراتيجية التجربة المجانية: يمكن تقديم نسخة تجريبية محدودة للمؤسسات الصغيرة أو للمستخدمين الأوائل، حتى يتمكنوا من تجربة التطبيق وفهم فائدته قبل شراء رخصة أو طلب نسخة مخصصة.
- استراتيجية العقود المخصصة: بالنسبة للمؤسسات الكبيرة، يتم تقديم عروض خاصة تشمل تخصيص التطبيق، ربطه بقاعدة بيانات المؤسسة، توفير الدعم التقني، والصيانة الدورية.
- استراتيجية التطوير المستمر: يتم جمع ملاحظات المستخدمين وتحسين التطبيق تدريجيا، سواء من حيث الواجهة، دقة النتائج، سرعة المعالجة، أو إضافة خصائص جديدة يحتاجها السوق.

من خلال هذه الاستراتيجيات، يسعى المشروع إلى دخول السوق بطريقة واقعية، تبدأ من نموذج أولي قابل للتجربة، ثم تتوسع تدريجيا نحو المؤسسات التي تحتاج إلى حلول عملية لتحليل مكالمات خدمة العملاء وتحسين جودة التفاعل مع الزبائن.

المحور 4

المحور الرابع: خطة الإنتاج والتنظيم

1 خطة الإنتاج

تتمثل خطة الإنتاج في هذا المشروع في المراحل والوسائل اللازمة لتطوير التطبيق المكتبي، تشغيله، اختباره، ثم تقديمه للعملاء في شكل نسخة قابلة للاستعمال. وبما أن المشروع عبارة عن منتج رقمي يعتمد على الذكاء الاصطناعي، فإن عملية الإنتاج لا ترتبط بتصنيع مادي، بل ترتبط أساسا بتطوير البرمجيات، إعداد النماذج، اختبار الأداء، وتجهيز النسخة التجارية حسب احتياجات المستخدمين.

يعتمد إنتاج التطبيق على مجموعة من الموارد التقنية والبشرية والتنظيمية، ويمكن عرضها كما يلي:

العنصر	التوضيح
اختيار الموقع	في المرحلة الأولى، يمكن إنجاز المشروع من مكتب صغير أو مساحة عمل بسيطة، لأن طبيعة المشروع رقمية ولا تتطلب ورشة إنتاج أو مساحة كبيرة. وعند التوسع، يمكن تخصيص مقر صغير للتطوير، التجارب التقنية، واستقبال العملاء أو الشركاء.
الموارد الأولية	تتمثل في الحاسوب، الاتصال بالإنترنت، بيئة البرمجة، قاعدة البيانات، نماذج التفريغ الآلي، تحليل الحالة الشعورية، والتلخيص الآلي، إضافة إلى تسجيلات صوتية أو مكالمات تجريبية لاختبار التطبيق وتحسينه.
الأجهزة والمعدات	يحتاج المشروع في مرحلته الأولى إلى حاسوب للتطوير والاختبار، ميكروفون أو جهاز تسجيل صوتي عند الحاجة، وسماعات لاختبار جودة الصوت. وعند التوجه إلى المؤسسات، يتم تشغيل التطبيق على حواسيب أعوان خدمة العملاء الموجودة مسبقا.
البرمجيات والأدوات	يعتمد المشروع على أدوات تطوير التطبيقات المكتبية، لغات البرمجة، مكتبات معالجة الصوت، مكتبات الذكاء الاصطناعي، ونظام لإدارة قاعدة البيانات، إضافة إلى أدوات لإدارة المشروع واختبار الأداء.

قاعدة البيانات	تستعمل لحفظ نتائج المكالمات، مثل النص الناتج عن التفريغ، ملخص المكالمة، الحالة الشعورية العامة، تاريخ المكالمة، وبيانات البحث. ويمكن أن تكون محلية في النسخة البسيطة أو مرتبطة بقاعدة بيانات المؤسسة في النسخة المخصصة.
طريقة الإنتاج	يتم إنتاج النسخة الأولى من خلال تطوير الواجهة، دمج وظائف معالجة الصوت، ربط النماذج الذكية بالتطبيق، ثم تخزين النتائج داخل قاعدة البيانات. بعد ذلك يتم اختبار التطبيق وتحسينه حسب النتائج والملاحظات.
الطاقة الإنتاجية	بما أن المنتج رقمي، فإن الطاقة الإنتاجية لا تقاس بعدد الوحدات المصنعة، بل بعدد النسخ التي يمكن تثبيتها أو تخصيصها للعملاء، سواء للمؤسسات الصغيرة أو المؤسسات الكبيرة.
الجودة والاختبار	يتم التأكد من جودة التطبيق من خلال اختبار دقة التفريغ، وضوح الملخص، سرعة المعالجة، سهولة استعمال الواجهة، وسلامة تخزين البيانات، مع تجربة التطبيق في ظروف صوتية مختلفة.

تمر عملية إنتاج التطبيق بعدة مراحل أساسية، يمكن تلخيصها كما يلي:

المرحلة	الوصف
مرحلة التحليل	تحديد احتياجات العملاء المحتملين، خاصة المؤسسات التي تعتمد على خدمة العملاء، وتحديد الخصائص الأساسية التي يجب أن يوفرها التطبيق.
مرحلة التصميم	تصميم واجهة التطبيق، تحديد طريقة عرض النص والملخص والحالة الشعورية، وتصميم قاعدة البيانات الخاصة بنتائج المكالمات.
مرحلة التطوير	برمجة التطبيق المكتبي، وربطه بوحدات التفريغ الآلي، تحليل الحالة الشعورية، والتلخيص الآلي، إضافة إلى قاعدة البيانات.
مرحلة الاختبار	اختبار التطبيق باستعمال مكالمات مسجلة ومباشرة للتأكد من عمل الوظائف الأساسية، واكتشاف الأخطاء التقنية، وتحسين عرض النتائج.

مرحلة التحسين	تحسين سرعة المعالجة، واجهة الاستخدام، طريقة التخزين، وإضافة خصائص جديدة حسب الحاجة.
مرحلة الإطلاق الأولي	تقديم نسخة تجريبية محدودة للمستخدمين الأوائل أو المؤسسات الصغيرة، بهدف جمع الملاحظات وتقييم قابلية الاستعمال.
مرحلة التخصيص التجاري	بالنسبة للمؤسسات الكبيرة، يتم تخصيص التطبيق حسب نظام العمل الداخلي، عدد المستخدمين، طريقة تخزين البيانات، ومتطلبات الأمن والخصوصية.

2 خطة التنظيم

تعتمد خطة التنظيم على توزيع المهام داخل المشروع بطريقة بسيطة وواضحة، خاصة أن المشروع في مرحلته الأولى يعتمد على صاحب المشروع كمطور رئيسي. ومع تطور المشروع، يمكن إضافة أعضاء جدد حسب الحاجة، خصوصا في الجوانب المتعلقة بالتسويق، الدعم التقني، أمن المعلومات، وخدمة العملاء.

في المرحلة الحالية، يتم تنظيم العمل كما يلي:

الوظيفة	المهام الأساسية
صاحب المشروع والمطور الرئيسي	تصميم وبرمجة التطبيق، دمج نماذج الذكاء الاصطناعي، إعداد قاعدة البيانات، اختبار النسخة الأولية، متابعة الأخطاء، وتطوير الخصائص الجديدة.
مسؤول الدراسة التجارية	دراسة السوق، تحديد العملاء المستهدفين، إعداد العروض التجارية، متابعة المنافسين، وتحضير خطة التسويق. وفي المرحلة الأولى يمكن لصاحب المشروع القيام بهذه المهمة.
مسؤول الدعم التقني	مساعدة المستخدمين، تثبيت التطبيق، معالجة المشاكل التقنية، وتقديم التوضيحات للمؤسسات التي تستعمل التطبيق. ويمكن إضافة هذا المنصب عند بداية التعامل مع عملاء حقيقيين.
مسؤول التسويق والاتصال	الترويج للتطبيق، إعداد المحتوى التعريفي، التواصل مع المؤسسات، تنظيم العروض التجريبية، وإدارة صفحات المشروع على المنصات الرقمية.

مختص أمن المعلومات	تظهر الحاجة إلى هذا الدور عند التعامل مع مؤسسات كبيرة أو بيانات حساسة، حيث يساعد على حماية قاعدة البيانات، تنظيم صلاحيات الوصول، وضمان سرية مكالمات العملاء.

أما من حيث التنظيم الداخلي للمشروع، فيمكن تقسيم العمل إلى عدة مساحات أو وظائف رئيسية:

- مساحة البحث والتطوير: تهتم بتحسين نماذج التفريغ، تحليل الحالة الشعورية، والتلخيص الآلي، إضافة إلى تطوير خصائص جديدة داخل التطبيق.
- مساحة التطوير التقني: تهتم ببرمجة التطبيق، تحسين الواجهة، ربط قاعدة البيانات، وتجهيز النسخ الخاصة بالعملاء.
- مساحة الاختبار والجودة: تهتم بتجربة التطبيق على مكالمات مختلفة، تسجيل الأخطاء، وتحسين الأداء قبل تسليم النسخة النهائية.
- مساحة التسويق والعلاقات مع العملاء: تهتم بالتواصل مع المؤسسات، تقديم العروض التجريبية، وشرح القيمة التي يقدمها التطبيق.
- مساحة الدعم والمتابعة: تهتم بمرافقة المستخدمين بعد تثبيت التطبيق، جمع الملاحظات، واقتراح التحسينات المستقبلية.

3 الموردون والشركاء الرئيسيون

بما أن المشروع رقمي، فإن الموردين لا يتمثلون في مواد خام تقليدية، بل في أدوات وخدمات تقنية تساعد على تطوير التطبيق وتشغيله. ويمكن تحديد أهم الموردين والشركاء كما يلي:

الطرف	الدور في المشروع
مزودو خدمات الإنترنت	توفير اتصال مستقر بالإنترنت من أجل تحميل الأدوات، تحديث التطبيق، التواصل مع العملاء، وإدارة النسخ أو الدعم عن بعد.
مزودو العتاد المعلوماتي	توفير الحواسيب، الميكروفونات، وحدات التخزين، أو أي معدات ضرورية للتطوير والاختبار.

المؤسسات المستعملة للتطبيق	تمثل شريكا مهما في اختبار التطبيق وتجربته ميدانيا، خاصة المؤسسات التي تمتلك خدمة عملاء أو تستقبل مكالمات بشكل منتظم.
حاضنة الأعمال الجامعية	يمكن أن تساعد في التوجيه، المرافقة، تطوير النموذج التجاري، وربط المشروع بفرص الدعم أو التكوين أو الشراكة.
مختصون في خدمة العملاء	يمكن الاستفادة من خبرتهم لفهم طريقة سير المكالمات، نوعية المعلومات المهمة، وطريقة عرض النتائج التي يحتاجها الأعوان والمشرفون.
مختصون في الأمن السيبراني	يمكن التعاون معهم عند تطوير نسخة موجهة للمؤسسات الكبيرة، خاصة لضمان حماية البيانات وتنظيم صلاحيات الوصول.

4 الشراكات الرئيسية

تعتبر الشراكات عاملا مهما في تطوير المشروع ونقله من مرحلة النموذج الأولي إلى مرحلة المنتج التجاري. ويمكن للمشروع أن يعتمد على الشراكات التالية:

- شراكات مع المؤسسات التي تملك خدمة عملاء: من أجل تجربة التطبيق في ظروف واقعية وجمع ملاحظات حول فعاليته وسهولة استعماله.
- شراكات مع مراكز الاتصال: بهدف اختبار التطبيق على عدد أكبر من المكالمات ومعرفة احتياجات المشرفين والأعوان.
- شراكات مع حاضنات الأعمال: من أجل الحصول على المرافقة في الجانب الإداري والتجاري، وتحسين طريقة تقديم المشروع للمستثمرين أو العملاء.
- شراكات تقنية: مع مختصين أو فرق تقنية يمكنها المساعدة في تحسين الأداء، الأمن، أو التكامل مع أنظمة المؤسسات.
- شراكات مع مؤسسات صغيرة: لتوفير نسخة تجريبية محدودة، والحصول على آراء المستخدمين قبل التوسع نحو مؤسسات أكبر.

5 المهام الرئيسية

تتمثل المهام الرئيسية للمشروع في ما يلي:

- تطوير وتحسين التطبيق المكتبي.

- دمج نماذج التفريغ الآلي وتحليل الحالة الشعورية والتلخيص الآلي.
- إعداد قاعدة البيانات وتطوير خاصية البحث.
- اختبار التطبيق على مكالمات مسجلة ومباشرة.
- تحسين واجهة الاستخدام وتجربة المستخدم.
- إعداد نسخة تجريبية للمؤسسات الصغيرة.
- التواصل مع العملاء المحتملين وتقديم عروض تجريبية.
- تقديم الدعم التقني والصيانة عند الحاجة.
- تطوير نسخ مخصصة للمؤسسات الكبيرة حسب احتياجاتها.

6 الشكل التنظيمي المقترح

في المرحلة الأولى، يكون التنظيم بسيطا لأن المشروع يعتمد أساسا على صاحب المشروع. ومع تطور النشاط، يمكن اعتماد شكل تنظيمي تدريجي كما يلي:

المستوى التنظيمي	المهام
الإدارة العامة	تحديد أهداف المشروع، متابعة التقدم، اتخاذ القرارات، والبحث عن فرص التعاون والتمويل.
القسم التقني	تطوير التطبيق، تحسين النماذج، معالجة الأخطاء، إدارة قاعدة البيانات، وتجهيز النسخ الخاصة بالعملاء.
قسم التسويق والمبيعات	التواصل مع العملاء، إعداد العروض التجارية، الترويج للتطبيق، وتنظيم التجارب الأولية.
قسم الدعم والمتابعة	مساعدة المستخدمين، متابعة المشاكل التقنية، جمع الملاحظات، واقتراح التحسينات.
قسم الأمن والخصوصية	تحديد سياسات حماية البيانات، تنظيم صلاحيات الوصول، ومتابعة الجوانب المتعلقة بسرية مكالمات العملاء.

من خلال هذه الخطة، يمكن للمشروع أن يبدأ بموارد بسيطة ومحدودة، ثم يتوسع تدريجيا حسب عدد العملاء

وحجم الطلب. وتتميز طبيعة المشروع بأنها لا تحتاج في البداية إلى استثمارات كبيرة في الإنتاج المادي، بل تعتمد أساسا على تطوير برمجي، تحسين مستمر، وتجارب ميدانية مع المؤسسات المستهدفة.

المحور 5

المحور الخامس: الخطة المالية

1 التكاليف والأعباء

تعتبر الخطة المالية من أهم العناصر التي تسمح بتقدير قابلية المشروع للتطبيق من الناحية الاقتصادية. وبما أن المشروع يتمثل في تطبيق مكتبي رقمي يعتمد أساسا على البرمجة والذكاء الاصطناعي، فإن تكاليف الانطلاق في المرحلة الأولى تبقى محدودة نسبيا مقارنة بالمشاريع التي تحتاج إلى تجهيزات إنتاج مادية أو مواد أولية كبيرة.

تعتمد النسخة الأولية من المشروع على وجود مطور رئيسي، حاسوب للتطوير، اتصال بالإنترنت، وبعض المصاريف الخاصة بالتسويق والتجربة. لذلك يمكن تقدير تكلفة الانطلاق الأولية بحوالي 150,000 دج، وهي تكلفة موجهة أساسا لإنجاز النموذج الأولي، تحسينه، وتجهيزه للتجارب الأولى مع المستخدمين المحتملين.

أما عند الانتقال إلى نسخة مؤسساتية مخصصة، فقد ترتفع التكاليف حسب حجم المؤسسة، عدد المستخدمين، نوع التكامل المطلوب، ومستوى الحماية والدعم التقني المطلوب.

يمكن تلخيص التكاليف الأولية للمشروع كما يلي:

العنصر	التوضيح	التكلفة التقديرية
حاسوب للتطوير والاختبار	يستعمل في برمجة التطبيق، تشغيل النماذج، اختبار المعالجة، وتجهيز النسخة الأولية.	100,000 دج
الإنترنت وأدوات العمل	اشتراك إنترنت، تحميل الأدوات، تحديث المكتبات، والتواصل مع المستخدمين أو المؤسسات.	10,000 دج
معدات بسيطة للاختبار	ميكروفون أو سماعات أو ملحقات بسيطة لتجربة تسجيل ومعالجة الصوت.	10,000 دج

التسويق والإشهار الأولي	إعلانات بسيطة على مواقع التواصل، إعداد عرض تعريفي، وتصميم محتوى للتعريف بالتطبيق.	20,000 دج
مصاريف إدارية ومختلفة	مصاريف غير متوقعة، تنقلات بسيطة، طباعة وثائق، أو إعداد ملفات عرض المشروع.	10,000 دج
المجموع	التكلفة الإجمالية التقديرية للانطلاق الأولي	150,000 دج

إضافة إلى تكاليف الانطلاق، توجد بعض الأعباء الشهرية أو الدورية التي يمكن أن تظهر أثناء تشغيل المشروع، خاصة عند بداية تسويق التطبيق أو تقديم الدعم للعملاء. ويمكن عرضها كما يلي:

نوع العبء	التوضيح	التكلفة الشهرية التقديرية
الإنترنت	استعمال الإنترنت في التطوير، التحديث، التواصل مع العملاء، والدعم عن بعد.	3,000 دج
التسويق الرقمي	إعلانات بسيطة، نشر محتوى، أو ترويج للنسخة التجريبية.	10,000 دج
الصيانة والتحديث	تحسين التطبيق، تصحيح الأخطاء، وتطوير خصائص جديدة.	حسب الحاجة
الدعم والتنقل	تنقلات أو تواصل مع مؤسسات من أجل عرض التطبيق أو تثبيته.	5,000 دج
مصاريف مختلفة	مصاريف إدارية أو تقنية غير متوقعة.	حسب الحاجة
المجموع الشهري التقريبي	أعباء تشغيلية أولية قابلة للتغير حسب النشاط	17,000 دج

الهيكل التمويلي:

يتم تمويل المشروع في مرحلته الأولى بالاعتماد على التمويل الذاتي لصاحب المشروع، وذلك لأن تكلفة الانطلاق الأولية محدودة نسبيا ولا تتطلب استثمارات كبيرة. ويمكن لاحقا الاستعانة ببرامج دعم المؤسسات الناشئة أو بحاضنة الأعمال عند الانتقال إلى نسخة تجارية أوسع.

البيان	النسبة	القيمة
أموال خاصة	100%	150,000 دج
قروض	0%	0 دج
المجموع	100%	150,000 دج

2 رقم الأعمال

يعتمد رقم أعمال المشروع على مصادر دخل متعددة، تختلف حسب نوع العميل وحجم المؤسسة. فالمشروع لا يعتمد على بيع منتج واحد فقط، بل يمكن أن يحقق دخله من خلال عقود مخصصة مع المؤسسات الكبيرة، ورخص استعمال للمؤسسات الصغيرة، ونسخة مجانية محدودة الخصائص يمكن أن تستعمل للتعريف بالتطبيق وجذب المستخدمين.

تتمثل مصادر الدخل الأساسية فيما يلي:

- عقود المؤسسات الكبيرة: وهي عقود تطوير أو تخصيص أو نشر التطبيق داخل مؤسسة كبيرة، مثل شركات الاتصالات، البنوك، شركات التأمين، أو مراكز الاتصال. يشمل العقد التثبيت، التخصيص، الدعم التقني، وربط التطبيق بقاعدة بيانات المؤسسة عند الحاجة.
- رخص الاستعمال قصيرة المدى: وهي موجهة للمؤسسات الصغيرة والمصغرة، حيث يمكن للعميل شراء رخصة شهرية أو سنوية لاستعمال التطبيق بعدد محدود من الأجهزة أو بعدد محدود من المكالمات.
- اشتراكات شهرية أو سنوية: يمكن توفير اشتراك بسيط للمؤسسات التي تريد استعمال التطبيق بشكل مستمر دون عقد مخصص.
- الصيانة والدعم التقني: يمكن أن يكون الدعم التقني مصدرا إضافيا للدخل، خاصة بالنسبة للمؤسسات التي تحتاج إلى متابعة دورية أو تحديثات مخصصة.
- الإعلانات في النسخة المجانية: يمكن استعمال الإعلانات كمصدر دخل ثانوي فقط في النسخة المجانية أو المحدودة، مع الحرص على عدم التأثير على تجربة المستخدم أو الطابع المهني للتطبيق.

يمكن اقتراح التسعير الأولي التالي، مع إمكانية تعديله حسب السوق وتجارب العملاء:

العرض	الفئة المستهدفة	السعر التقديري
نسخة تجريبية مجانية	مؤسسات صغيرة أو مستخدمون أوائل، مع عدد محدود من المكالمات والخصائص.	مجانية
رخصة شهرية بسيطة	مؤسسات صغيرة ومصغرة تحتاج إلى معالجة عدد محدود من المكالمات.	3,000 دج شهريا
رخصة سنوية بسيطة	مؤسسات صغيرة تريد استعمال التطبيق لمدة أطول بتكلفة أقل من الدفع الشهري.	30,000 دج سنويا
نسخة مهنية للمؤسسات المتوسطة	مؤسسات تحتاج إلى عدد أكبر من المكالمات، قاعدة بيانات محلية، ودعم تقني محدود.	80,000 دج إلى 150,000 دج سنويا
عقد مخصص للمؤسسات الكبيرة	مراكز اتصال، شركات اتصالات، بنوك، شركات تأمين، أو مؤسسات تحتاج إلى تخصيص وربط داخلي.	ابتداء من 300,000 دج حسب المتطلبات
الصيانة والدعم	تحديثات، متابعة تقنية، إصلاح أخطاء، وتعديلات بسيطة بعد التسليم.	حسب العقد

وبناء على هذه الفرضيات، يمكن وضع تقدير أولي لرقم الأعمال خلال السنوات الأولى كما يلي:

	PREVISION					
Produit A destiné Client	N	N+1	N+2	N+3	N+4	N+5
Quantité produit A	10	15	20	25	30	35
Prix HT produit A	000,00.200	000,00.200	000,00.180	000,00.180	000,00.150	000,00.150
Ventes produit A	10	15	20	25	30	35
CHIFFRE D'AFFAIRES GLOBAL	000,00.000.2	000,00.000.3	000,00.600.3	000,00.500.4	000,00.500.4	000,00.250.5

تبقى هذه الأرقام تقديرية وقابلة للتعديل حسب نتائج التجارب الأولى، قدرة المشروع على جذب العملاء، ومدى تقبل السوق لهذا النوع من الحلول. كما أن الحصول على عقد واحد مع مؤسسة كبيرة يمكن أن يرفع رقم الأعمال بشكل واضح مقارنة بالاعتماد فقط على الرخص الصغيرة.

3 خطة الخزينة

تهدف خطة الخزينة إلى متابعة التدفقات المالية الداخلة والخارجة خلال المرحلة الأولى من المشروع، من أجل معرفة قدرة المشروع على تغطية مصاريفه وتحقيق التوازن المالي تدريجيا. وفي بداية النشاط، من الطبيعي أن تكون المصاريف أعلى من المداخيل، خاصة خلال مرحلة التطوير والتجربة والتسويق الأولي.

يمكن عرض خطة خزينة مبسطة للسنة الأولى كما يلي:

الفترة	المداخيل المتوقعة	المصاريف المتوقعة	الرصيد المتوقع
الثلاثي الأول	0 دج	100,000 دج	-100,000 دج
الثلاثي الثاني	30,000 دج	40,000 دج	-10,000 دج
الثلاثي الثالث	50,000 دج	35,000 دج	15,000 دج
الثلاثي الرابع	70,000 دج	35,000 دج	35,000 دج
المجموع السنوي	150,000 دج	210,000 دج	-60,000 دج

يوضح هذا الجدول أن المشروع قد يحتاج في السنة الأولى إلى تمويل ذاتي أو دعم خارجي بسيط لتغطية فترة التطوير والانطلاق، خاصة قبل الوصول إلى عدد كاف من العملاء. لكن هذا الوضع يعتبر طبيعيا بالنسبة لمشروع رقمي في مرحلته الأولى، حيث تكون الأولوية هي تطوير المنتج، اختباره، وجمع ملاحظات المستخدمين.

ومع بداية بيع الرخص أو توقيع عقود صغيرة، يمكن أن تتحسن الخزينة تدريجيا. أما في حالة الحصول على عقد مخصص مع مؤسسة متوسطة أو كبيرة، فإن المشروع يمكن أن يغطي تكاليفه بسرعة أكبر.

4 جدول حسابات النتائج المتوقع

يسمح جدول حسابات النتائج المتوقع بتقدير الفرق بين رقم الأعمال والتكاليف خلال السنوات الأولى من المشروع. وبما أن المشروع يعتمد على منتج رقمي، فإن التكاليف الثابتة تكون أكبر في البداية، بينما يمكن أن ترتفع الأرباح لاحقا مع زيادة عدد العملاء دون الحاجة إلى زيادة كبيرة في تكاليف الإنتاج.

العنصر	السنة 1	السنة 2	السنة 3	السنة 4	السنة 5
رقم الأعمال	150,000 دج	450,000 دج	900,000 دج	1,500,000 دج	2,500,000 دج
تكاليف التطوير والصيانة	120,000 دج	150,000 دج	200,000 دج	300,000 دج	400,000 دج
تكاليف التسويق	40,000 دج	80,000 دج	120,000 دج	180,000 دج	250,000 دج
تكاليف الدعم والتنقل	30,000 دج	60,000 دج	100,000 دج	150,000 دج	220,000 دج
مصاريف أخرى	20,000 دج	40,000 دج	60,000 دج	80,000 دج	120,000 دج
مجموع التكاليف	210,000 دج	330,000 دج	480,000 دج	710,000 دج	990,000 دج
النتيجة الصافية التقديرية	-60,000 دج	120,000 دج	420,000 دج	790,000 دج	1,510,000 دج

من خلال هذا التقدير، يمكن ملاحظة أن المشروع قد لا يحقق ربحا صافيا في السنة الأولى بسبب تكاليف الانطلاق والتطوير، لكنه يمكن أن يبدأ في تحقيق نتيجة إيجابية ابتداء من السنة الثانية إذا نجح في بيع عدد كاف من الرخص أو الحصول على أول عقود مؤسساتية.

5 نقطة التعادل

تمثل نقطة التعادل المستوى الذي تصبح فيه مداخيل المشروع مساوية لمصاريفه، أي أن المشروع لا يحقق خسارة ولا ربحا. وبما أن التكاليف الأولية للمشروع تقدر بحوالي 150,000 دج، والأعباء التشغيلية الأولية قد تصل إلى حوالي 23,000 دج شهريا، فإن الوصول إلى نقطة التعادل يتطلب تحقيق مداخيل شهرية تغطي هذه المصاريف.

على سبيل المثال، إذا تم اعتماد رخصة شهرية بقيمة 3,000 دج، فإن المشروع يحتاج إلى حوالي 8 رخص شهرية لتغطية الأعباء التشغيلية الشهرية الأساسية. أما إذا تم بيع رخص سنوية أو توقيع عقد واحد مع مؤسسة متوسطة أو كبيرة، فإن الوصول إلى نقطة التعادل يكون أسرع.

يمكن عرض ذلك كما يلي:

نوع الدخل	القيمة التقديرية	الأثر على نقطة التعادل

رخصة شهرية بسيطة	3,000 دج شهريا	تحتاج المؤسسة الناشئة إلى حوالي 8 رخص شهرية لتغطية الأعباء الشهرية الأساسية.
رخصة سنوية بسيطة	30,000 دج سنويا	بيع خمس رخص سنوية يمكن أن يغطي تقريبا تكلفة الانطلاق الأولية.
نسخة مهنية لمؤسسة متوسطة	80,000 دج إلى 150,000 دج	عقد واحد أو عقدان من هذا النوع يمكن أن يغطي جزءا كبيرا من التكاليف الأولية.
عقد مؤسسة كبيرة	ابتداء من 300,000 دج	يمكن لعقد واحد أن يغطي تكلفة الانطلاق ويمنح المشروع قدرة أكبر على التطوير.

بناء على ذلك، فإن أفضل استراتيجية مالية للمشروع هي عدم الاعتماد فقط على النسخة المجانية أو الرخص الصغيرة، بل محاولة الحصول على عقود مخصصة مع مؤسسات متوسطة أو كبيرة، لأنها تسمح بتغطية التكاليف بسرعة أكبر وتساعد على تطوير التطبيق بشكل أكثر احترافية.

6 خلاصة الخطة المالية

تبين الخطة المالية أن المشروع قابل للانطلاق بتكلفة أولية محدودة نسبيا تقدر بحوالي 150,000 دج، وذلك لأن المنتج رقمي ولا يحتاج إلى معدات إنتاج كبيرة. كما أن مصادر الدخل المقترحة متنوعة وتشمل العقود المؤسساتية، الرخص قصيرة المدى، الاشتراكات، الدعم التقني، والإعلانات في النسخة المجانية بشكل ثانوي.

ورغم أن السنة الأولى قد تعرف عجزا ماليا بسيطا بسبب تكاليف التطوير والتسويق، إلا أن المشروع يمكن أن يصبح مربحا تدريجيا عند الوصول إلى عدد كاف من العملاء أو عند الحصول على عقود مخصصة مع مؤسسات تعتمد على خدمة العملاء بشكل مستمر. لذلك فإن نجاح المشروع ماليا يرتبط أساسا بجودة النسخة الأولية، قوة العرض التجاري، وقدرة التطبيق على إثبات فائدته في تقليل الوقت والتكلفة وتحسين استغلال مكالمات خدمة العملاء.

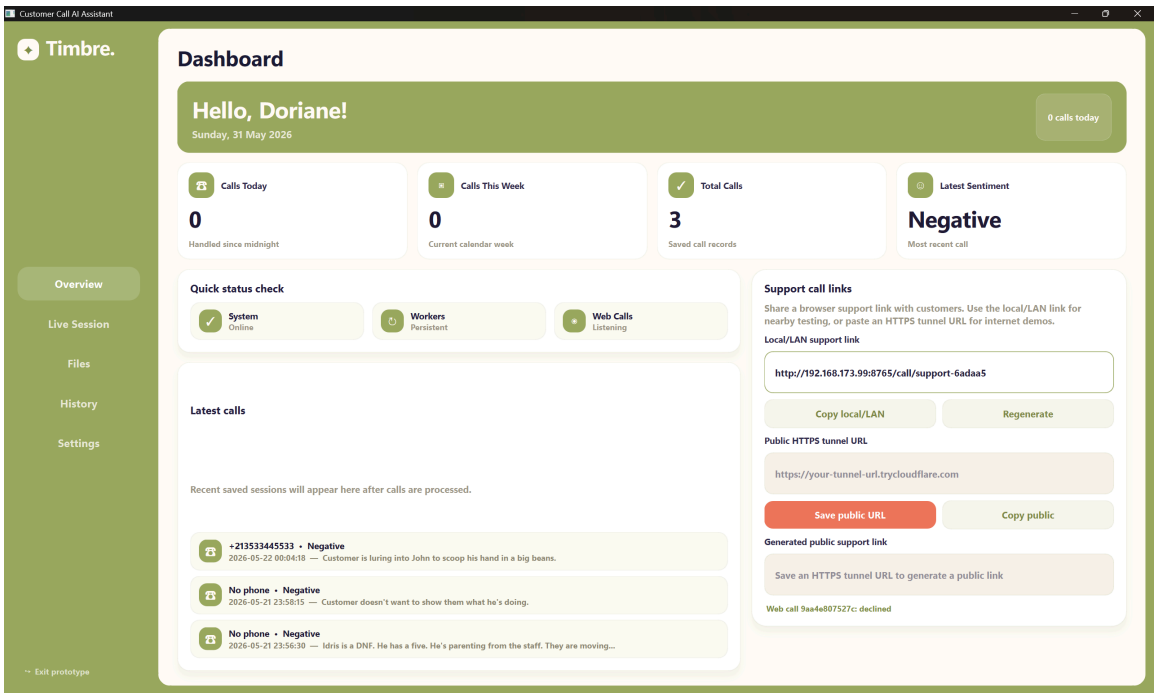
المحور 6

المحور السادس: النموذج الأولي التجريبي

1 تقديم النموذج الأولي

يتمثل النموذج الأولي التجريبي للمشروع في تطبيق مكتبي ذكي موجه لتحليل مكالمات خدمة العملاء، حيث يسمح للمستخدم بتفريغ المكالمات الصوتية إلى نص مكتوب، تقدير الحالة الشعورية العامة للعميل، ثم تلخيص محتوى المكالمة وحفظ النتائج في قاعدة بيانات يمكن الرجوع إليها لاحقا.

تم تطوير هذا النموذج الأولي من أجل إثبات قابلية تطبيق الفكرة من الناحية التقنية والتجارية. فهو لا يمثل النسخة النهائية الكاملة للمنتج، ولكنه يوضح الوظائف الأساسية التي يمكن أن تقدم قيمة حقيقية للمؤسسات التي تعتمد على خدمة العملاء. كما يسمح هذا النموذج بجمع الملاحظات من المستخدمين الأوائل وتحسين التطبيق قبل الانتقال إلى نسخة تجارية أكثر اكتمالا.



شكل 1: لوحة التحكم الرئيسية للتطبيق

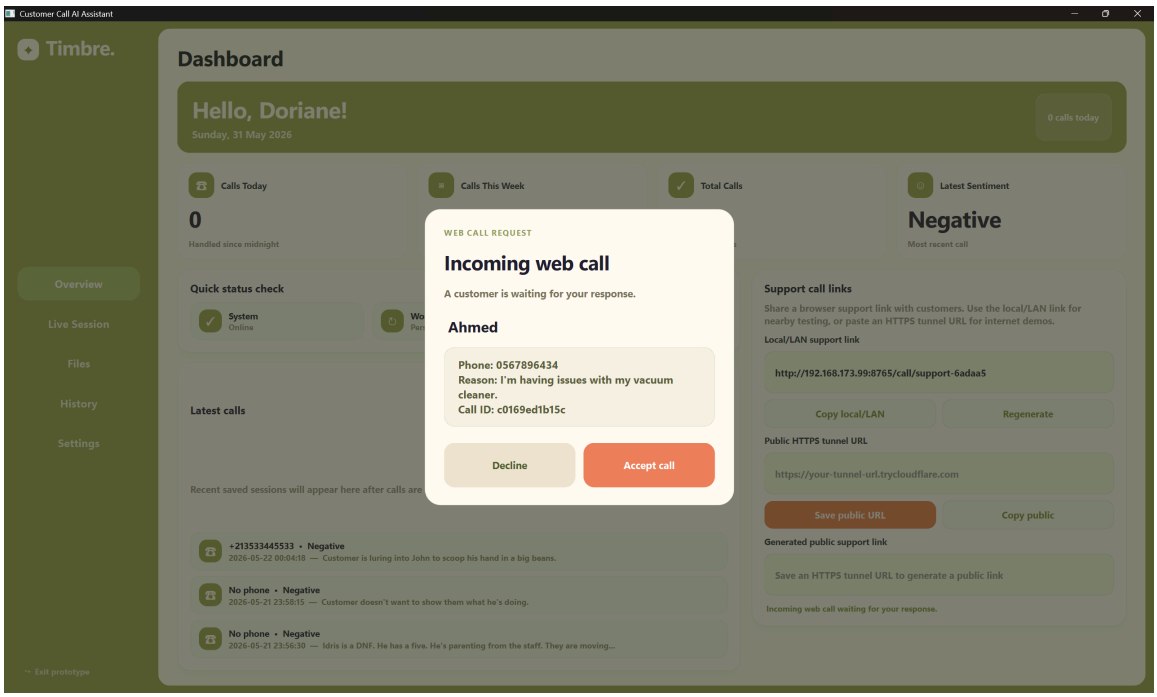
يرتكز النموذج الأولي على فكرة بسيطة، وهي تحويل المكالمة من ملف صوتي أو تفاعل مباشر يصعب تحليله يدويا إلى بيانات واضحة ومنظمة، تتكون من النص الكامل للمكالمة، ملخص مختصر، ومؤشر عام حول الحالة

الشعورية للعميل.

2 وصف التطبيق

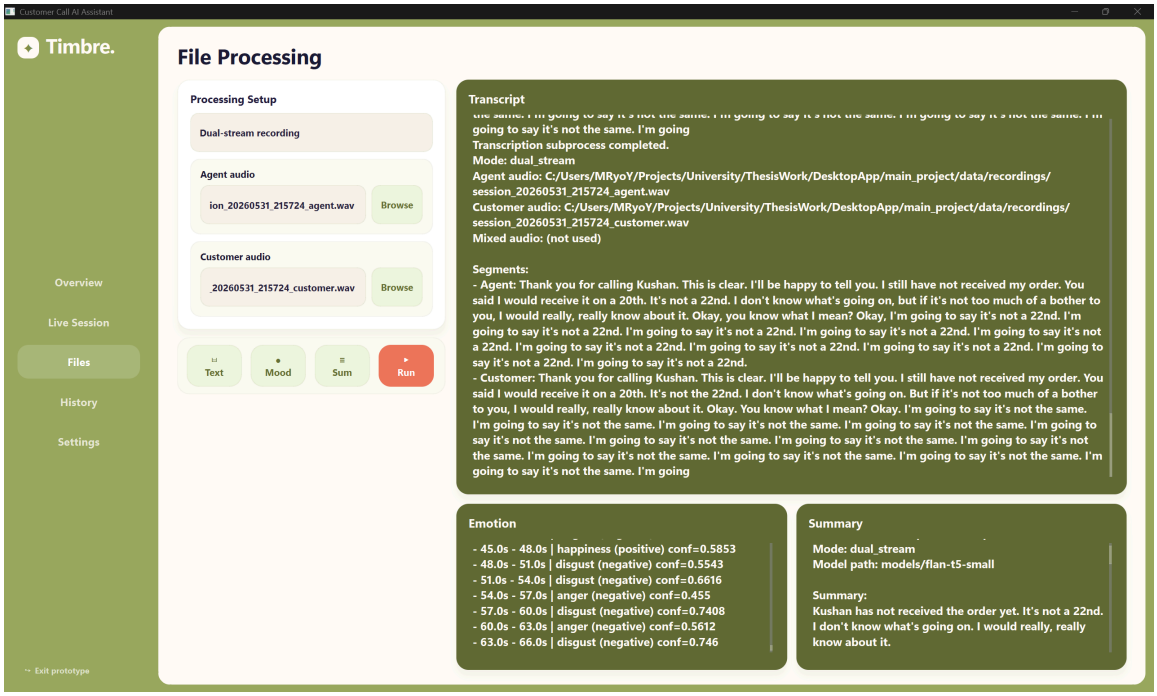
التطبيق عبارة عن برنامج مكتبي يمكن تثبيته على حاسوب عون خدمة العملاء أو على حاسوب تابع للمؤسسة. يسمح التطبيق بمعالجة المكالمات بطريقتين أساسيتين:

- معالجة مكالمة مباشرة: حيث يتم تسجيل أو استقبال الصوت أثناء المكالمة، ثم يقوم التطبيق بتحليلها وعرض النتائج للمستخدم.



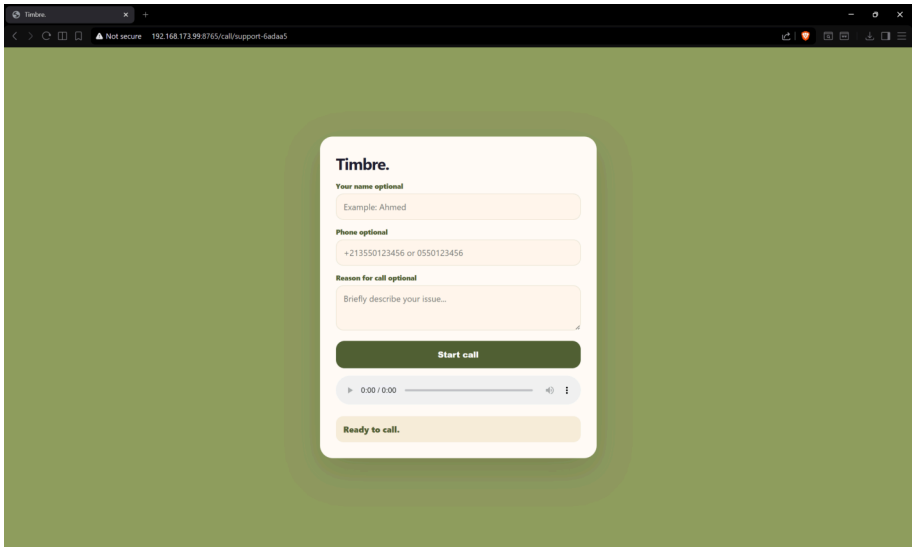
شكل 2: واجهة الاتصال المباشر أثناء استقبال وتحليل المكالمة

- معالجة مكالمة سابقة: حيث يمكن للمستخدم اختيار ملف صوتي مسجل سابقا، ثم يقوم التطبيق بتفريغه وتحليله وتلخيصه.



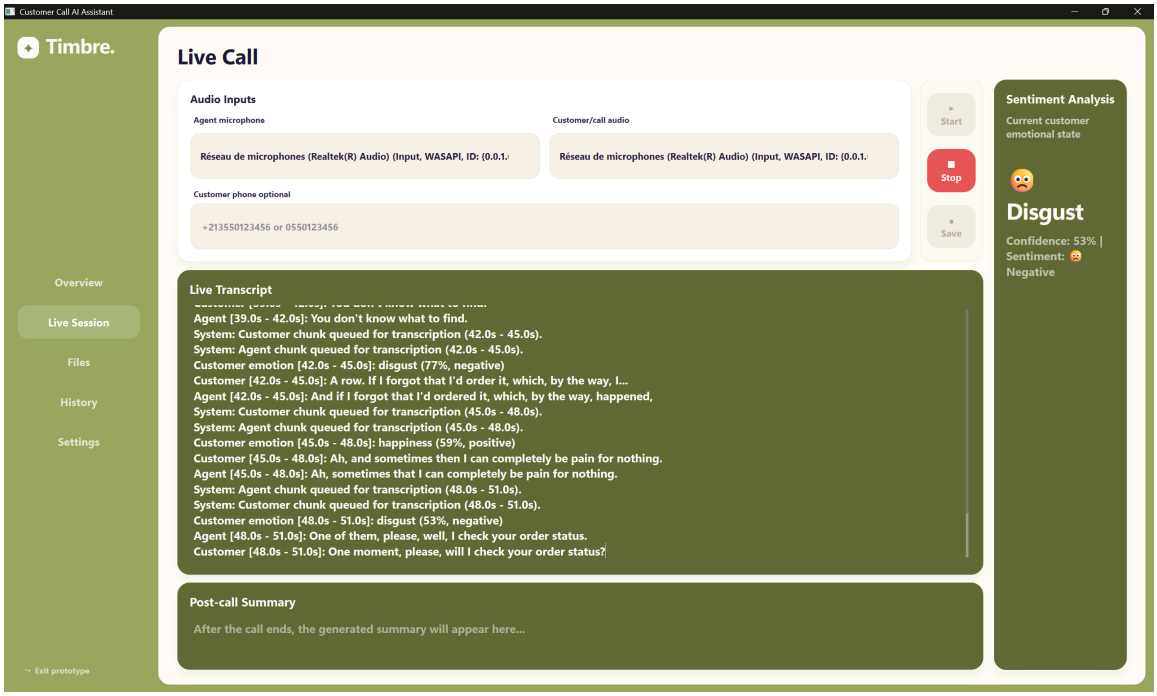
شكل 3: واجهة معالجة الملفات الصوتية المسجلة سابقا

كما يمكن للنموذج الأولي أن يدعم فكرة طلب الاتصال عبر المتصفح، حيث يستطيع العميل فتح رابط خاص بالمؤسسة، إدخال معلومات بسيطة مثل الاسم ورقم الهاتف وسبب الاتصال، ثم إرسال طلب مكالمة إلى عون خدمة العملاء داخل التطبيق.



شكل 4: واجهة العميل لطلب مكالمة دعم عبر المتصفح

وعند وصول طلب الاتصال، يظهر إشعار داخل لوحة التحكم حتى يتمكن عون خدمة العملاء من قبول المكالمة أو رفضها حسب الحالة.



شكل 5: إشعار وصول طلب مكالمة عبر الويب داخل لوحة التحكم

بعد معالجة المكالمة، يعرض التطبيق النتائج الأساسية التالية:

- النص الناتج عن التفريغ الآلي للمكالمة.
- الحالة الشعورية العامة للعميل، مثل إيجابية، سلبية، أو محايدة.
- ملخص مختصر يوضح أهم ما دار في المكالمة.
- إمكانية حفظ النتائج داخل قاعدة بيانات محلية أو قاعدة بيانات خاصة بالمؤسسة.
- إمكانية الرجوع إلى المكالمات السابقة والبحث داخلها.

3 مكونات النموذج الأولي

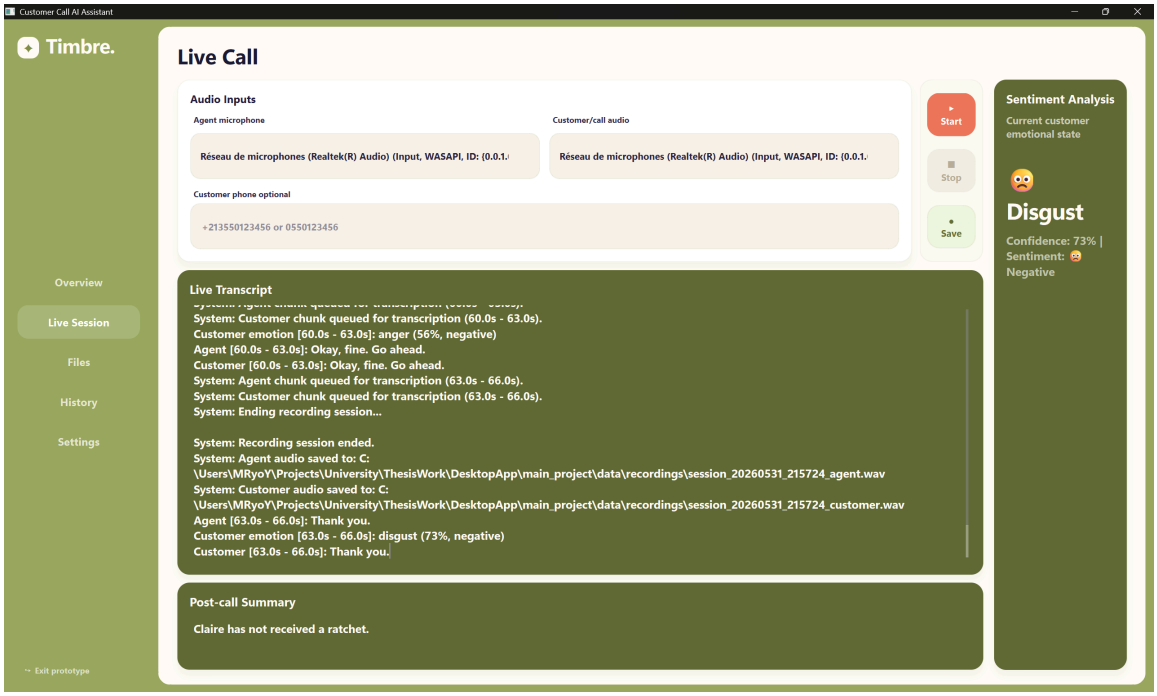
يتكون النموذج الأولي من عدة مكونات مترابطة، ويمكن عرضها كما يلي:

المكون	الدور داخل التطبيق
واجهة المستخدم	تمثل الجزء الذي يتعامل معه المستخدم مباشرة، وتسمح له ببدء معالجة المكالمة، اختيار ملف صوتي، عرض النص، عرض الحالة الشعورية، مشاهدة الملخص، وحفظ النتائج.

وحدة التقاط الصوت	تسمح بالحصول على الصوت من مكالمة مباشرة أو من ملف صوتي مسجل سابقا، ثم تمريره إلى وحدات المعالجة داخل التطبيق.
وحدة التفريغ الآلي	تقوم بتحويل الكلام المنطوق داخل المكالمة إلى نص مكتوب، حتى يصبح محتوى المكالمة قابلا للقراءة والحفظ والبحث.
وحدة تقدير الحالة الشعورية	تقوم بتحليل الإشارات الصوتية للمكالمة من أجل إعطاء مؤشر عام حول الحالة الشعورية للعميل، مثل حالة إيجابية، سلبية، أو محايدة.
وحدة التلخيص الآلي	تقوم بمعالجة النص الناتج عن التفريغ الآلي وتوليد ملخص مختصر يبرز أهم نقاط المكالمة.
قاعدة البيانات	تستعمل لحفظ نتائج المكالمات، مثل النص، الملخص، الحالة الشعورية، تاريخ المعالجة، وبعض المعلومات التي تساعد على البحث والرجوع إلى المكالمات السابقة.
واجهة البحث والأرشيف	تسمح للمستخدم أو المؤسسة بالوصول إلى المكالمات السابقة، والبحث داخل قاعدة البيانات حسب الكلمات المفتاحية أو محتوى المكالمة أو الحالة الشعورية.

4 طريقة عمل النموذج الأولي

يمر استعمال التطبيق بعدة خطوات متتابعة، تبدأ من إدخال الصوت وتنتهي بحفظ النتائج. ويمكن تلخيص طريقة العمل كما يلي:



شكل 6: واجهة الاتصال المباشر بعد انتهاء المعالجة وظهور الملخص والحالة الشعورية

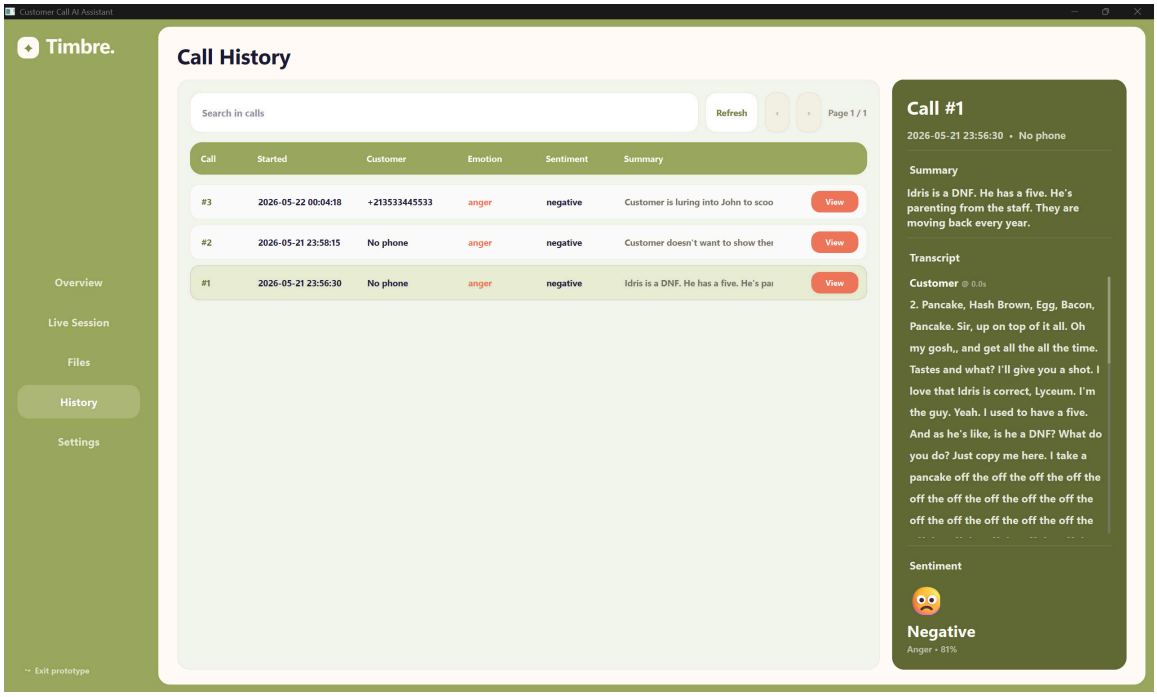
المرحلة	الوصف
إدخال المكالمة	يقوم المستخدم إما ببدء معالجة مكالمة مباشرة أو اختيار تسجيل صوتي سابق من جهاز الحاسوب.
معالجة الصوت	يقوم التطبيق بقراءة الصوت وتحضيره حتى يصبح قابلا للتمرير إلى وحدات الذكاء الاصطناعي.
تفريغ المكالمة	يتم تحويل الكلام الموجود في المكالمة إلى نص مكتوب، مما يسمح للمستخدم بقراءة محتوى المكالمة دون الرجوع إلى التسجيل الصوتي الكامل.
تقدير الحالة الشعورية	يقوم النظام بإعطاء مؤشر عام حول حالة العميل أثناء المكالمة، ويصنفها إلى إيجابية، سلبية، أو محايدة.
تلخيص المكالمة	بعد الحصول على النص، يقوم التطبيق بتوليد ملخص قصير يوضح موضوع المكالمة، المشكلة المطروحة، والإجراء أو النتيجة النهائية.
عرض النتائج	تعرض النتائج داخل واجهة التطبيق، بحيث يستطيع المستخدم مشاهدة النص الكامل، الملخص، والحالة الشعورية العامة.

الحفظ في قاعدة البيانات	يمكن حفظ النتائج داخل قاعدة بيانات محلية أو قاعدة بيانات خاصة بالمؤسسة، من أجل الرجوع إليها لاحقا.
البحث والمتابعة	يمكن للمستخدم البحث في المكالمات السابقة حسب الكلمات المفتاحية أو نوع الحالة الشعورية أو محتوى المكالمة.

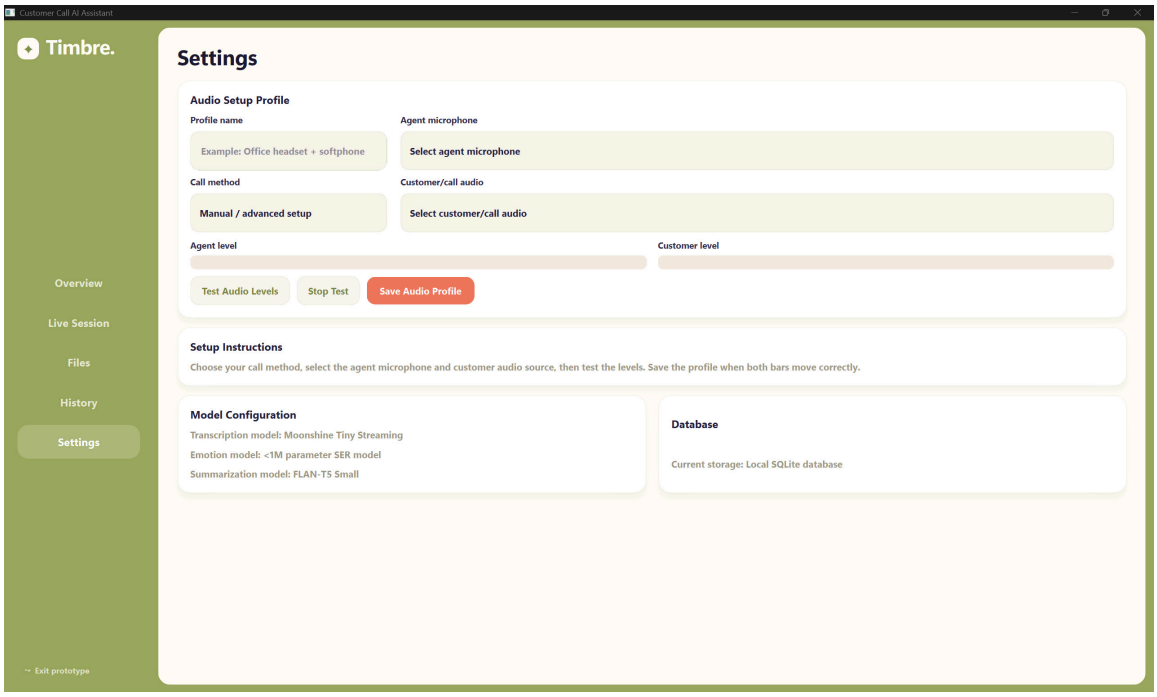
5 الواجهات الأساسية للتطبيق

يحتوي النموذج الأولي على مجموعة من الواجهات البسيطة التي تساعد المستخدم على استعمال التطبيق بسهولة. ويمكن عرض أهم الواجهات كما يلي:

الواجهة	الوصف
لوحة التحكم	تسمح للمستخدم بمتابعة المؤشرات العامة مثل عدد المكالمات، آخر حالة شعورية، وروابط الاتصال عبر الويب.
واجهة الاتصال المباشر	تعرض حالة المعالجة الحالية، وتظهر فيها النتائج الأساسية مثل النص الناتج عن التفريغ، الحالة الشعورية العامة، والملخص الناتج عن التلخيص الآلي.
واجهة معالجة الملفات	تسمح باختيار ملفات صوتية مسجلة سابقا، ثم تنفيذ التفريغ، تقدير الحالة الشعورية، والتلخيص الآلي.
واجهة المكالمات السابقة	تعرض قائمة بالمكالمات التي تمت معالجتها سابقا، مع إمكانية فتح تفاصيل كل مكالمة والاطلاع على النص والملخص والحالة الشعورية.
واجهة الإعدادات	تسمح بضبط مصادر الصوت، اختبار مستويات الصوت، ومعرفة نماذج الذكاء الاصطناعي وقاعدة البيانات المستعملة في التطبيق.
واجهة طلب الاتصال عبر المتصفح	تمكن العميل من إدخال اسمه ورقم هاتفه وسبب الاتصال، ثم إرسال طلب اتصال إلى عون خدمة العملاء.



شكل 7: واجهة سجل المكالمات السابقة وتفاصيل كل مكالمة



شكل 8: واجهة الإعدادات الخاصة بالصوت والنماذج وقاعدة البيانات

6 نتائج النموذج الأولي

يظهر النموذج الأولي أن فكرة المشروع قابلة للتطبيق من الناحية التقنية، لأنه يجمع بين عدة وظائف مهمة داخل تطبيق واحد. كما يوضح أن التطبيق يمكن أن يساعد المؤسسات على تحويل المكالمات من تسجيلات صوتية صعبة

الاستغلال إلى بيانات منظمة يمكن قراءتها، تلخيصها، تصنيفها، والبحث فيها.
ويمكن تلخيص النتائج التي يقدمها النموذج الأولي كما يلي:

- إمكانية تفريغ المكالمات الصوتية إلى نصوص مكتوبة.
- إمكانية إعطاء مؤشر عام حول الحالة الشعورية للعميل.
- إمكانية توليد ملخص مختصر للمكالمة.
- إمكانية حفظ نتائج المكالمات في قاعدة بيانات.
- إمكانية الرجوع إلى المكالمات السابقة والبحث داخلها.
- إمكانية معالجة مكالمات مباشرة أو تسجيلات صوتية سابقة.

هذه النتائج تجعل النموذج الأولي مناسبا للعرض أمام المؤسسات المحتملة، خاصة بهدف توضيح الفكرة، إثبات فائدتها، وجمع الملاحظات قبل تطوير نسخة تجارية أكثر استقرارا.

7 حدود النموذج الأولي

رغم أن النموذج الأولي يوضح الفكرة الأساسية للمشروع، إلا أنه لا يزال يحتاج إلى تحسينات قبل اعتماده كمنتج تجاري كامل. ومن أهم الحدود الحالية ما يلي:

- دقة التفريغ الآلي قد تتأثر بجودة التسجيل أو وجود ضجيج في الخلفية.
- تقدير الحالة الشعورية يبقى مؤشرا عاما وليس حكما نهائيا على رضا العميل.
- أداء التطبيق قد يختلف حسب قوة الحاسوب المستعمل.
- التطبيق يحتاج إلى تجارب ميدانية أكثر داخل مؤسسات حقيقية.
- النسخة المؤسساتية تحتاج إلى تحسينات إضافية في الأمن، الصلاحيات، والتكامل مع قواعد بيانات المؤسسة.
- دعم اللغات واللهجات المحلية يحتاج إلى تطوير تدريجي حسب احتياجات السوق.

الاعتراف بهذه الحدود مهم من أجل تقديم المشروع بطريقة واقعية، ويؤكد أن النموذج الحالي هو خطوة أولى نحو منتج قابل للتطوير والتحسين.

8 آفاق تطوير النموذج الأولي

يمكن تطوير النموذج الأولي مستقبلا ليصبح منتجا تجاريا أكثر اكتمالا من خلال إضافة مجموعة من الخصائص، من بينها:

- تحسين دقة التفريغ الآلي في وجود الضجيج أو اختلاف اللهجات.
- تحسين وحدة تقدير الحالة الشعورية وربطها بسياق المكالمة.
- إضافة لوحة تحكم خاصة بالمشرفين لمتابعة مؤشرات المكالمات.
- إضافة إحصائيات حول عدد المكالمات السلبية أو الإيجابية أو المحايدة.
- تصنيف المكالمات حسب نوع المشكلة أو موضوع الاتصال.
- توفير نظام صلاحيات للمستخدمين داخل المؤسسة.
- ربط التطبيق بأنظمة إدارة علاقات العملاء.
- تحسين نظام البحث داخل قاعدة البيانات.
- دعم لغات ولهجات إضافية حسب السوق المستهدف.
- توفير نسخة مخصصة للمؤسسات الكبيرة ونسخة مبسطة للمؤسسات الصغيرة.

9 خلاصة المحور

يمثل النموذج الأولي التجريبي خطوة أساسية في تحويل فكرة المشروع إلى منتج قابل للاستعمال. فهو يوضح كيف يمكن استعمال الذكاء الاصطناعي في معالجة مكالمات خدمة العملاء من خلال التفريغ الآلي، تقدير الحالة الشعورية، والتلخيص الآلي، مع حفظ النتائج في قاعدة بيانات قابلة للبحث.

كما يبين النموذج أن المشروع يمكن أن يقدم قيمة عملية للمؤسسات، من خلال تقليل الوقت اللازم لمراجعة المكالمات، تحسين استغلال بيانات العملاء، تقليل تكلفة التخزين، والحفاظ على سرية البيانات عبر المعالجة المحلية. وبالرغم من أن النسخة الحالية تحتاج إلى تطوير وتجارب ميدانية إضافية، إلا أنها تمثل أساسا قويا لبناء منتج تجاري قابل للتوسع مستقبلا.

الملحق : نموذج العمل التجاري

شرائح العملاء	العلاقات مع العملاء	القيم المقترحة	الأنشطة الرئيسية	الشراكات الرئيسية
- شركات الاتصالات - البنوك وشركات التأمين - مراكز الاتصال وخدمة العملاء - شركات التجارة الإلكترونية - مؤسسات الدعم التقني - المؤسسات الصغيرة التي تعتمد على الهاتف - مؤسسات تحتاج إلى أرشفة وتحليل المكالمات	- عروض تجريبية للمؤسسات - دعم تقني مباشر - تدريب المستخدمين - عقود صيانة وتحديث - تخصيص حسب احتياجات المؤسسة **القنوات** - التواصل المباشر مع المؤسسات - عروض توضيحية وتجريبية - موقع إلكتروني للمشروع - شبكات التواصل الاجتماعي - حاضنة الأعمال الجامعية	- تحويل المكالمات إلى نصوص مكتوبة - تقدير الحالة الشعورية العامة للعميل - تلخيص المكالمات تلقائيا - تسهيل البحث في المكالمات السابقة - تقليل وقت مراجعة التسجيلات - تقليل تكلفة التخزين - حماية خصوصية البيانات - المعالجة محليا داخل المؤسسة - استغلال أجهزة المؤسسة الموجودة	- تطوير التطبيق المكتبي - دمج نماذج الذكاء الاصطناعي - تقدير الحالة الشعورية - التلخيص الآلي - حفظ النتائج والبحث فيها - الصيانة والدعم التقني **الموارد الرئيسية** - صاحب المشروع / المطور الرئيسي - نماذج الذكاء الاصطناعي - قاعدة بيانات محلية - حواسيب أعوان خدمة العملاء - بيانات صوتية للتجربة - خبرة تقنية في الصوت والنصوص	- مؤسسات تملك خدمة عملاء أو مراكز اتصال - شركات الاتصالات والبنوك - مزودو خدمات الإنترنت والعتاد المعلوماتي - حاضنة الأعمال الجامعية - مختصون في خدمة العملاء - مختصون في أمن المعلومات - مؤسسات صغيرة لتجربة النسخة الأولية

مصادر الإيرادات	هيكل التكاليف
- عقود مخصصة مع المؤسسات الكبيرة - رخص استعمال قصيرة المدى للمؤسسات الصغيرة - اشتراكات شهرية أو سنوية - خدمات التثبيت والتخصيص والصيانة - دعم تقني مدفوع حسب العقد - نسخة مجانية محدودة الخصائص للتجربة	- تطوير البرمجيات والواجهة - اختبار نماذج الذكاء الاصطناعي - معدات بسيطة للاختبار الصوتي - تجهيز العروض التجريبية - حماية البيانات وتحسين الأمن - التسويق والإشهار الأولي - الدعم التقني والصيانة - الإنترنت وأدوات العمل - التنقل والتواصل مع المؤسسات