



People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
IBN KHALDOUN UNIVERSITY OF TIARET

Dissertation

Presented to:

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

in order to obtain the degree of : MASTER

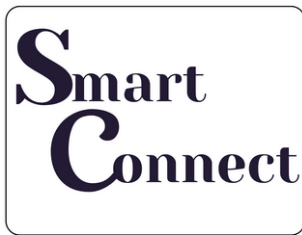
Specialty: software engineering

With a view to creating a startup

Presented by:

Hatem Fatima Zohra

On the theme:



Development of an Emotional and Contextual Chatbot to Enhance Communication Between Public Institutions and Citizens

Defended publicly on 15/06 /2025 in Tiaret in front the jury composed of :

Mr Talbi Omar	MCA	UIK	President
Mr Bekki Khadir	MAA	UIK	Supervisor
Mr Ouared Abdelkader	MCA	UIK	Co-Supervisor
Mr Boudaa Boujemaa	MCA	UIK	Examiner
Mr Daoud Mohamed Amine	MCB	UIK	Incubator Representative
Mr Bendjeloul Mokhtar	Professor	UIK	Economic Partner Representative

2024-2025

Acknowledgements

Praise be to God, the Almighty who granted me the strength, patience, and clarity of mind to complete this research. Without His grace and guidance, this work would not have been completed.

I extend my sincere thanks and gratitude to my esteemed supervisor, Professor Bekki Khathir, who contributed to the success of this work with his advice and guidance.

I also express my heartfelt thanks to the members of the jury for accepting to evaluate our work.

Dedication

*To my father, my greatest support and the best person in my
life.*

*Thank you for always believing in me, for your endless
encouragement,
and for being my strength in every step of this journey.*

*To my dear mother, your love, prayers, and patience have
carried me through.*

Your presence means everything to me.

*To my brothers and sisters, thank you for your constant
support, love,
and for always being there.*

*To my teachers, your guidance, knowledge, and
encouragement
have been a light throughout this academic path.*

*And to everyone who believed in me, cheered me on,
or simply reminded me to keep going thank you.*

Abstract

This thesis presents the design of an emotional and context-aware chatbot to improve communication between public institutions and citizens. Leveraging natural language processing and large language models, the chatbot can understand user queries, detect emotional cues, and generate empathetic responses tailored to the context. By offering more human-like and sensitive interactions, the system enhances user experience, promotes trust, and supports more effective public service delivery across various sectors.

Keywords:

Chatbot, Natural Language Processing (NLP), Large Language Models (LLMs), Emotion Detection, Context-Aware Systems, Public Services, Human-Computer Interaction, Empathetic Communication, Citizen Engagement.

Résumé

Ce mémoire présente la conception d'un chatbot émotionnel et contextuel visant à améliorer la communication entre les institutions publiques et les citoyens. Grâce au traitement automatique du langage naturel et à l'utilisation de modèles de langage avancés, le chatbot est capable de comprendre les requêtes des utilisateurs, de détecter leurs émotions et de fournir des réponses empathiques adaptées au contexte. En offrant des interactions plus humaines et sensibles, ce système renforce la confiance, améliore l'expérience utilisateur et soutient une meilleure qualité de service public dans divers secteurs.

Mots-clés :

Chatbot, Traitement Automatique du Langage Naturel (TALN), Modèles de Langage de Grande Taille, Détection des Émotions, Systèmes Contextuels, Services Publics, Interaction Homme-Machine, Communication Empathique, Engagement Citoyen.

Contents

Acknowledgements	3
Dedication	4
Abstract	5
Glossary	12
1 Designing Human-Centered AI Systems : The Role of NLP and LLMs	15
1.1 Introduction	16
1.2 Natural Language Processing	16
1.2.1 NLP techniques	16
1.2.2 Applications of NLP	17
1.2.3 NLP challenges	18
1.3 Understanding Large Language Models (LLMs)	18
1.3.1 Open-source vs. Closed-source Models	18
1.3.2 Large Language Model Families	19
1.3.3 LLM Open sources framework	19
1.3.4 LLM Pipeline: From Pre-training to Inference	20
1.3.5 Advanced Techniques: RAG and Prompt Engineering	22
1.4 LLM Capabilities	24
1.4.1 Basic Capabilities	24
1.4.2 Emerging Capabilities	25
1.4.3 Augmented Capabilities	25
1.5 Considering Human Aspects in Software Design	25
1.5.1 Human aspects	25
1.5.2 Human-Centered Approaches in Software Engineering	29
1.6 Context and Work Positioning	30
1.6.1 Toward Human-Centered Digital Interactions in Algeria	30
1.6.2 Challenges	30
1.6.3 Our Missions	31
1.6.4 Planned Timeline	31
1.7 Conclusion	31
2 Design of Our Solution	32
2.1 Introduction	33
2.2 Motivation and Challenges	33
2.2.1 Motivation	33
2.2.2 Emotion and context awareness	36
2.2.3 Challenges	37
2.3 Our Framework	37
2.3.1 Possible Scenarios for Empathetic System	38

2.4	Solution Overview	42
2.4.1	Why a Hybrid Approach?	42
2.4.2	Formalization	43
2.4.3	Architecture Modules	44
2.4.4	System Architecture	46
2.5	UX Design of our solution	47
2.5.1	Push Notifications and Timely Alerts	48
2.5.2	Chat History Display and Session Persistence	48
2.5.3	Emotion Selection Menu and Emotional State Detection	49
2.5.4	Intent Confirmation and Clear Messaging	49
2.5.5	Personalization and Adaptive Language Style	49
2.5.6	Progressive Disclosure and Guided Conversations	49
2.5.7	Quick Replies, Menu Options, and FAQ Integration	49
2.5.8	Sentiment Analysis and Emotional Intelligence Integration	49
2.6	Instantiation of our framework	49
2.6.1	First step: Need, Intent, and Emotion	49
2.6.2	Second step: Response strategy and generation	50
2.6.3	Third step: Delivery, Feedback, and Learning	51
2.7	Conclusion	51
3	SmartConnect Chatbot : Implementation	52
3.1	Introduction	53
3.2	Technologies and Tools Used	53
3.2.1	Programming Languages	53
3.2.2	Frameworks and Libraries	53
3.2.3	Dataset Handling	53
3.2.4	Model Architectures	53
3.2.5	Justification of Technology Choices	54
3.2.6	Development Environment	54
3.3	System Architectures	54
3.3.1	RULE BASED MODEL	54
3.3.2	System Architecture(AI MODEL)	56
3.4	Dataset Preparation	57
3.4.1	Rule Based model	57
3.4.2	LLM-Assisted Empathetic Data Creation	58
3.5	Model Implementation	59
3.5.1	Rule based model	59
3.6	Model Fine-Tuning	61
3.6.1	Data Loading and Preprocessing	61
3.6.2	Tokenizer and Model Initialization	61
3.6.3	Tokenization of Dataset	61
3.6.4	Training Setup	62
3.6.5	Fine-Tuning Procedure	62
3.6.6	Model Saving	62
3.7	UI/UX Design	63
3.7.1	Objective of UI/UX	63
3.7.2	Tools Used	63
3.7.3	Wireframes	63
3.7.4	Admin Dashboard (UI)	64
3.7.5	User Experience	64
3.8	Challenges and Solutions	64
3.8.1	Model Size and Computation Cost	64

3.8.2 Prompt Design for Empathy	64
3.8.3 Balancing Response Accuracy and Warmth	65
3.9 Test and Evaluation	65
3.10 Conclusion	66
General Conclusion	67
Annexes	69
Bibliography	80

List of Figures

1.1	NLP pipeline	16
1.2	Tokenization	17
1.3	Lemmatization	17
1.4	LLM Families	19
1.5	Decision Tree for Selecting an Open-Source Large Language Model Based on User Preferences, Machine Constraints, and Task	20
1.6	Overview of Fine Tuning Techniques including Instruction Fine Tuning and Parameter Efficient Fine Tuning (PEFT) approaches.	22
1.7	Core pipeline of a Large Language Model (LLM)	22
1.8	Combinations of Retrieval and Generation in RAG-based Architectures	23
1.9	Improved structure of prompt engineering components for large language models	23
1.10	LLM Capabilities	24
1.11	Taxonomy of Human Aspects Relevant to Software Engineering	25
1.12	Visual Protocol of the Think-Aloud Method in Software Engineering/User Studies	30
2.1	Semantic relationships connecting empathy with emotion, context, and intent.	36
2.2	General vision	38
2.3	Example of Input Prompt with Emotional and Contextual Guidance	39
2.4	Example of Output Response Guided by Prompt Engineering	39
2.5	Step-by-step Evolution of the Empathetic Chatbot Solution	39
2.6	Rule-based model	40
2.7	Alternative Styled Rule-Based Chatbot Interaction Schema	41
2.8	Emotion and Intent Detection with Prompted LLM	41
2.9	Pipeline of an Empathy-Fine-Tuned LLM from User Input to Output	42
2.10	Compact vertical flow of empathetic response generation pipeline.	45
2.11	Response transition scenarios: from automated response to human intervention	46
2.12	Hybrid model	47
2.13	Organigram of the chatbot interaction pipeline	47
3.1	System Architecture Diagram of the Empathetic Chatbot	55
3.2	AI Model Data Flow in Development Environment	57
3.3	dataset building	57
3.4	Intermediate solution between customers and stakeholders	63
3.5	Wirframe of our application	63
3.6	Admin dashbord	64

List of Tables

1.1	Comparison of selected open-source LLMs by task, machine constraints, and platform	20
1.2	Planned schedule for the master’s project	31
2.1	Example telecom responses and extracted empathy keywords	37
2.2	UX Design Strategies and Components for a Dedicated Client Chatbot	48
2.3	Chatbot interaction pipeline (Part 1/3): Need, Intent, Emotion, Motivation . . .	50
2.4	Chatbot interaction pipeline (Part 2/3): Response strategy and generation . . .	50
2.5	Chatbot interaction pipeline (Part 3/3): Delivery, Feedback, and Learning . . .	51
3.1	Sample user queries and their associated intents from the dataset.	58
3.2	Test Questions with Expected and Predicted Intents	65
3.3	Empathy Evaluation of Responses	66

Glossary

- **AI** : Artificial Intelligence
- **UI** : User Interface
- **UX** : User eXperience
- **ML** : Machine Learning
- **LLM** : Large Language Models
- **NLP** : Natural Language Processing
- **RAG** : Retrieval Augmented Generation
- **JSON** : *JavaScript Object Notation*
- **GPT** : *Generative Pre-trained Transformer*
- **BERT** : *Bidirectional Encoder Representations from Transformers*
- **EI** : Emotional Intelligence

General Introduction

Introduction

Today, companies and organizations face strong competition. To succeed, they must respond quickly and clearly to clients' questions and problems. This helps keep clients happy and improves company performance [1]. The idea of using business thinking is also growing in universities and other service areas [2]. As a result, many organizations now see their clients as customers who need continuous support and attention [3], [4], [5].

Problem Statement

Companies often deal with many clients asking different questions every day [6], [7]. Employees must handle these requests quickly and correctly. In large companies, like those in telecom, this task becomes very difficult and time-consuming. Providing fast and clear answers becomes a challenge, especially when many questions are repeated.

Chatbots can help solve this problem. These tools use natural language technology to talk with users and answer their questions automatically [8]. For example, new clients often ask: "What are the current offers?", "How do I activate my service?", or "Who can help me with billing?". In financial services, questions like "Am I eligible for a discount?" or "How do I pay my bill?" are very common.

But giving the right answer depends on many things: who the client is, their personal details, their service history, and their current situation. If this context is not used, the chatbot may give the wrong response [9]. Understanding the client's intent requires combining all this information in a smart way.

This thesis answers the following research question:

How can we use the client profile, their context, and their questions together to detect their real needs and give better answers?

Research Contribution

To solve this problem, this thesis proposes a framework for a context-aware chatbot that uses rules to help company staff, especially in telecom settings. The chatbot combines client profile, context, and the question being asked. This allows the system to better understand what the client wants.

The chatbot gives a list of good answers, based on the client's situation. It works all day, every day (24/7), helping both clients and employees. A dashboard is also included, so staff can track common questions and client needs in real time.

This system is tested in a telecom company in Algeria, where workers lose a lot of time answering repeated questions from clients. With this chatbot, time is saved, support is improved, and client satisfaction goes up.

Chapter 1

Designing Human-Centered AI Systems : The Role of NLP and LLMs

1.1 Introduction

Natural language processing (NLP) and large language models (LLMs) stand as major milestones in the evolution of artificial intelligence, empowering systems to understand, interpret, and generate human language with remarkable accuracy and nuance. As these technologies continue to advance, it becomes increasingly essential to embed human-centered principles into their design ensuring they serve not only technical goals but also the *emotional*, *cognitive*, and *behavioral needs* of users.

By merging computational power with a deep understanding of human factors, we pave the way for developing intelligent, empathetic systems capable of delivering richer, more meaningful digital experiences. This intersection of technology and human insight opens new possibilities for creating interactions that are not only functional but also engaging, adaptive, and deeply human-centered.

1.2 Natural Language Processing

NLP is a branch of AI that tries to process and analyze natural language data. It involves developing algorithms and models that enable computers to *understand*, *interpret*, and *generate* human language.

Natural Language Processing is a field of computer science that combines machine learning and computational linguistics to facilitate effective and seamless communication between humans and computers. It focuses on enabling machines to comprehend the structure and meaning of human language, interpret it, and produce relevant responses.

The primary goal of NLP is to develop computer systems capable of performing useful tasks using natural human-readable language [10].

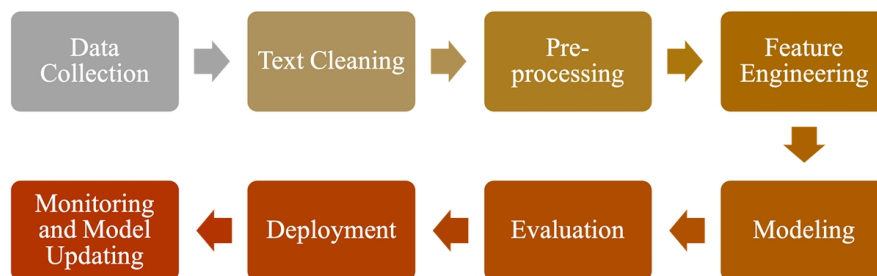


Figure 1.1: NLP pipeline

1.2.1 NLP techniques

In this section, we present the pipeline of a natural language processing (NLP) system.

Tokenization :

Tokenization is a crucial preprocessing step in NLP. It is the process of parsing a text into tokens, which can be words, numbers or punctuation marks [11]. Figure 1.3 shows Splitting a sentence into individual words for NLP processing.

Lemmatization :

Lemmatization is the process of reducing words to their base forms. A Lemma is the base form of a token.[11] (See Figure 1.3)

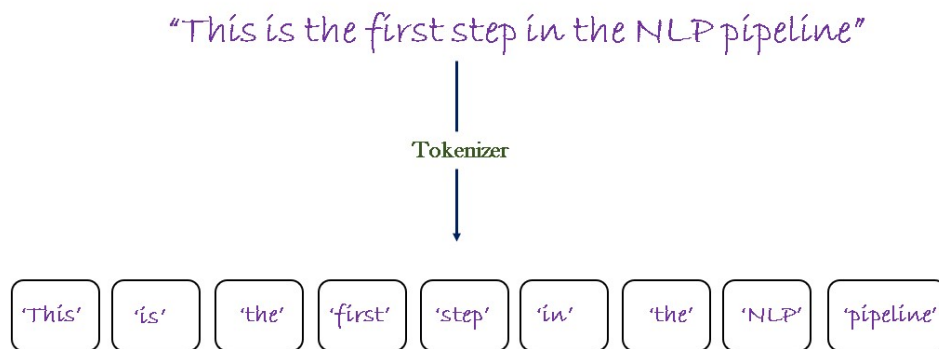


Figure 1.2: Tokenization

Lemmatization

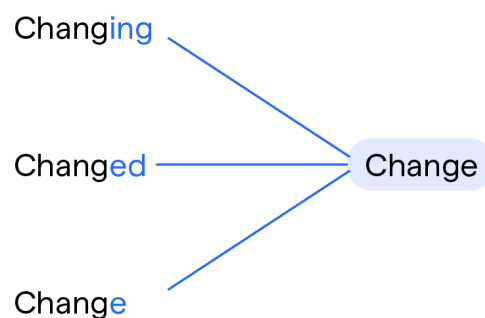


Figure 1.3: Lemmatization

Named Entity Recognition :

Named Entity Recognition is a crucial task in Natural Language Processing. It involves identifying and classifying words or phrases that refer to specific entities within a text. A named entity represents a real-world object that can be referenced by a proper name. It can be a person, organization, location or other entities.[12]

1.2.2 Applications of NLP

NLP plays a vital role in transforming how we interact with machines and information. Some key applications of NLP[10] include:

- Chatbots
- Machine Translation
- Text Categorization

- Spam Filtering
- Information Extraction
- Summarization
- Dialogue System

1.2.3 NLP challenges

Natural Language Processing (NLP) faces several complex challenges due to the complexity and diversity of human language:

- **Linguistic Complexity:** Natural languages are characterized by intricate structures, subtle nuances, and diverse interpretations. Effectively understanding and processing these complexities presents a significant challenge in NLP.
- **Dialectal Variations:** Language differences across regions and social groups lead to various dialects, which NLP models must account for to facilitate effective communication.
- **Ambiguity and Misinterpretation:** Ambiguous words, phrases, and meanings that depend on context can result in misinterpretations in NLP systems, necessitating a deeper, context-aware understanding.
- **Considering Human Aspects:** Capturing emotional tone, user intent, and socio-cultural context remains a key challenge for making NLP systems more empathetic and user-aligned.

1.3 Understanding Large Language Models (LLMs)

Large Language Models (LLMs) are advanced AI systems designed to understand and generate human language. Trained on massive amounts of textual data, these models utilize deep learning techniques, particularly transformer architecture, enabling them to recognize patterns and structures of language. These models can perform a wide range of tasks, including machine translation, text generation, and summarization. Fine-tuning LLMs on specific tasks enhances their performance. As a result, LLMs have become essential in numerous artificial intelligence applications, such as virtual assistants and chatbots.[\[13\]](#)

1.3.1 Open-source vs. Closed-source Models

Large Language Models (LLMs) can be broadly categorized into open-source models, such as LLaMA and Falcon, and closed-source models, like GPT-4 and Claude. Open-source models provide transparency, flexibility, and allow researchers and developers to fine-tune and adapt the models to specific domains or languages. Closed-source models, on the other hand, are typically backed by extensive resources, large proprietary datasets, and robust commercial support, often achieving state-of-the-art results but without offering access to their underlying architectures or training data.

Recent evaluations, such as the one conducted in the *Natural Language Processing Journal* (Volume 10, March 2025), have compared the performance of open and closed-source LLMs in low-resource languages using zero-shot, few-shot, and chain-of-thought prompting techniques. The findings highlight that while closed-source models often outperform in general benchmarks, open-source models demonstrate competitive results when carefully fine-tuned or enhanced through advanced prompting strategies. This indicates that, despite resource constraints, open-source models hold significant potential for innovation and democratization in language technologies, especially in regions and languages underrepresented in mainstream AI research [\[14\]](#).

1.3.2 Large Language Model Families

Large language models refer to transformer-based neural language models that consist of tens to hundreds of billions of parameters. which are pre-trained on massive text data. These models are divided into three main families: GPT, Llama, and Palm.[13]

1. **The GPT Family:** Generative Pre-trained Transformers (GPT) are decoder-only, Transformer-based language models developed by OpenAI. This family consists of GPT-1, GPT-2, GPT-3, Instruct GPT, Chat GPT, GPT-4, CODEX, and Web GPT. Although earlier models like GPT-1 and GPT-2 are open-source, the more recent versions, including GPT-3 and GPT-4, are closed-source and available only through APIs.
2. **The Llama Family:** LLaMA is a collection of foundation language models released by Meta. These open-source models are growing rapidly in popularity, as many research groups use them to develop enhanced open-source LLMs and create task-specific LLMs for mission-critical applications.
3. **The Palm Family :** PaLM (Pathways Language Model) is a family of large language models developed by Google, designed with a transformer-based architecture that optimizes scale and efficiency. The first model, announced in April 2022, features 540 billion parameters and is pre-trained on a vast corpus of 780 billion tokens. Utilizing 6,144 TPU v4 chips, PaLM achieves state-of-the-art results in few-shot learning and multi-step reasoning tasks. Its successor, PaLM-2, offers improved multilingual capabilities and efficiency. Additionally, Med-PaLM is a domain-specific model fine-tuned for medical inquiries, achieving notable performance in healthcare tasks. Overall, these models demonstrate significant advancements in language understanding and generation.

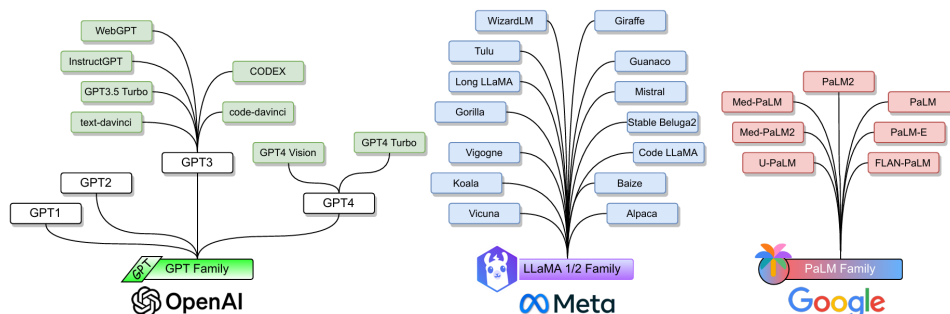


Figure 1.4: LLM Families

1.3.3 LLM Open sources framework

We conducted a survey on LLM ranking sites and recent strategies concerning 555 LLMs specialized in various domains, including the most used, the best performing, and those adapted to specific tasks. This analysis allowed us to identify key trends and develop a decision support model to help developers choose the most suitable LLM for their needs.

By analyzing the Hugging Face community¹ and recommendations from LLM AI forums, we have elaborated a decision tree that guides developers to choose the open-source LLM according to machine constraints, training requirements, user preferences, and addressed tasks.

¹<https://huggingface.co>

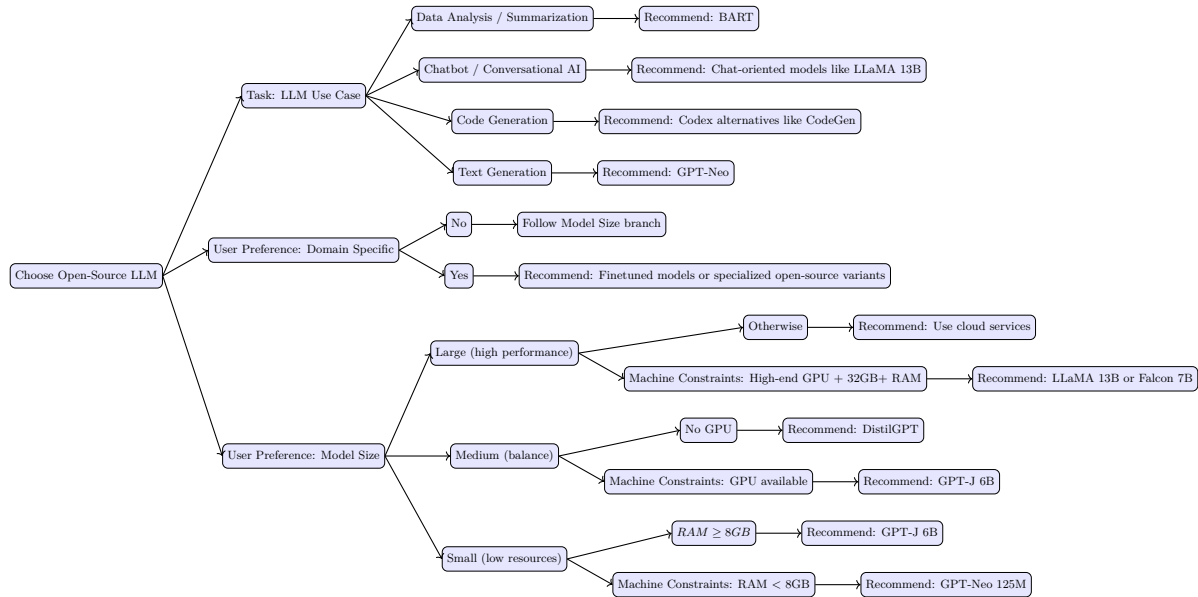


Figure 1.5: Decision Tree for Selecting an Open-Source Large Language Model Based on User Preferences, Machine Constraints, and Task

Table 1.1: Comparison of selected open-source LLMs by task, machine constraints, and platform

LLM	Suitable Tasks	Machine Constraints	Platform	Notes
GPT-Neo 125M	Text gen., NLP	Low RAM (< 8GB), CPU-friendly	Hugging Face	Lightweight, easy to deploy
GPT-J 6B	Text gen., code	Medium RAM (≥ 16GB), GPU recommended	Hugging Face	Good balance between size and performance
LLaMA 7B	Chatbot, conversational AI	High RAM (≥ 32GB), GPU required	Meta AI release	State-of-the-art open weights
Falcon 7B	Text gen., summarization	High RAM (≥ 32GB), GPU required	Hugging Face	Fast and efficient inference
CodeGen	Code generation	Medium RAM (≥ 16GB), GPU recommended	Hugging Face	Specialized for coding tasks

1.3.4 LLM Pipeline: From Pre-training to Inference

LLM pipeline typically begins with data preprocessing, which includes tokenization, normalization, and cleaning to prepare textual inputs [15], [16]. This is followed by model training phases, where large language models are pretrained on massive datasets to learn language patterns [17], [18]. After pretraining, models may be fine-tuned on specific downstream tasks using labeled data to improve task-specific performance [19]. Finally, the inference step involves using the trained model to generate predictions or responses in real applications [20].

Pre-training Phase

In the pre-training phase, the LLM is exposed to extremely large corpora of unannotated text, often sourced from the web, books, or other large-scale text datasets. The goal is to learn statistical patterns, grammar, syntax, semantics, and broad world knowledge through self-supervised learning tasks. Common pre-training objectives include masked language modeling (as used in BERT) or autoregressive language modeling (as used in GPT) [17], [18].

Tokenization plays a critical role in preparing text for large language models. One commonly used method is:

BPE (Byte Pair Encoding)

Algorithm based on the frequency of character pairs.

Compresses the vocabulary by merging the most frequent pairs of characters or subwords.

Used by GPT, BERT, and other Transformer models.

Fine-tuning Phase

Fine-tuning is the process of adapting a large pre-trained language model, such as LLaMA, to a specific task by continuing its training on a smaller, task-relevant dataset. This process slightly adjusts the model's parameters, enabling it to improve performance on new objectives while retaining the general knowledge acquired during pre-training. Fine-tuning enhances task-specific performance efficiently, without the need to train the model from scratch, making it a practical and widely used approach for applying large language models to various NLP tasks.[21].

The fine-tuning process (Figure 1) iteratively adjusts a pretrained language model on labeled data to adapt it to a specific task or domain.

Algorithm 1 Simplified Fine-Tuning of a Large Language Model

Require: Pretrained model $\mathcal{M}_0$, dataset $\mathcal{D}$, learning rate η , epochs E

Ensure: Fine-tuned model $\mathcal{M}_{\text{ft}}$

```

1: for  $e \leftarrow 1$  to  $E$  do
2:   for all batch  $(x, y) \in \mathcal{D}$  do
3:      $\hat{y} \leftarrow \mathcal{M}_0(x)$  ▷ Forward pass
4:     Compute loss  $\mathcal{L}(y, \hat{y})$ 
5:     Update model:  $\theta \leftarrow \theta - \eta \nabla_{\theta} \mathcal{L}$ 
6:   end for
7: end for
8: return  $\mathcal{M}_{\text{ft}}$ 
```

Fine Tuning Techniques

Fine-tuning adapts large language models (LLMs) to specific tasks or domains, improving efficiency and performance. It includes **Instruction Fine-Tuning** for task alignment via instructions, and **Parameter-Efficient Fine-Tuning (PEFT)** to minimize trainable parameters while preserving effectiveness.

- **Instruction Fine Tuning**
 - **Single Task:** Tuning for a single, specific task.
 - **Multitask:** Tuning the model simultaneously on multiple tasks.
 - **Selective:** Tuning only a subset of the language model.
- **Parameter Efficient Fine Tuning (PEFT)**
 - **Reparameterization:** Reducing the number of trainable parameters using methods such as LoRA and QLoRA.
 - **Additive Approaches**
 - * **Adapters:** Small modules inserted within the model layers to adjust behavior without changing the base weights.
 - * **Soft Prompts:** Learnable embeddings prepended to input sequences to guide model behavior.

Figure 1.6 presents a taxonomy of fine-tuning techniques for large language models. It categorizes approaches into Instruction, Fine Tuning and Parameter Efficient Fine Tuning (PEFT), each with distinct strategies.

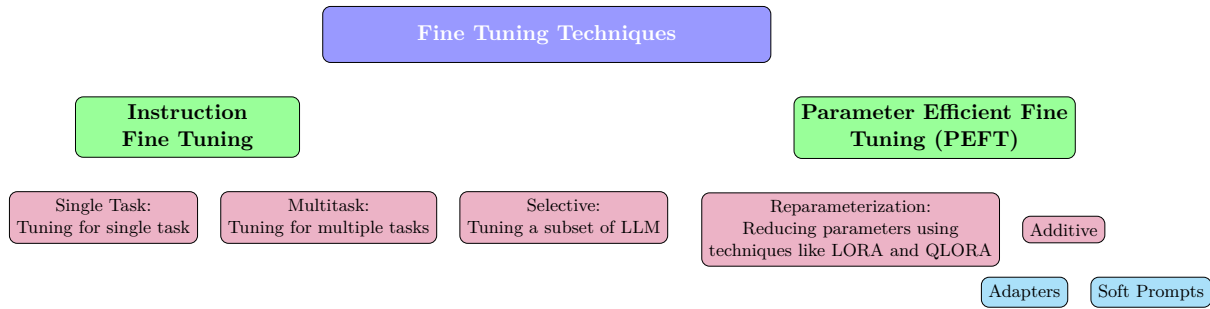


Figure 1.6: Overview of Fine Tuning Techniques including Instruction Fine Tuning and Parameter Efficient Fine Tuning (PEFT) approaches.

Inference Phase

Inference refers to the deployment and use of the trained model to generate outputs in real-world scenarios. During inference, the LLM takes user inputs (prompts) and generates predictions or responses without further parameter updates. Efficiency during inference is crucial, especially for large-scale applications, due to the high computational and memory demands of large models. Techniques such as quantization, pruning, and model distillation are often applied to reduce latency and resource consumption [21].

Deployment and Monitoring

Once the LLM is ready for production, it is integrated into applications via APIs or embedded systems. Continuous monitoring is essential to ensure stable performance, fairness, and safety. Issues such as model drift, unintended biases, or hallucinated outputs must be detected and addressed. Updating models periodically with new data or through continued fine-tuning helps maintain relevance and quality over time [13].

Figure 1.7 illustrates the core pipeline of a Large Language Model (LLM), starting from user input to output generation through tokenization, embedding, and transformer layers.

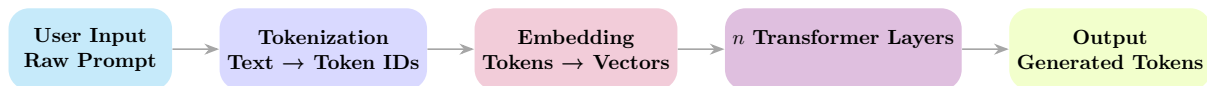


Figure 1.7: Core pipeline of a Large Language Model (LLM)

1.3.5 Advanced Techniques: RAG and Prompt Engineering

This section presents RAG and prompt engineering as strategies to improve the relevance and accuracy of LLM outputs.

Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) enhances Large Language Models (LLMs) by integrating external knowledge sources during inference. This approach addresses limitations such as hallucinations and outdated information inherent in standalone LLMs. Gao et al. (2023) categorize RAG into three paradigms: Naive RAG, which retrieves and appends documents to prompts; Advanced RAG, which incorporates sophisticated retrieval mechanisms; and Modular RAG, which decouples retrieval and generation components for flexibility and scalability [22].

Figure 1.8 illustrates key RAG scenarios, including retrieval before generation, after generation, and in iterative loops. These strategies improve the contextual relevance and factual grounding of generated outputs.

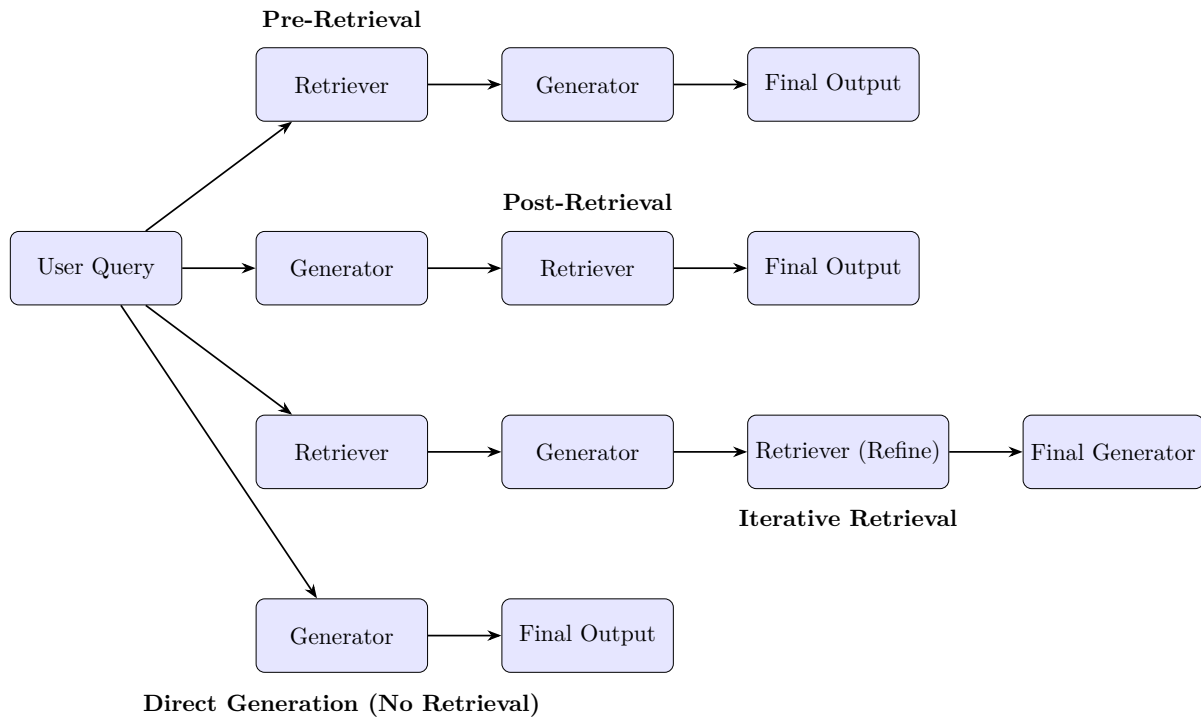


Figure 1.8: Combinations of Retrieval and Generation in RAG-based Architectures

Prompt Engineering

Prompt Engineering involves crafting input prompts to guide LLMs toward desired outputs without modifying model parameters. Sahoo et al. (2024) provide a systematic survey of prompt engineering techniques, highlighting methods like zero-shot, few-shot, and chain-of-thought prompting. These techniques enable LLMs to perform diverse tasks by leveraging contextual cues within prompts [23]. Additionally, Debnath et al. (2024) emphasize the role of prompt engineering in optimizing LLM performance across various applications, noting its significance in enhancing coherence, accuracy, and task alignment [24].

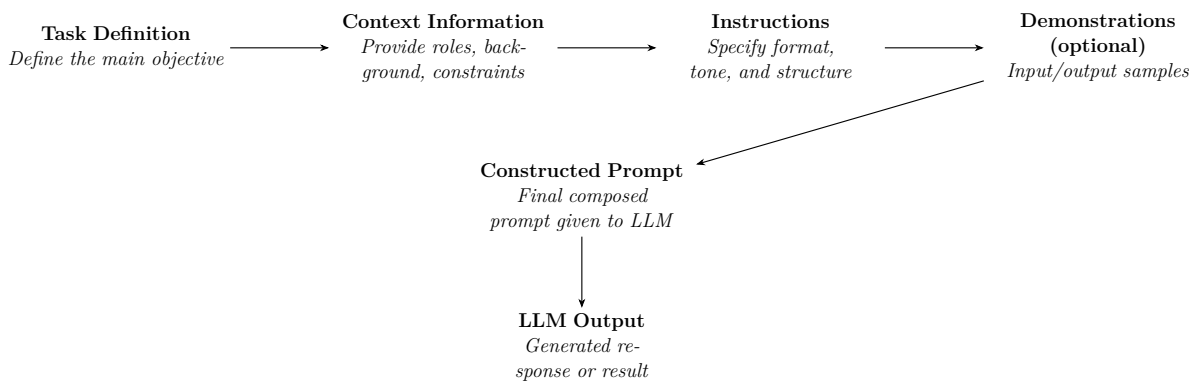


Figure 1.9: Improved structure of prompt engineering components for large language models

For example, enhancing *context* and *instructions* allows prompts to reflect emotions and user roles, making LLM responses more empathetic and user-centered.

Task Definition: Provide emotional support and practical advice to users expressing frustration with a software tool.

Context Information: The user is a non-technical adult who is feeling overwhelmed and frustrated by repeated software errors. They need clear, empathetic, and encouraging guidance.

Instructions: Use a warm and understanding tone. Avoid technical jargon. Break solutions into small, manageable steps. Acknowledge the user’s feelings and provide reassurance.

Demonstration (optional):

- *User:* “I keep getting this stupid error and I’m about to give up!”
- *Assistant:* “I completely understand how frustrating that must be. Let’s take it step by step, I’m here to help. Can you tell me exactly what the error message says?”

Constructed Prompt: You are a helpful and empathetic assistant. A user is experiencing frustration with a software error. Respond with kindness, avoid technical language, and guide them step-by-step. First, ask for the specific error message and reassure them that you’re here to help.

1.4 LLM Capabilities

Large Language Models (LLMs) have a wide range of abilities. These capabilities can be grouped into three main categories: **Basic**, **Emerging**, and **Augmented**. Each category supports different tasks depending on the model’s training and context. As we can see in 1.10

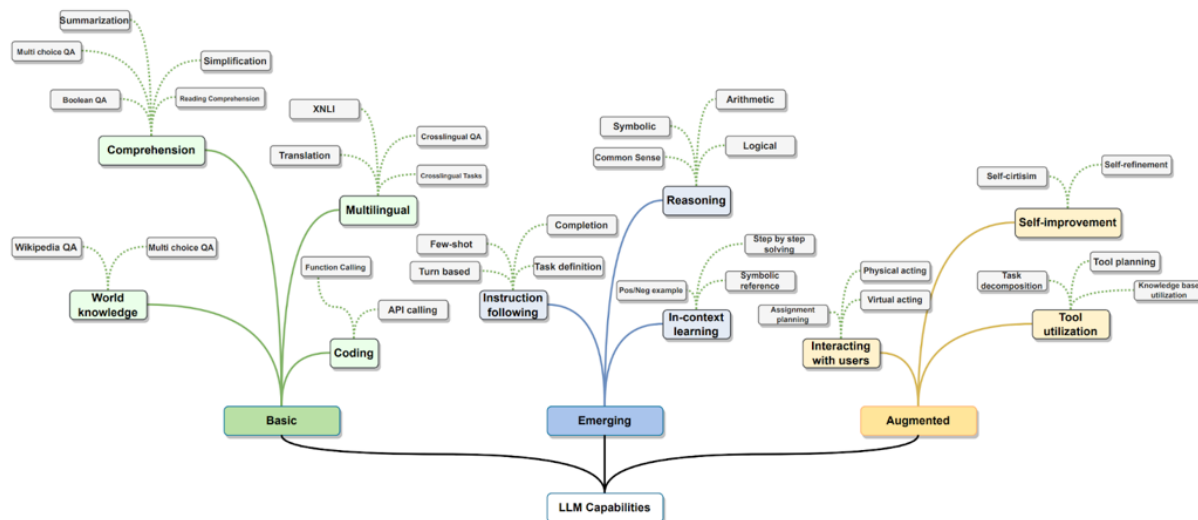


Figure 1.10: LLM Capabilities

1.4.1 Basic Capabilities

These are the fundamental tasks LLMs can perform, such as understanding, translating, or simple coding.

- **Comprehension:** Includes summarization, simplification, and answering multiple choice or true/false questions.
- **Multilingual:** Translation between languages, handling cross-lingual questions and tasks.
- **World Knowledge:** Using external facts, like Wikipedia, to answer questions.

- **Coding:** Performing simple programming tasks like function calling and using APIs.

1.4.2 Emerging Capabilities

These abilities reflect more advanced behaviors developed through improved training.

- **Instruction Following:** Completing tasks by following natural language instructions in few-shot or turn-based formats.
- **In-Context Learning:** Learning from examples given in the input without extra training.
- **Reasoning:** Applying logical and symbolic thinking, doing arithmetic, and using common sense to solve problems.

1.4.3 Augmented Capabilities

These are advanced features where LLMs interact with tools, people, or improve themselves.

- **Interacting with Users:** Engaging in physical or virtual actions and personalized responses.
- **Tool Utilization:** Planning and using external tools to solve tasks or gather knowledge.
- **Self-Improvement:** Reflecting on their actions to self-correct and refine over time.

This structured view shows how LLMs grow from basic understanding to interactive and intelligent agents.

1.5 Considering Human Aspects in Software Design

This section highlights the role of human-centered principles in software design.

1.5.1 Human aspects

Figure 1.11 shows four key human aspects in software engineering: Emotional, Cognitive, Social, and Well-being, each with related factors like Emotion, Motivation, Communication, and Stress. It highlights important human elements affecting software teams and work.



Figure 1.11: Taxonomy of Human Aspects Relevant to Software Engineering

In the next sections, we discuss some human aspects.

Empathy

Empathy is the capacity to understand, share, and appropriately respond to the emotions and perspectives of others. In human psychology, it involves both **affective empathy** (feeling what another person feels) and **cognitive empathy** (understanding another's emotional state) [25].

When applied to artificial intelligence (AI) and human-computer interaction (HCI), empathy refers to the system's ability to detect emotional cues, interpret the user's emotional or psychological state, and generate responses that are emotionally aware and supportive [26].

Empathetic AI aims not just to deliver correct answers, but to **engage users in a human-like, emotionally intelligent way** that builds trust, reduces frustration, and enhances user satisfaction.

Example: In medical chatbots, empathetic responses can significantly improve patient experience. For instance, when a patient expresses fear or uncertainty before a procedure, an empathetic bot might respond:

“I understand you’re feeling worried that’s completely normal. Let me explain the steps so you feel more comfortable.”

This emotional acknowledgment helps reduce anxiety and increases trust in the system.

Motivation

Motivation in AI refers to a system’s capacity to stimulate and sustain user engagement through tailored feedback, goal-setting mechanisms, and positive reinforcement [27]. Unlike human motivation, AI-driven motivation operates algorithmically, leveraging behavioral psychology principles such as rewards, progress tracking, and nudges to guide users toward desired actions.

In educational contexts, AI-powered systems have been developed to personalize learning experiences. For instance, the Artificial Intelligence Intelligent Assistant (AIIA) framework offers personalized and adaptive learning in higher education by understanding and responding to student inquiries, generating quizzes, and offering tailored learning pathways [28]. Similarly, generative AI models, like large language models, are being integrated into Intelligent Tutoring Systems (ITS) to enhance personalized education through dynamic content generation and real-time feedback [29].

Such systems enhance user persistence by combining progress visibility (e.g., analytics dashboards), positive reinforcement (e.g., praise), and goal personalization. The theoretical foundation behind this lies in Self-Determination Theory (SDT), which distinguishes between intrinsic and extrinsic motivation and guides AI reward system design [27]. Recent studies highlight the potential of AI-driven personalized learning to align with modern educational goals, emphasizing the need for systems that support cognitive engagement and the development of general competencies [30].

Furthermore, comprehensive analyses of personalized learning in smart education underscore the importance of student modeling and personalized recommendations. These approaches aim to meet individual learner needs and enhance their abilities by providing tailored educational experiences [31].

Personality and Personalization in AI

Personality in AI: Crafting Human-Like Engagement

Personality in AI refers to the deliberately designed behavioral traits (e.g., warmth, humor, formality) that shape user perceptions of trust, competence, and relatability. Research indicates that aligning a robot’s appearance and behavior with its intended task can enhance human-robot cooperation [32].

Example: A healthcare chatbot adopts an empathetic, reassuring tone: “I notice you’ve been logging more headaches this week. Let’s talk about what might help. ” Conversely, a finance chatbot employs a formal, data-driven personality: “Your Q2 savings grew by 12%. Recommend reallocating 15% to bonds. [View Analysis]”

Personalization in AI: Beyond One-Size-Fits-All

Personalization involves tailoring interactions based on user data (preferences, behavior, context) to deliver adaptive, uniquely relevant experiences. Unlike static personality traits, personalization evolves with the user, adjusting in real time [33].

Why It Works: Behavioral triggers such as time-based nudges (“You usually study Spanish at 9 PM. Ready for today’s lesson?”) and dynamic tone adjustments (softer feedback for

stressed users vs. challenge-focused feedback for confident users) enhance user engagement and satisfaction.

Example: Duolingo’s AI adjusts encouragement intensity (“Nice try!” vs. “You’re unstoppable! ”) based on user error patterns, leading to improved user motivation and retention.

Age

Age refers to the number of years a person has lived, but in the context of AI interaction, it represents more than just a number. Age influences cognitive abilities, sensory perception, memory retention, and preferred interaction styles. Younger users may be more familiar with fast-paced, playful, and visually rich interfaces, while older adults often appreciate clearer layouts, slower interaction speeds, and simplified language.

Example: An empathetic healthcare chatbot might use cheerful, casual language with younger users: *“Hey there! Want to check your health stats today?”* but switch to a more formal, reassuring tone with older adults: *“Good afternoon. Let’s review your recent health reports together.”*

Educational Level

Educational level refers to the highest degree or level of schooling a person has completed. This factor shapes a user’s vocabulary, conceptual understanding, and problem-solving approaches. A well-designed AI system adapts its explanations, feedback, and recommendations according to the user’s educational background, ensuring that interactions are neither too simplistic nor too complex.

Example: An educational chatbot teaching programming might explain a concept to a beginner as *“A variable is like a box where you store a number or a word.”* Whereas for an advanced user, it could say *“Variables act as references to memory locations, supporting dynamic type assignment.”*

Emotional Intelligence

Emotional Intelligence (EI) is defined as the ability to monitor one’s own and others’ feelings and emotions, to discriminate among them, and to use this information to guide one’s thinking and actions. This concept, introduced by *Salovey and Mayer* in 1990 [34], encompasses a set of skills that contribute to the accurate appraisal and expression of emotion, effective regulation of emotion in self and others, and the use of feelings to motivate, plan, and achieve in one’s life.

Example: In a project-based learning environment, an intelligent system detects that most students feel *frustration*. It responds by activating a supportive agent that rephrases instructions and offers targeted help.

When the dominant emotion shifts to *pride* (e.g., after completing a task), the system suggests peer-to-peer praise to boost group cohesion and engagement.

Adapting to dominant emotions helps sustain motivation throughout collaborative activities.

This algorithm selects a motivational strategy based on the dominant emotion detected each day. It personalizes encouragement by mapping emotions to appropriate supportive actions.

Example: Consider a scenario where a user expresses frustration with a technical issue. An emotionally intelligent AI assistant might respond: *“I understand this can be frustrating. Let’s work through this together step by step.”*

Such responses demonstrate the AI’s capacity to perceive emotional states and adapt its interactions accordingly, thereby improving user satisfaction and trust.

Importance in Empathetic AI Design Incorporating EI into AI systems is crucial for creating more natural and effective human-computer interactions. By understanding and appropriately responding to users’ emotional states, AI can provide support that is not only

Algorithm 2 Motivate Based on Dominant Emotion

```

1: Input: emotionLog[ ]
2: Output: motivationPlan[ ]
3: for each day in emotionLog do
4:   emotion  $\leftarrow$  GetDominantEmotion(day)
5:   if emotion in emotionToStrategy then
6:     strategy  $\leftarrow$  emotionToStrategy[emotion]
7:   else
8:     strategy  $\leftarrow$  "Stay positive"
9:   end if
10:  motivationPlan.append((day, strategy))
11: end for
12: return motivationPlan

```

functional but also emotionally resonant, leading to improved engagement and outcomes in various applications, from customer service to mental health support.

Active Listening

Active listening is the process of fully focusing on, understanding and thoughtfully responding to what another person is saying. Originally introduced by Rogers and Farson (1957), active listening goes beyond simply hearing words; it involves paying attention to both the explicit content and the underlying emotions, reflecting back understanding, and showing the speaker that they are truly heard and valued.

Example: In the telecommunications field, customers often reach out to report issues such as poor network coverage, billing errors, or service interruptions. An empathetic AI chatbot equipped with active listening capabilities would handle such complaints by:

- Acknowledging the issue empathetically, e.g., *"I understand how frustrating this must be for you."*
- Clarifying the problem with follow-up questions to ensure accurate understanding.
- Offering helpful steps or escalating the issue while maintaining a supportive tone.
- **Acknowledging the complaint:** Instead of jumping directly to solutions, the chatbot first paraphrases the user's message to show understanding. For example, if a customer writes, "I've been experiencing dropped calls all day, and I'm really frustrated!" the chatbot might respond, "I understand you've been facing repeated call drops today and it's causing a lot of frustration. I'm really sorry for this inconvenience."
- **Clarifying details if needed:** The chatbot may ask gentle follow-up questions to ensure it fully understands the problem, e.g., "Can you tell me if this issue happens in a specific location or everywhere you make calls?"
- **Offering solutions only after understanding:** Once the user feels heard, the chatbot moves to provide practical help: "Thank you for clarifying. Let me run a quick check on the network in your area. Meanwhile, I can also guide you through a few steps to improve the connection."

By actively listening, the chatbot not only addresses the technical issue but also helps reduce customer frustration, building trust and improving the user experience.

1.5.2 Human-Centered Approaches in Software Engineering

User-Centered Design (UCD)

UCD is a design approach that focuses on putting the needs, goals, and experiences of users at the center of the software development process. This means the focus is not only on the technical aspects of a system but also on making it easy to use, helpful, and engaging for the users. The approach involves including users throughout all stages of design and development, using methods like interviews, prototype testing, and gathering feedback. By doing this, software systems can be created that increase user satisfaction and improve how people interact with the system. When designing systems aimed at enhancing emotional intelligence and empathy, using UCD principles helps ensure that both the emotional and mental needs of users are carefully addressed, leading to more meaningful and human-centered experiences.[35]

Empathy-Driven Design (EDD)

EDD is a design approach that goes beyond usability and functionality by focusing deeply on understanding the emotional states, motivations, frustrations, and desires of users. It involves stepping into the users' shoes to design solutions that truly resonate with their feelings and create meaningful, human-centered experiences. EDD often uses tools like empathy mapping, storytelling, and user journey mapping to uncover the emotional context behind user behaviors, ensuring that the final product supports not only what users do but also how they feel.[36]

Personalization and adaptivity

Personalization and adaptivity are fundamental principles within the user-centered approach in software engineering, aiming to enhance system usability, relevance, and user satisfaction. Personalization refers to the system's ability to tailor its behavior, content, and interface to align with the individual characteristics, preferences, and past interactions of the user, thereby creating a more meaningful and efficient experience. Adaptivity complements this by enabling the system to dynamically adjust its responses and functionalities in real time, based on contextual factors such as the user's current goals, behavior, or environmental conditions. Together, personalization and adaptivity transform static software systems into responsive, intelligent solutions capable of evolving alongside users' changing needs. By integrating these principles, software products can achieve higher engagement, foster long-term user trust, and deliver more impactful, human-centered digital interactions.[37], [38]

Design thinking

Design Thinking is a user-centered, iterative approach to problem-solving that emphasizes deep empathy with users, collaborative ideation, rapid prototyping, and continuous testing. In software engineering, it serves as a framework to align product development closely with users' real needs and behaviors, ensuring that solutions are both usable and meaningful. By focusing on understanding the user context and involving stakeholders early and often, Design Thinking fosters innovation and reduces the risk of developing features that fail to deliver value. This approach complements personalization and adaptivity strategies by prioritizing human experiences and promoting flexible solutions that can evolve with user feedback.[39]

Think Aloud

The think-aloud protocol is a qualitative usability evaluation technique in which users verbalize their thoughts, perceptions, intentions, and decision-making processes while interacting with a system or interface. Beyond simply identifying usability problems, the think-aloud method provides designers with rich, real-time insights into how users mentally model the system, where

they encounter friction or confusion, and what aspects of the interface enhance or hinder engagement. By capturing this immediate cognitive feedback, teams can iteratively refine designs to improve not only task completion but also to optimize for fluidity, intuitiveness, and overall user satisfaction. Think-aloud thus plays a critical role in shaping software systems that are not only functional but also emotionally engaging and aesthetically appealing.[40]

By integrating these human-centered approaches including UCD, empathy-driven design, personalization, design thinking, and think-aloud evaluation software systems can be designed to go beyond mere functionality, creating interactions that feel intuitive, engaging, and emotionally meaningful. This foundation paves the way for solutions that not only meet technical requirements but also resonate deeply with users on a human level, setting the stage for advanced, emotionally aware digital experiences.

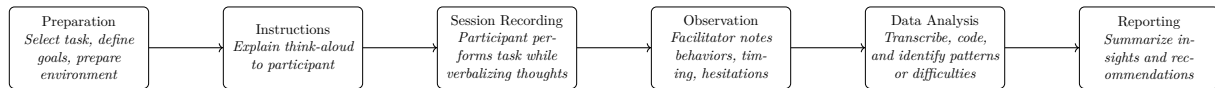


Figure 1.12: Visual Protocol of the Think-Aloud Method in Software Engineering/User Studies

1.6 Context and Work Positioning

1.6.1 Toward Human-Centered Digital Interactions in Algeria

The ongoing digital transformation is reshaping how individuals, institutions, and governments interact with technology worldwide. In Algeria, this shift is accelerating, with the rapid adoption of digital services, virtual assistants, and AI-driven tools across various sectors. As citizens increasingly rely on digital platforms to communicate, access services, and express concerns, ensuring that these interactions remain meaningful, empathetic, and human-centered interactions becomes crucial.

However, despite technological advancements, many digital solutions both globally and locally struggle to incorporate human aspects such as emotional intelligence, empathy, and cultural sensitivity. This oversight is particularly visible in the Algerian context, where digital products often prioritize technical performance and scalability over user experience that accounts for local social and emotional nuances. This creates a gap between citizens and institutions, reducing trust, satisfaction, and long-term engagement.

1.6.2 Challenges

Several challenges emerge within this context:

- A scarcity of localized datasets that reflect Algerian dialects, emotional expressions, and cultural communication styles.
- Limited adoption of user-centered design principles in the development of local digital services.
- Technical constraints in adapting large language models to recognize and respond appropriately to the emotional and cultural context of Algerian users.

This project is positioned within this landscape, addressing the need for digital solutions that go beyond basic functionality to embrace human aspects. By considering the realities of Algeria's digital transition, the mass use of virtual assistants, and the importance of emotion management, the work aims to design a framework that integrates advanced natural language processing with empathetic, user-centered interaction. The next chapter will present the design and conceptual foundation of this framework, showing how it seeks to bridge the gap between citizens and institutions through emotionally aware conversational systems.

1.6.3 Our Missions

Our mission is to develop an AI-based assistant tool designed to support citizens through a chatbot that considers human aspects, starting with sensitivity to context and emotions. This chatbot is capable of generating both typical and empathetic responses that can influence the psychological state of users. It aims to provide meaningful assistance to citizens who interact with the digital solution for their informational needs, inquiries, and expressions.

1.6.4 Planned Timeline

The following table presents the preliminary schedule for the master's project, covering the period from November 2024 to May 2025:

Project Phase	Time Period
Problem analysis and literature review	November – December 2024
Dataset creation and data collection	December 2024 – January 2025
Language model development	January – February 2025
UX design and interface development	February – March 2025
Testing and validation phase	March – April 2025
Writing the master's thesis	April – May 2025

Table 1.2: Planned schedule for the master's project

1.7 Conclusion

In this chapter, we explored the fundamentals of natural language processing and large language models, emphasizing the importance of incorporating human-centered aspects in their design. We examined user-centered design approaches that focus on understanding users' emotional and behavioral needs to create more intuitive and empathetic interactions. These concepts highlight how integrating human factors can lead to more meaningful and effective digital experiences. Building on this theoretical foundation, the next chapter will present the context and positioning of this work, setting the stage for the design and development of a comprehensive framework for emotionally intelligent chatbot systems.

Chapter 2

Design of Our Solution

2.1 Introduction

In this chapter, we present the foundations of our empathetic chatbot framework, designed to enhance user interaction through emotional and contextual awareness. We begin by discussing the motivation for building more human-centric conversational agents and the challenges that arise in detecting user emotions, interpreting intent, and delivering appropriate responses.

We then introduce our proposed framework, which integrates rule-based techniques and machine learning models to manage complex dialogue situations. A step-by-step overview of our solution is provided, illustrating the evolution from a basic system to a more advanced architecture that incorporates user experience (UX) design components. Through this approach, we aim to demonstrate how empathetic interaction can be technically realized and practically deployed in real-world scenarios.

2.2 Motivation and Challenges

2.2.1 Motivation

In recent years, the integration of human-centric qualities such as empathy into human-computer interaction has emerged as a critical research direction especially in the realm of conversational agents and chatbots.

Scientific Gap: While existing systems are effective in delivering factual or transactional responses, they often fail to consider **emotional** and **contextual** cues, which limits their ability to simulate human-like interaction.

Scientific Importance: Bridging this gap pushes the boundaries of **natural language processing** (NLP) and **large language models** (LLMs) toward more nuanced, context-aware, and empathetic systems. The capacity to simulate empathy in AI improves:

- User trust
- User engagement
- Satisfaction with digital services

Practical Impact: In customer service, empathetic chatbots reduce pressure on human agents by handling routine or emotionally sensitive cases. This results in:

- Cost reduction
- Operational efficiency
- Stronger user loyalty

Social Relevance: Context-aware systems using **retrieval-augmented generation** (RAG) or **rule-based intent detection** can respond more appropriately across diverse user populations.

Example: In countries like Algeria, where many companies still lack basic chatbot infrastructure, an empathetic chatbot serves as a digital bridge, enhancing communication between citizens and institutions.

Novel Contribution: Unlike traditional models that focus solely on information delivery, our solution integrates:

- User emotion
- User motivation
- Adaptive response generation

This enables both a **richer user experience** and deeper insight into user sentiment and evolving needs.

Example: In a telecom company in Algeria:

- 30 employees each answer 20 repetitive questions/day.
- Each answer takes 2 min.

Daily loss: $30 \times 20 \times 2 = 1200 \text{ min} = 20 \text{ hours}$.

Cost: $20 \times 700 \text{ DZD} = 14\,000 \text{ DZD/day} \Rightarrow 308\,000 \text{ DZD/month}$.

Deploying a chatbot can reduce this recurring loss significantly.

First Motivating Example

Imagine *Amina*, a young woman traveling alone for the very first time. She arrives at the airport with a mix of nervousness and excitement, clutching her ticket and documents tightly. This trip is more than a journey it's a dream coming true, an opportunity to explore a new place and create unforgettable memories.

Suddenly, Amina's heart sinks: her flight has been delayed, or worse, she's misunderstood the boarding gate instructions and just watched the plane take off without her. A wave of **panic** crashes over her. She feels **confused**, **scared**, and **overwhelmed** in this vast, unfamiliar airport. Her mind races with questions: *"What do I do now? Where can I get help? Am I going to be stuck here?"*

Desperate, *Amina* reaches for her phone to call customer service but the line is busy. She heads to the service counter, only to find a long queue and tired, overworked staff who barely meet her gaze. The mounting anxiety turns into **frustration** and a growing sense of **isolation**.

Now, imagine a different scenario: Amina opens the airport's virtual assistant app on her phone.

- **If the chatbot responds with cold facts:** *"The flight has departed. Please check the next available flight."*

Amina feels **dismissed** and **alone**, her distress ignored. The message offers no comfort or guidance, deepening her despair.

- **But if the chatbot is emotionally intelligent and empathetic:**

"Amina, I understand how frightening and overwhelming this must feel, especially on your first solo trip. Take a deep breath I'm here to help you."

"Let me guide you through your options. You can rebook on the next flight at counter A5, and I'll send you a clear step-by-step guide on what to do next."

"If you need special assistance or just want someone to talk to, I can connect you with an airport agent immediately. You're not alone in this."

Impact identified

For Amina (the user):

She feels genuinely **heard**, **understood**, and **supported**.

Her overwhelming anxiety eases as she regains a sense of control and direction.

Her trust in the airline and airport's digital services grows, increasing her loyalty.

For the Airport/Company:

Emotional support reduces the strain on human agents, allowing them to focus on critical cases.

Enhanced customer satisfaction leads to positive reviews and stronger brand reputation.

The collection of emotional and contextual data helps the airport improve future services and crisis handling.

This example demonstrates how an empathetic chatbot can transform a stressful, confusing situation into a manageable, reassuring experience. For citizens like Amina, such emotionally aware technology offers not just information, but genuine care building trust and loyalty in a way cold facts never could.

Second Motivating Example

Imagine a customer named **Samir**, who relies heavily on his mobile phone for work and staying connected with family. One day, he suddenly loses network coverage in his area due to a technical issue. He tries to make urgent calls but can't get through. Frustration quickly turns to anxiety and worry especially because he was expecting an important call.

Samir immediately contacts the telecom company's customer service through their app.

If the chatbot responds factually:

"There is a network outage in your area. We are working on it. No estimated time for resolution."

Samir feels ignored and more stressed. He has no idea when the problem will be fixed and feels helpless. His frustration grows, increasing the chance he might switch providers or complain publicly.

If the chatbot is empathetic and emotionally aware:

"Hi Samir, I completely understand how upsetting it is to lose your connection when you need it most. We are really sorry for the inconvenience this outage is causing you."

"Our technicians are working hard to fix the issue as quickly as possible. Based on the latest update, service should be restored within the next 2 hours."

"Meanwhile, if there's anything urgent, I can help you find alternative ways to stay connected or assist with compensation requests."

"Thank you for your patience I'm here with you every step of the way."

Impact identified

For Samir (the user):

- Feels understood and valued despite the disruption.
- Experiences reduced anxiety knowing the expected resolution time.
- More likely to stay loyal to the telecom provider.
- Gains trust that the company cares about his experience.

For the Telecom Company:

- Decreased volume of angry calls or complaints to live agents.
- Improved customer satisfaction and brand reputation.
- Ability to collect emotional feedback and use it to improve service.
- Potentially higher customer retention during outages.

2.2.2 Emotion and context awareness

Emotion and context awareness are central to designing conversational agents that go beyond transactional efficiency to deliver truly human-centered interactions. Emotion awareness enables the system to detect, interpret, and respond to the user's emotional state whether it's frustration over a service issue or satisfaction with a solution allowing the chatbot to act not just as a tool, but as a responsive companion. This is particularly powerful in domains like telecom customer service, where an emotionally-aware chatbot can identify when a user is upset about recurring network failures and offer an empathetic acknowledgment before proceeding with troubleshooting steps. Such responsiveness builds trust and reduces perceived coldness in machine interactions.

Context awareness complements this by allowing the system to adapt its behavior based on situational cues such as user history, current activity, time, or location. Together, emotion and context awareness shift chatbots from reactive tools to proactive, intelligent assistants. In environments like Algeria where digital support systems are often limited or impersonal this dual awareness helps bridge the gap between citizens and institutions by delivering relevant, sensitive, and culturally adaptive communication. It also empowers companies to better understand evolving user sentiment, improving both service quality and customer relationships. By embedding these capabilities, AI systems become more empathetic, accessible, and aligned with real human needs.

We express the relationships between empathy and three key constructs emotion, context, and intent using semantic operators inspired by formal modeling.

1. Empathy depends on Emotion

Formal relation: $\text{Empathy} \Leftarrow \text{Emotion}$

Empathy arises from the recognition and internal simulation of emotional states. Emotional awareness is a prerequisite for empathic engagement.

2. Empathy requires Context

Formal relation: $\text{Empathy} \models \text{Context}$

Empathy requires contextual grounding to be accurate and ethically appropriate. Without context, emotional cues may be misread.

3. Empathy informs Intent

Formal relation: $\text{Empathy} \Rightarrow \text{Intent}$

Empathy plays a generative role in shaping communicative or behavioral intent, aligning actions with the perceived emotional and situational needs of others.

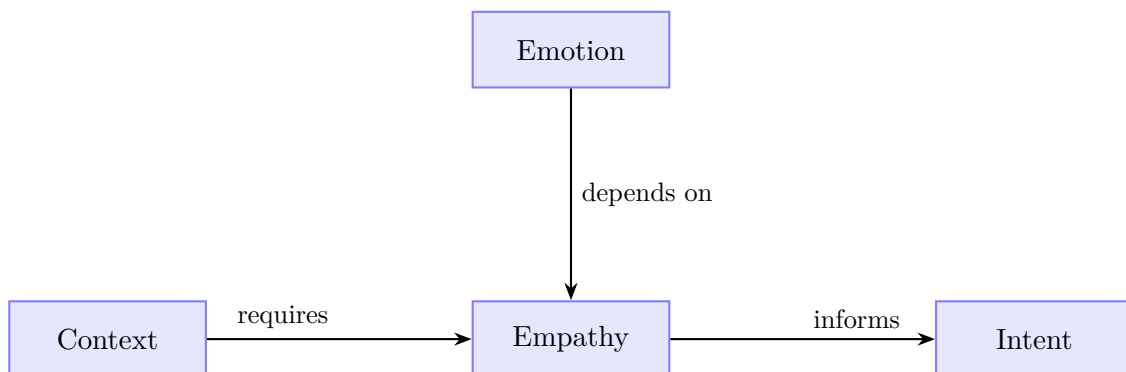


Figure 2.1: Semantic relationships connecting empathy with emotion, context, and intent.

This table (2.1) presents example telecom customer responses alongside the extracted empathy-related keywords or phrases. It highlights how emotional understanding is embedded in typical service communications.

Telecom Response	Empathy Keywords Extracted
We <i>understand</i> your <i>frustration</i> with the slow internet speed and <i>sincerely apologize</i> for the <i>inconvenience</i> caused. Our team is <i>committed</i> to <i>resolving</i> this issue as quickly as possible to restore your service.	<i>understand, frustration, sincerely apologize, inconvenience, committed, resolving</i>
Thank you for your <i>patience</i> during this outage. We <i>realize</i> how <i>important</i> your connection is, and we're <i>working hard</i> to <i>fix</i> the problem <i>promptly</i> .	<i>patience, realize, important, working hard, fix, promptly</i>
We <i>truly regret</i> the disruption you are facing and <i>appreciate</i> your <i>understanding</i> as we investigate the root cause. Please <i>rest assured</i> we are doing everything to <i>assist</i> you.	<i>truly regret, appreciate, understanding, rest assured, assist</i>

Table 2.1: Example telecom responses and extracted empathy keywords

2.2.3 Challenges

Based on the discussion, this paper takes a first step toward designing emotionally and contextually aware conversational agents to enhance user experience in service disruptions. It identifies three key challenges:

- C1.** How to accurately detect and interpret users' emotions for empathetic responses?
- C2.** How to incorporate context (user history, situation, culture) for personalized guidance?
- C3.** How to balance automated support with human intervention to maintain trust and satisfaction?

2.3 Our Framework

In this section, we introduce our proposed solution: an empathetic, context-aware and emotionally intelligent chatbot framework designed to enhance communication between citizens and various institutions, including telecommunications, healthcare, education, finance, and e-government sectors. See Figure 2.2

The key idea behind this framework is to provide a flexible and modular system that can be tailored to the specific context of each deployment. Depending on the available data, the target sector and operational needs. The framework supports multiple implementation scenarios, each exploring different strategies for embedding empathy in chatbot interactions.

- **User Input:** A citizen or customer sends a message to the chatbot.
- **Empathetic System:** This is the core component that processes the input and generate an empathetic response.
- **Output:** The chatbot delivers an empathetic and contextually relevant response back to the user.
- **User Feedback:** The user can provide feedback on the chatbot's response.

- **Admin Interaction:** Administrators have access to the conversation logs and may intervene by responding directly to users when necessary.
- **Bias Detection and Reformulation:** The empathetic system monitors admin responses for cognitive biases and can reformulate messages to maintain empathy before delivering them to the user.

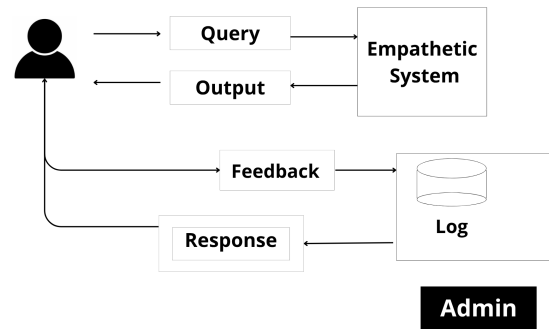


Figure 2.2: General vision

2.3.1 Possible Scenarios for Empathetic System

To adapt the empathetic chatbot framework to different needs and technical capacities, we outline several possible scenarios for embedding empathy. These scenarios illustrate distinct strategies for generating emotionally intelligent responses using Large Language Models (LLMs), ranging from basic usage to more sophisticated techniques.

Base Model

The **base model** scenario refers to using a pre-trained Large Language Model (LLM) directly, without any additional adaptation or customization. This model such as OpenAI’s GPT or Google’s Gemini is trained on a broad corpus of general data and provides out-of-the-box capabilities for natural language understanding and generation.

While base models offer a powerful starting point, they typically lack contextual sensitivity and domain-specific alignment. As a result, their responses may appear generic or misaligned with the user’s emotional state or the institutional context. This often leads to less effective communication, especially in scenarios that require nuanced understanding and empathetic interaction.

Input

User’s query or prompt, e.g., a question or statement.

Output

Response generated by the base model, which is often generic, lacking contextual sensitivity or domain-specific alignment.

Prompt Engineering

In this scenario, empathy is introduced through carefully crafted prompts that guide the LLM's response style and emotional tone. Prompt engineering allows developers to simulate empathy by incorporating emotional cues, context, and intent directly into the model's input.

This method enhances the model's ability to generate more emotionally aligned and human-like responses without requiring retraining. However, it comes with limitations. If prompts become too lengthy, the response time may increase significantly. Additionally, prompt engineering cannot fully correct underlying model biases or ensure deep contextual adaptation especially in sensitive domains where emotional nuance and domain accuracy are critical.

Input

A carefully designed prompt embedding emotional cues, context, and intent to guide the model's empathetic response.

Figure 2.3: Example of Input Prompt with Emotional and Contextual Guidance

Output

A model-generated response that aligns emotionally and contextually with the input prompt, producing more human-like and empathetic answers.

Figure 2.4: Example of Output Response Guided by Prompt Engineering

In order to achieve our solution, we proceeded in an interactive and incremental way, starting from a basic approach and progressively integrating new dimensions to improve it, as shown in Figure 2.5.

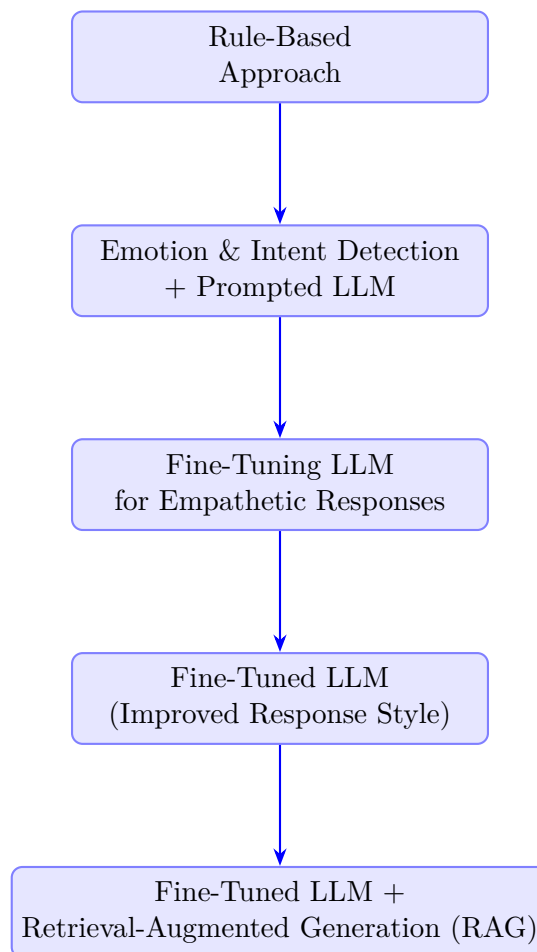


Figure 2.5: Step-by-step Evolution of the Empathetic Chatbot Solution

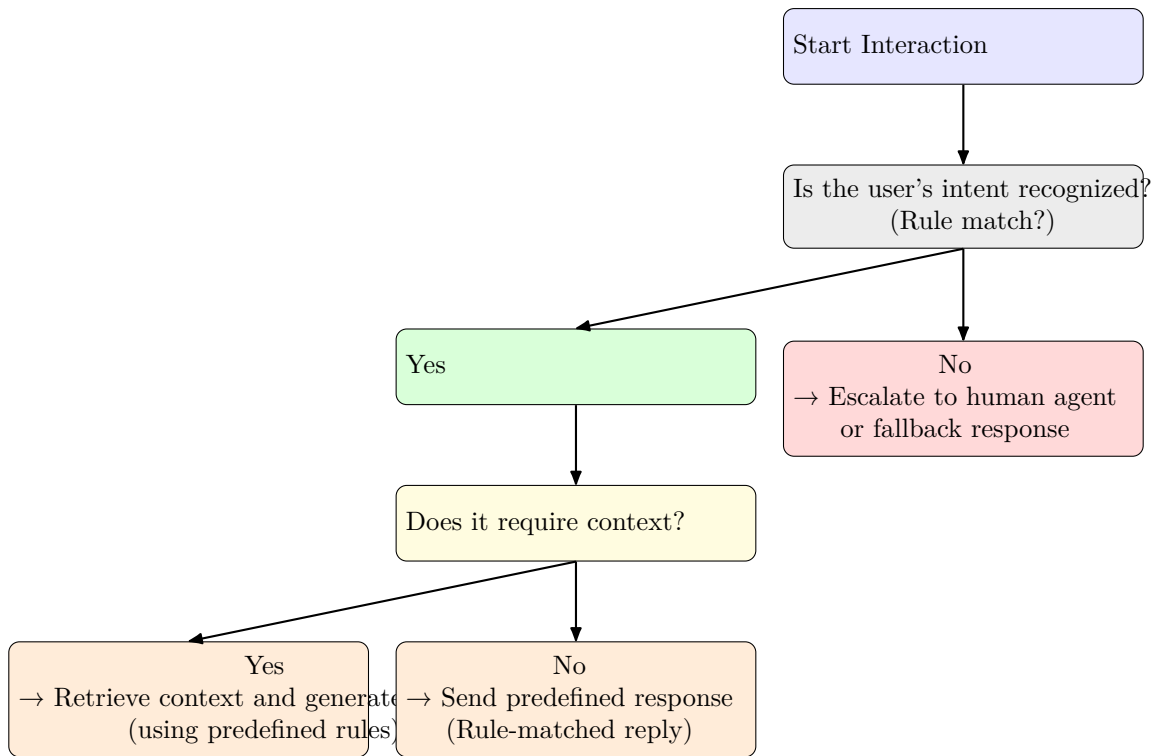


Figure 2.7: Alternative Styled Rule-Based Chatbot Interaction Schema

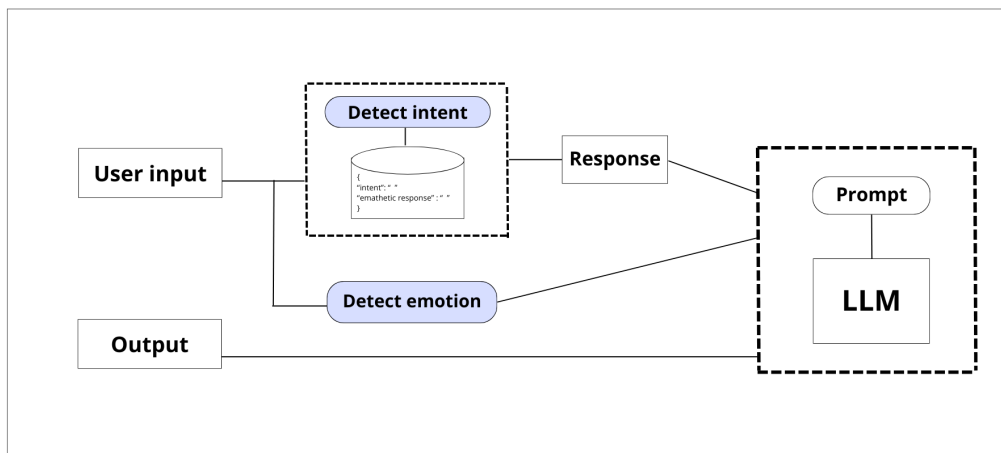


Figure 2.8: Emotion and Intent Detection with Prompted LLM

Fine-Tuning LLM for Empathetic Responses

In this scenario, the system leverages a Large Language Model (LLM) that has been fine-tuned on empathetic dialogue datasets. The fine-tuning process enables the model to generate responses that naturally convey empathy, adapting its language style and tone to the emotional nuances found in the training data.

However, while the fine-tuned LLM improves the quality and emotional resonance of responses, it may still lack dynamic contextual awareness of the specific user or real-time inter-

action details. To address this limitation, additional components such as context integration layers or retrieval-augmented generation (RAG) can be incorporated to supply up-to-date and domain-relevant information.

This scenario offers a balance between enhanced empathy through fine-tuning and the potential need for external context modules to ensure relevance and personalization in chatbot interactions.

As shown in Figure 2.9, the fine-tuning process enhances the LLM’s ability to produce empathetic and emotionally nuanced responses.

Input User query or prompt $\longrightarrow$ **Fine-Tuned LLM** Empathy-trained model $\longrightarrow$ **Empathetic Response**

Figure 2.9: Pipeline of an Empathy-Fine-Tuned LLM from User Input to Output

Fine-Tuned LLM for Empathetic Responses

In this scenario, a Large Language Model (LLM) is fine-tuned on domain-specific and empathetic dialogue datasets. This adaptation enables the model to generate responses that naturally reflect empathy and emotional intelligence tailored to the target sector. The fine-tuned model improves the chatbot’s ability to communicate in a more human-like and emotionally aware manner.

However, this approach may still lack real-time access to updated contextual information or user-specific data, potentially limiting response relevance.

Fine-Tuned LLM with Retrieval-Augmented Generation (RAG)

Building upon the fine-tuned LLM scenario, this approach integrates Retrieval-Augmented Generation (RAG) to enhance contextual awareness. Here, the fine-tuned model accesses an external knowledge base or domain-specific documents during interaction, retrieving up-to-date, relevant information.

The combination of a fine-tuned empathetic model and RAG ensures that responses are not only emotionally intelligent but also factually accurate and contextually grounded, further improving user satisfaction and trust in the chatbot.

2.4 Solution Overview

Our proposed solution is a hybrid framework that integrates rule-based logic with advanced AI components to generate emotionally appropriate and contextually relevant responses. This architecture is designed to dynamically adapt based on the complexity of the user’s input, the emotional tone, and the system’s confidence in understanding the user’s intent.

2.4.1 Why a Hybrid Approach?

This hybrid design addresses both technical challenges and user experience expectations:

- Rule-based modules provide fast, explainable, and controllable behavior for simpler or well-defined queries.
- Prompt engineering allows us to use powerful pre-trained LLMs efficiently for lightweight yet emotionally enriched generation.
- Fine-tuned LLMs and Retrieval-Augmented Generation (RAG) enable deep understanding and contextual relevance in complex or knowledge-driven conversations.

2.4.2 Formalization

We formally define the chatbot system and its core semantic-affective components queries, intents, emotions, context, empathy, and response functions as follows:

Definition 3.1. Chatbot System A chatbot is defined as a tuple:

$$\mathcal{C} = \langle Q, R, I, E, C, \mathcal{E}m, M, \mathcal{F} \rangle$$

where:

- Q : Set of user queries/questions.
- R : Set of system responses.
- I : Set of detected user intents.
- E : Set of user emotional states.
- C : Set of context factors (temporal, social, cultural).
- $\mathcal{E}m$: Empathy function over E, I, C .
- $\mathcal{F}$: Response generation function (rule-based or ML-based).

Definition 3.2. User Dataset Let $\mathcal{D} = \{(q_i, r_i)\}_{i=1}^n \subset Q \times R$ be the dataset of question-response pairs used to train or evaluate the chatbot system.

Definition 3.3. Intent (I) A communicative goal inferred from a user's query. Each $i \in I$ represents the user's intention, such as requesting, informing, or complaining. Formally:

$$f_I : Q \rightarrow I$$

Definition 3.4. Emotion (E) A subjective affective state extracted from the user's message or tone. Each $e \in E$ may reflect sadness, anger, joy, etc. Formally:

$$f_E : Q \rightarrow E$$

Definition 3.5. Context (C) A set of external or situational factors $C = \{c_1, c_2, \dots, c_k\}$ that influence how intents and emotions are interpreted. These include temporal, historical, and socio-cultural attributes.

Definition 3.6. Empathy ($\mathcal{E}m$) The capability to simulate or respond appropriately to a user's internal state. We model empathy as a function:

$$\mathcal{E}m : E \times I \times C \rightarrow \mathbb{R}$$

where the output reflects an empathy score or degree of empathic engagement that can be used to inform response generation or motivation inference.

Definition 3.7. Response Generation (Rule-Based) A handcrafted rule dictionary:

$$\mathcal{R} : I \times E \rightarrow R$$

maps intent-emotion pairs to appropriate responses.

Definition 3.8. Response Generation (ML-Based) A data-driven model:

$$\mathcal{F}_{ML} : Q \rightarrow R$$

learned from $\mathcal{D}$, directly mapping queries to responses.

2.4.3 Architecture Modules

The solution operates via two main processing pathways, selected based on interaction complexity and confidence in detected intent.

Rule-based + Prompted LLM Pipeline

Used in real-time, lightweight interactions.

1. **User Input:** Received through the mobile interface.
2. **Emotion Detection:** A pre-trained LLM (e.g., Maslowe) classifies the user's emotional state.
3. **Intent Detection:** A lightweight model (e.g., BERT) detects the user's intent.
4. **Response Retrieval:** An empathetic response is selected from a curated knowledge base based on intent.
5. **Prompt Construction:** The retrieved response is enriched with emotional and linguistic cues tailored to the user profile.
6. **Prompted LLM Generation:** The prompt is passed to a general-purpose LLM for final response generation or refinement.

Fine-tuned AI Model + Retrieval-Augmented Generation (RAG)

Used for deeper conversations and complex user needs.

1. **Fine-tuned Empathetic LLM:** A domain-specific LLM trained on empathetic dialogue and custom data is queried.
2. **Contextual Retrieval (RAG):** External up-to-date information is retrieved based on the user's query and injected into the LLM's context window.
3. **Emotionally-Aware Generation:** The LLM generates a personalized response informed by both retrieved context and emotional state.

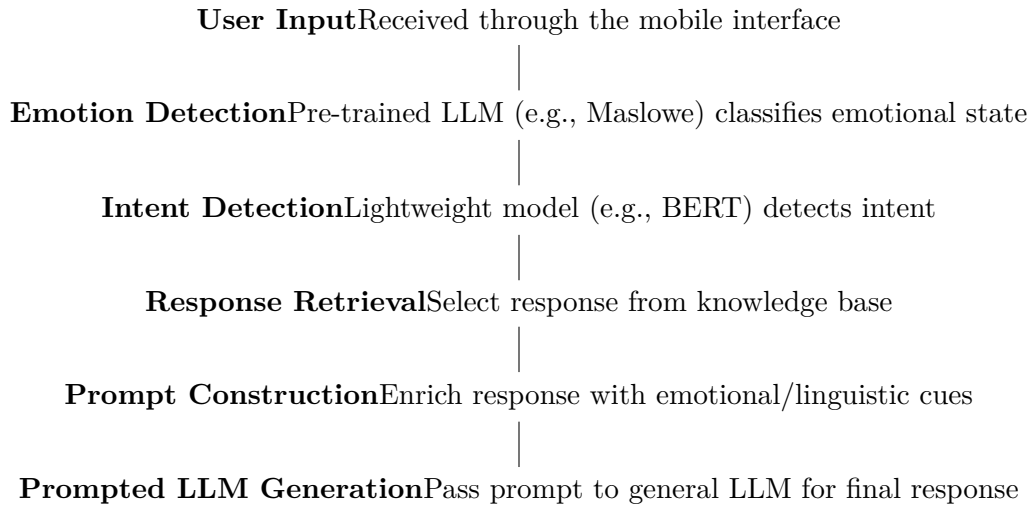


Figure 2.10: Compact vertical flow of empathetic response generation pipeline.

User Profiling and Personalization Layer

To deliver user-centric interactions, a dynamic profile is maintained for each user, containing:

- Interaction history (frequent intents, emotions, feedback).
- Communication preferences (tone, formality, verbosity).
- Specific needs (e.g., anxiety-prone, prefers reassurance or concise responses).

This profile is used in multiple modules to ensure personalization:

- **Emotion & Intent Detection:** Classifiers take into account the user's prior emotional patterns and known expressions.
- **Prompt Construction:** Prompts are adjusted based on preferred tone, phrasing, and sensitivity thresholds.
- **RAG Retrieval:** Retrieved content is filtered based on user interests or relevance history.
- **Final Generation:** Output is tailored linguistically and emotionally to fit the user's communication style and psychological needs.

Human-in-the-loop and Bias Management

To ensure quality assurance, ethical oversight, and system adaptability:

- Users may rate or comment on responses to provide direct feedback.
- A dashboard enables human admins to review flagged interactions and provide or adjust responses.
- To prevent biased or inappropriate human replies, the system integrates an empathetic system-bias-aware to do post-processing reformulation. This step ensures that any manual input is emotionally calibrated before reaching the user.

Use Case Scenarios

Three response strategies are used: LLM prompting, fine-tuned AI with retrieval, and admin intervention for unresolved queries.

- **Scenario 1 – Prompted LLM:** When the user’s request is clear but emotionally sensitive, the system retrieves a base response and enriches it via emotion-aware prompting.
- **Scenario 2 – Fine-tuned AI with RAG:** For complex, ambiguous, or information-seeking queries, the system uses fine-tuned models and external retrieval to generate deep, context-aware replies.
- **Scenario 3 – Admin Intervention:** When feedback indicates dissatisfaction, the admin can intervene and provide a revised response, which is then reprocessed through the empathy layer to ensure consistency.

Figure 2.11 shows the system’s adaptive response flow from automated to human intervention.

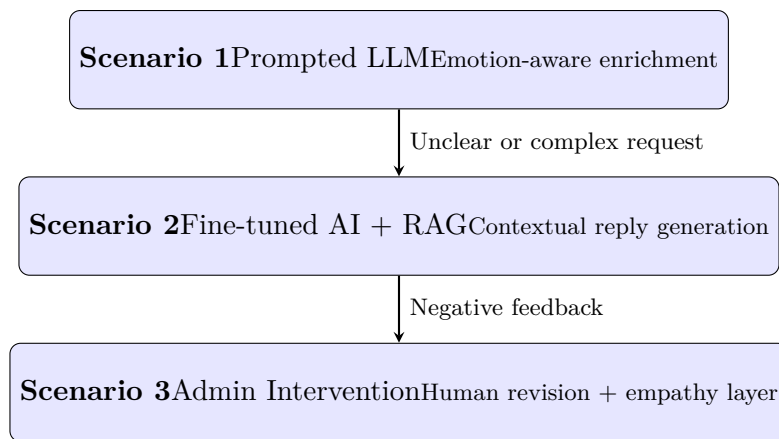


Figure 2.11: Response transition scenarios: from automated response to human intervention

2.4.4 System Architecture

In this section, we present the *System Architecture* of our chatbot.

Conceptual Overview

Our system follows a modular, hybrid architecture designed to process emotional and intent-driven user queries in real time. It integrates both **lightweight rule-based pipelines** and **fine-tuned AI models**, depending on the complexity of the interaction. The system is accessed through a **Flutter mobile application**, with a Python-based backend and external LLM services.

Data Flow Summary

We summarize the data flow as shown in Figure 2.13.

1. User sends input via mobile app
2. Access through API:
 - Detects emotion

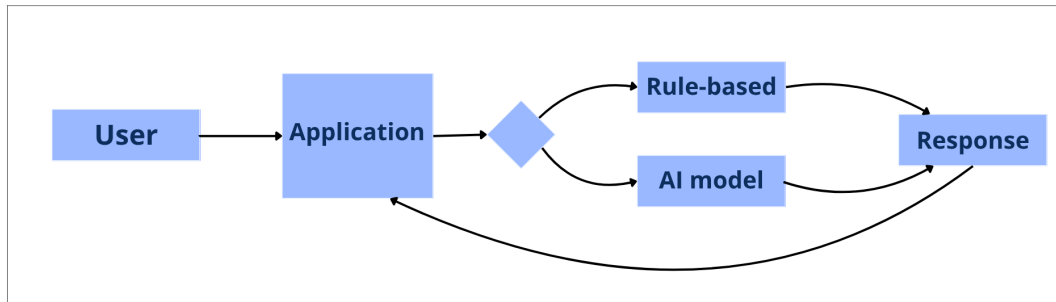


Figure 2.12: Hybrid model

- Detects intent
3. If input is simple:
 - Retrieves prewritten empathetic response
 - Constructs enriched prompt
 - Sends to LLM
 4. If input is complex: Sends to fine-tuned TinyLLaMA + RAG
 5. Output is personalized using user profile
 6. Response is returned to mobile app

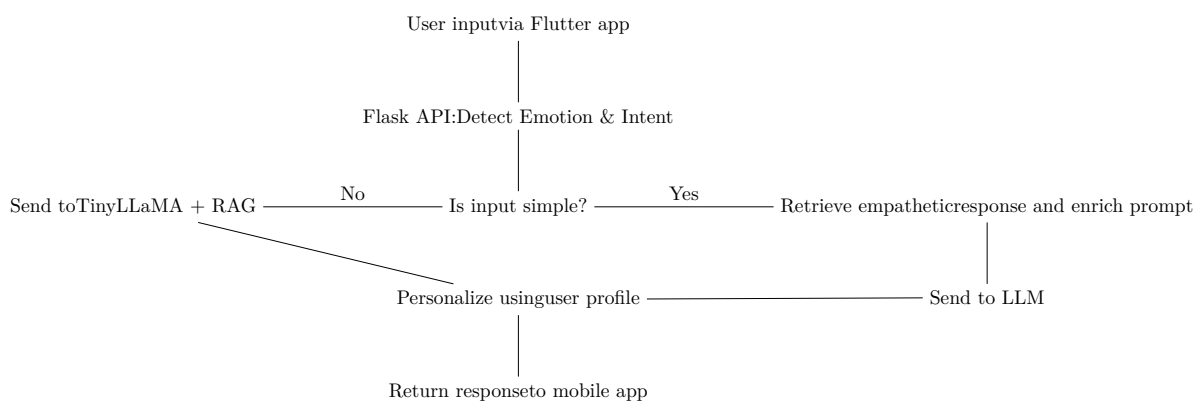


Figure 2.13: Organigram of the chatbot interaction pipeline

2.5 UX Design of our solution

In this section, we discuss the UX design of our solution, which can also be used to enhance user interaction with our system.

This table (2.2) summarizes common UX challenges in dedicated client chatbots, along with effective strategies to address them. It highlights key design components and principles such as push notifications, emotion menus, and conversation history that enhance user engagement and emotional connection. These elements collectively improve usability, trust, and interaction quality in chatbot systems.

Problem / UX Challenge	Strategies Used	UX Design Components / Principles
Limited User Engagement	Use push events (notifications/prompts) to proactively guide users	Push Notifications, Timely Alerts, Context Awareness, User Attention Management
Users Forget Previous Interactions	Maintain conversation history accessible	Chat History Display, Session Persistence, Context Retention, Easy Navigation
Emotionally Flat Interaction	Integrate emotion menu for users to express feelings	Emotion Selection Menu, Emotional State Detection, Personalization, Empathy Design
Users Confused by Response	Provide clear intent feedback and confirmation	Intent Confirmation, Feedback Prompts, Clear Messaging, Error Handling
Low Trust or Connection	Use personalized messages and adaptive tone	Personalization, Adaptive Language Style, Trust-building UX, Consistency
User Overload or Frustration	Break tasks into smaller steps, offer suggestions	Progressive Disclosure, Guided Conversations, Suggestion Buttons, Minimalist Design
Difficulty in Expressing Needs	Offer quick-reply buttons and FAQs	Quick Replies, Menu Options, FAQ Integration, Ease of Use
Lack of Emotional Awareness	Detect sentiment and adapt chatbot responses accordingly	Sentiment Analysis, Context-aware Response, Emotional Intelligence Integration

Table 2.2: UX Design Strategies and Components for a Dedicated Client Chatbot

2.5.1 Push Notifications and Timely Alerts

Push notifications attract user attention at the right moment → increase engagement → reduce abandonment.

The chatbot sends a notification: *"Hello! It looks like you need help completing your order. May I assist you?"*

Targeted notification → encourages user return → improves conversion.

2.5.2 Chat History Display and Session Persistence

Displaying conversation history → improves continuity → increases contextual understanding → reduces frustration.

The user can review previous product inquiries → facilitates decision making → improves satisfaction.

2.5.3 Emotion Selection Menu and Emotional State Detection

An emotion menu → encourages user expression → strengthens emotional connection → improves response personalization.

The chatbot asks: *"How are you feeling today?"* with options like **Happy, Confused, Frustrated** → tone adaptation → increased empathy.

2.5.4 Intent Confirmation and Clear Messaging

Confirming user intent → reduces errors → increases trust → improves communication clarity.

The chatbot asks: *"You want details on 'Wireless Headphones,' correct?"*
Confirmation → avoids misunderstandings → improves conversation efficiency.

2.5.5 Personalization and Adaptive Language Style

Personalized messages → create a sense of attention → strengthen trust → increase engagement.

The chatbot greets: *"Welcome back, Sarah! Would you like to continue browsing the headphones you viewed last time?"*

Personalization → loyalty → better user experience.

2.5.6 Progressive Disclosure and Guided Conversations

Breaking complex tasks into steps → reduces cognitive overload → effectively guides the user → improves task completion.

The chatbot asks: *"What is your delivery address?"* then *"Do you prefer standard or express shipping?"*

Clear steps → simplification → higher success rate.

2.5.7 Quick Replies, Menu Options, and FAQ Integration

Quick reply buttons → speed up interaction → reduce user effort → improve conversation flow.

The chatbot offers: **Track my order, Return an item, Contact support** → quick options → reduced response time.

2.5.8 Sentiment Analysis and Emotional Intelligence Integration

Sentiment analysis → mood detection → empathetic response adaptation → strengthened user-chatbot bond.

When frustration is detected, the chatbot responds: *"I'm sorry you're having difficulties. Let's see how I can help you quickly."*

Empathetic response → satisfaction improvement → increased loyalty.

2.6 Instantiation of our framework

In this section, we present an instantiation of our proposed chatbot through three main steps: (i) First step: Need, Intent, and Emotion; (ii) Second step: Response Strategy and Generation; and (iii) Third step: Delivery, Feedback, and Learning.

2.6.1 First step: Need, Intent, and Emotion

The step-by-step pipeline begins with understanding the **user's information need**, as shown in Table 2.3. This part focuses on identifying the customer's request, detecting their intent using rule-based or language model methods, and analyzing their emotional state. For example, if a

user expresses frustration about internet speed, the system detects both the intent (e.g., complaint) and the emotion (e.g., anger). This combination helps the system infer user motivation and urgency, allowing it to prepare for an appropriate response.

N°	Step Name	Example
1	Information Need	A customer needs to ask the chatbot why their internet speed is suddenly very slow.
2	Intent Identification	If the intent matches a known category (e.g., network issues, billing), a rule-based approach is used. Otherwise, a language model like BERT is used to detect the intent (e.g., "complaint").
3	Emotion Detection	The chatbot detects that the customer is angry: "This internet is useless! I can't do anything!" It classifies the emotion as "anger".
4	Motivation Inference	The combination of anger and the complaint intent implies urgency and the need for immediate support or compensation.

Table 2.3: Chatbot interaction pipeline (Part 1/3): Need, Intent, Emotion, Motivation

2.6.2 Second step: Response strategy and generation

Table 2.4 illustrates how the chatbot proceeds with **response planning and generation**. Based on the detected intent and emotion, the system decides whether to offer a predefined answer, generate a new one, or escalate the issue. The response is enriched to reflect the customer's emotional state for instance, calming language for angry users. Then, a tailored prompt is constructed for the language model (LLM), which generates a personalized message aimed at addressing the customer's concern effectively.

N°	Step Name	Example
5	Response Type Selection	The system determines whether to give a predefined answer, generate a custom one, or forward the user to a human agent.
6	Response Enrichment	The chatbot adjusts tone accordingly: for anger, it uses a respectful and calming tone: "We understand your frustration. Let's try to solve this together."
7	Prompt Construction	A prompt is composed for the LLM: "The customer is angry and wants to understand why the internet is slow."
8	LLM Generation	The language model generates a contextual response: "I'm truly sorry for the inconvenience. I'll guide you through a quick check to improve your connection."

Table 2.4: Chatbot interaction pipeline (Part 2/3): Response strategy and generation

2.6.3 Third step: Delivery, Feedback, and Learning

Finally, Table 2.5 shows how the chatbot **delivers the response and manages learning**. The message is sent through the appropriate interface, and user feedback is captured and analyzed. If the user confirms that the issue is resolved, this information is stored, and the full session (including intent, emotion, response, and feedback) is logged. This data becomes part of the system’s learning process, helping improve future interactions and make the chatbot smarter over time.

N°	Step Name	Example
9	Response Delivery	The response is delivered via the chatbot interface on the telecom provider’s app or website.
10	Feedback Capture	The customer replies: “Thanks, it’s better now,” which is saved for future analysis.
11	Session Logging	All elements of the interaction (intent, emotion, response, feedback) are logged for future analytics.
12	Learning	The session is added to the training data to fine-tune the system’s future behavior.

Table 2.5: Chatbot interaction pipeline (Part 3/3): Delivery, Feedback, and Learning

2.7 Conclusion

In this chapter, we introduced the foundations of our empathetic chatbot framework by presenting the underlying motivations, key challenges, and the importance of emotion and context awareness. We proposed a hybrid approach combining rule-based and machine learning techniques, detailed its architectural components, and emphasized user-centered UX design strategies. This comprehensive foundation paves the way for instantiating and validating our framework in real-world scenarios.

Chapter 3

SmartConnect Chatbot : Implementation

3.1 Introduction

This chapter presents the implementation of *SmartConnect*, an empathetic chatbot designed to enhance communication between public institutions and citizens. The system combines large language models (LLMs), rule-based logic, and modern AI frameworks to deliver context-aware and emotionally intelligent interactions. Throughout development, we adopted *user-centered approaches* to ensure that the chatbot reflects real user needs, motivations, and emotional cues. This chapter outlines the tools, technologies, and system architecture used, followed by the development steps, interface design,

3.2 Technologies and Tools Used

This section describes the main technologies, programming languages, frameworks, and libraries employed in the development of the intent-based empathetic chatbot system.

3.2.1 Programming Languages

- **Python:** The entire backend logic and model training were developed in Python, chosen for its rich support in machine learning and natural language processing (NLP).¹

3.2.2 Frameworks and Libraries

- **Transformers (Hugging Face):** Used to load and fine-tune pre-trained language models like BERT and FLAN-T5 for intent classification and empathetic response generation, also used to fine-tune the TinyLlama-1.1B-Chat-v1.0 model on a custom empathy dataset.²
- **PyTorch:** The deep learning framework used for model training and inference.³
- **scikit-learn:** Employed for traditional ML preprocessing tasks such as label encoding and data splitting.⁴

3.2.3 Dataset Handling

Data processing and handling were facilitated by the Pandas library, which provides easy-to-use data structures and tools for reading, manipulating, and preparing the textual intent dataset for model consumption. Additionally, the Hugging Face Datasets library was employed to load, pre-process, and format the custom empathetic dialogue dataset used for fine-tuning the TinyLlama model.

3.2.4 Model Architectures

- **BERT (Bidirectional Encoder Representations from Transformers):** A transformer-based model widely used for natural language understanding tasks. The base BERT model was fine-tuned for intent classification to detect user intents from input queries.
- **RoBERTa (Robustly optimized BERT approach):** Employed in the emotion detection module, the RoBERTa model fine-tuned on the GoEmotions dataset was used to classify the emotional tone of the user's input.

¹<https://www.python.org/>

²<https://huggingface.co/transformers>

³<https://pytorch.org/>

⁴<https://scikit-learn.org/>

- **FLAN-T5 (Text-to-Text Transfer Transformer):** This sequence-to-sequence model was used as a response enhancer to rewrite base chatbot replies empathetically according to the detected emotion, using linguistic markers of empathy and motivation.
- **TinyLlama** (via Hugging Face Transformers): Used as the main language model fine-tuned on a custom empathetic dataset. This replaced the previous multi-model approach with a single large language model capable of generating empathetic responses.⁵

3.2.5 Justification of Technology Choices

The selection of models, frameworks, and tools was guided by several important factors to ensure both performance and maintainability of the chatbot system:

- **Open-source and Active Community:** We choose models like BERT, RoBERTa, and TinyLlama primarily because they are open-source and supported by vibrant communities. This allows us to benefit from ongoing research improvements, shared knowledge, and troubleshooting support.
- **Robustness and Accuracy:** BERT and RoBERTa are proven state-of-the-art architectures for intent and emotion classification, respectively, providing robust understanding of natural language inputs.
- **Ease of Integration and Deployment:** Libraries like Hugging Face Transformers facilitate seamless integration and deployment of models.
- **Flexibility for Future Development:** The chosen tools provide extensibility for future model updates, data expansion, and feature addition without extensive re-engineering.

Overall, this technology stack balances innovation, practicality, and scalability, aligning with project requirements and constraints.

3.2.6 Development Environment

- **IDE:** The code was developed using **Visual Studio Code (VSCode)**, providing a versatile environment with debugging, Git integration, and extensions supporting Python and machine learning development.

3.3 System Architectures

3.3.1 RULE BASED MODEL

This section presents the overall architecture of the intent-based empathetic chatbot system. It describes the key components, their interactions, and how the system processes user input to generate context-aware, empathetic responses.

Overview

The system is designed as a modular pipeline composed of three main components:

- **Intent Recognition Module:** Classifies the user input into predefined intent categories.
- **Emotion Detection Module:** Analyzes the emotional tone of the user's input to identify their current feelings.

⁵<https://huggingface.co>

- **Response Generation and Enhancement Module:** Produces an initial response based on the detected intent and then refines it to match the user’s emotion, creating an empathetic and natural reply.

Data flow

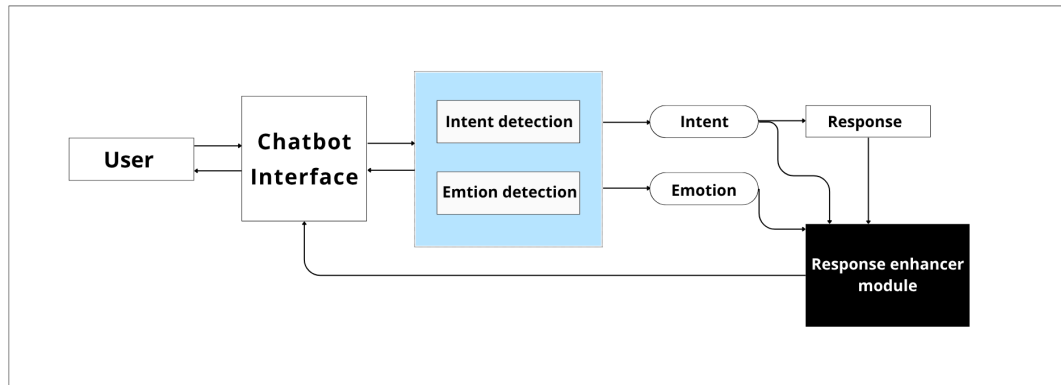


Figure 3.1: System Architecture Diagram of the Empathetic Chatbot

The processing flow is as follows:

1. The user inputs a text query through the chatbot interface.
2. The input text is passed to the **Intent Recognition Module**, which uses a fine-tuned BERT model to classify the intent.
3. Simultaneously, the input is processed by the **Emotion Detection Module** utilizing a RoBERTa-based model fine-tuned on emotion-labeled data to identify the emotional state of the user.
4. Based on the detected intent, a base response is selected from a predefined set of intent-specific replies.
5. The base response, user input, and detected emotion are fed into the **Response Enhancement Module**, which employs a FLAN-T5 model to rewrite the response empathetically, adapting the tone and style to the user’s emotional state.
6. The final enhanced response is then delivered back to the user via the chatbot interface.

Illustrative Example

User Query: “I’ve been trying to recharge my line for two days but it’s not working. I’m really frustrated.”

- **Intent Detected:** Recharge issue
- **Emotion Detected (planned):** Frustration

- **Generated Response:** “I’m really sorry you’ve been experiencing this issue for so long. I understand how frustrating that must be. Let me help you fix it right away.”

This example demonstrates how the system recognizes both the intent and emotional state, and tailors the response accordingly to maintain user trust and satisfaction.

Advantages of the Architecture

- **Modularity:** Each component can be developed, tested, and improved independently, enabling scalability and maintainability.
- **Flexibility:** The architecture allows easy integration of new models or replacement of existing components without affecting the overall system.
- **Empathy-Driven Interaction:** By combining intent detection with emotion analysis and response enhancement, the chatbot delivers more human-like and emotionally aware interactions, improving user satisfaction.

3.3.2 System Architecture(AI MODEL)

Overview

The core AI component of the system is a fine-tuned large language model specialized in empathetic response generation. The model is based on **TinyLlama-1.1B-Chat**, a lightweight transformer optimized for conversational tasks. The training data consisted of 644 user-emotion-response triples, each containing a user message, a labeled emotional state (e.g., *overwhelmed*, *appreciative*, etc.), and a corresponding empathetic response.

After training, the model demonstrated strong capability in generating emotionally aware and contextually appropriate responses to diverse user inputs. The final model will be integrated into the chatbot system to ensure real-time, emotionally aligned interactions with users.

Data Flow Pipeline

The architecture of the empathetic AI model follows a straightforward and efficient data processing pipeline.

Once the user message is entered, it is formatted into a prompt template that includes a system role and emotion annotation, such as:

```
<|system|> You are an empathetic assistant. <|user|> I'm feeling exhausted
and stressed. (feeling: exhausted)
```

This prompt is then passed to the fine-tuned **TinyLlama-1.1B-Chat-v1.0** model, which generates an empathetic response conditioned on both the user message and the emotion embedded within the prompt.

The model internally infers the emotional context and generates a contextually appropriate response in a single step. This design simplifies the architecture and leverages the model’s instruction-tuning capabilities to perform multi-role reasoning (understanding and responding) simultaneously.

The overall data flow can be illustrated as shown in Figure 3.2.

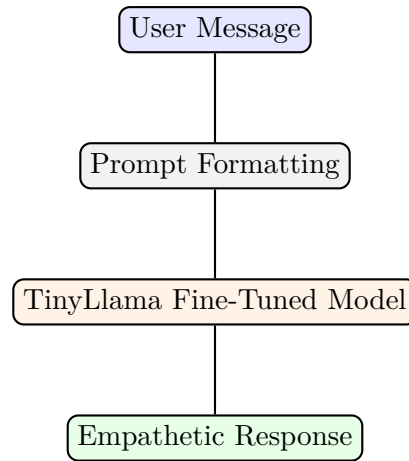


Figure 3.2: AI Model Data Flow in Development Environment

3.4 Dataset Preparation

3.4.1 Rule Based model

In this section we will dive deeper into the dataset preparation process see Figure 3.3

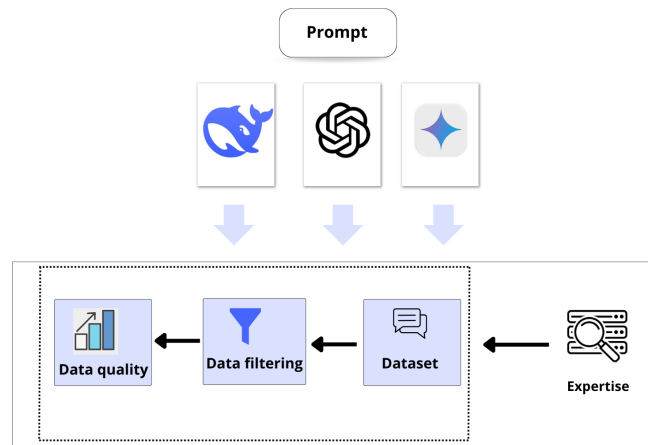


Figure 3.3: dataset building

Data Collection

The dataset used for intent detection was collected through extensive effort, as no publicly available datasets exist for telecommunication FAQs with empathetic responses, especially in the Algerian context. We gathered frequently asked questions primarily from social media platforms, where responses were not available. To complement this, we extracted additional data from official telecommunication websites.

The collected data was compiled into a `data.csv` file containing user queries (text) and their corresponding intent labels. The dataset includes 17 distinct intents, with a total of 423 labeled examples, balanced across intents. **Data Examples :** 3.1 shows some examples from the dataset, including user queries and their corresponding intent labels.

User Query	Intent
What is you number phone ?	support.contact number
which internet plans do you offer?	offers.view all
why route is not working	network.modem issues
Worst experience ever	complaints

Table 3.1: Sample user queries and their associated intents from the dataset.

Empathetic Response Generation

To generate empathetic responses for each identified intent, we leveraged advanced large language models including Gemini, DeepSeek, and ChatGPT. Using prompt engineering techniques, we carefully crafted prompts incorporating role prompting, chain-of-thought reasoning, and illustrative examples. This approach guided the models to produce responses that reflected empathy and understanding appropriate to the user’s input.

Each intent was paired with a corresponding empathetic response generated by the models. These responses were then reviewed by a psychologist with expertise in human emotions to assess the authenticity and quality of the empathetic aspects.

This expert validation ensured that the generated responses were not only syntactically correct but also genuinely conveyed human empathy, which is essential for enhancing user engagement and satisfaction in our system.

Data Preprocessing and Labeling

The dataset was preprocessed using tokenization and label encoding techniques to prepare for model training. We implemented a custom PyTorch `Dataset` class for efficient data handling, applying the BERT tokenizer with padding and truncation to a fixed maximum length.

Label encoding was performed to convert textual intent labels into numerical format. The data was split into training and validation sets with an 80/20 ratio.

Labeling was manual and carefully verified to ensure quality and consistency.

3.4.2 LLM-Assisted Empathetic Data Creation

As described in the Rule-Based Model section (see Figure 3.3), we created a dataset by combining user messages, emotions, and empathetic responses. Since no publicly available empathetic dataset exists for the Algerian telecom context, we designed a semi-automated data generation pipeline using large language models (LLMs) such as ChatGPT, Gemini, and DeepSeek.

Prompt Design for Dataset Generation

To generate high-quality data suitable for fine-tuning, we crafted a system prompt that instructed the LLM to simulate realistic telecom user queries, label them with appropriate emotions, and respond empathetically.

Prompt Template:

LLM Prompt for Empathetic Dataset Generation

Generate a fine-tuning dataset for a large language model (LLM) specialized in responding empathetically to telecommunications users.

Each dataset example must have three fields:

"user_message": a realistic message from a telecom user.

"emotion": the user's dominant emotion (choose from: neutral, happy, curious, confused, frustrated, angry, sad, anxious, hopeful, appreciative).

"empathetic_response": an empathetic, motivational, and supportive reply.

Requirements: - Simulate a variety of emotional intensities (low, medium, high). - Messages must be natural and typical for telecom users. -

Responses must validate the user's emotions and show genuine empathy. -

Integrate motivational theories where appropriate: * Maslow's Hierarchy of Needs (belonging, esteem, security) * Self-Determination Theory (autonomy, competence, relatedness) * Herzberg's Two-Factor Theory (satisfiers and hygiene factors) - Responses must maintain clarity, warmth, and professionalism. - Encourage positive emotional outcomes (hope, trust, belonging).

Output format: Generate as a list of JSON objects:

```
{
  "user_message": "string",
  "emotion": "neutral | happy | curious | confused | frustrated | angry | sad |
anxious | hopeful | appreciative",
  "empathetic_response": "string"
}
```

Generate at least 100 examples with a diverse emotional distribution.

Expert Review and Filtering

Generated examples were reviewed by a psychologist specialized in emotional communication. The expert filtered out emotionally inconsistent or generic responses and ensured psychological appropriateness and authenticity.

Use in the LLM Fine-Tuning

The dataset was used for supervised fine-tuning of TinyLlama, a lightweight open-source LLM. Each (user_message, emotion, empathetic_response) tuple trained the model to recognize emotional cues and respond supportively. This helped the model generalize across a range of telecom user experiences and emotional intensities.

3.5 Model Implementation

3.5.1 Rule based model

This section describes the development and implementation of the core components of our chatbot system, which include the intent detection model, the planned emotion detection module, and the response enhancer module used for empathetic reply generation.

Dataset Loading and Preprocessing

The dataset used was stored in a `data.csv` file containing two columns: `text` (user queries) and `intent` (corresponding intent labels). We first loaded the dataset using the `pandas` library. The intent labels were converted to numeric values using `LabelEncoder`. We then split the dataset into training and validation sets in an 80/20 ratio to prepare for model training.

Model Selection

We used the `bert-base-uncased` model from Hugging Face's `transformers` library, which is pre-trained and well-suited for sequence classification tasks. The model was fine-tuned on our dataset with the number of output classes set to 17, corresponding to the number of unique intents.

Custom Dataset Class

A custom `PyTorch Dataset` class was implemented to handle the input preparation. Each text sample was tokenized using the `BertTokenizer`, with padding and truncation applied to a maximum length of 64 tokens. The dataset returns input IDs, attention masks, and encoded labels.

Training Setup

The training was conducted using Hugging Face's `Trainer` API, which simplifies the process of fine-tuning transformer models. Training arguments included the output directory, batch sizes of 8, number of epochs set to 5, and an evaluation strategy based on epoch intervals. Logging and model checkpoints were also handled by the API.

Model Saving

After training, the model was saved to disk along with the tokenizer and the label encoder to facilitate future inference.

```
model.save_pretrained("model/classifier")
tokenizer.save_pretrained("model/tokenizer")
with open("model/label_encoder.pkl", "wb") as f:
    import pickle
    pickle.dump(le, f)
```

Listing 3.1: Model Training and Saving

Emotion Detection Module

While our system was initially centered on intent detection, we have successfully integrated an emotion detection module to further enhance the empathy and relevance of chatbot responses. This module leverages a transformer-based model currently implemented using RoBERTa-base fine-tuned on emotion classification datasets to identify the user's underlying emotional state.

It processes user inputs to detect emotions such as anger, frustration, sadness, or confusion. The identified emotion is then used to influence both the tone and content of the chatbot's reply. By incorporating this emotional layer, our system ensures that responses are not only contextually accurate but also emotionally attuned to the user's state, thereby delivering a more human-like and supportive interaction experience.

Response Enhancer Module

To enhance the quality of responses, especially in terms of empathy, we utilized large language models (LLMs) such as Gemini, DeepSeek, and ChatGPT. We applied prompt engineering techniques to generate empathetic replies for each intent in our dataset.

Our prompt design strategy included:

- **Role prompting:** Assigning the LLM the role of a helpful telecom assistant.
- **Chain-of-thought reasoning:** Guiding the model to reflect on the user's possible emotional state.

- **Few-shot examples:** Including sample queries and responses to enhance consistency.

Each intent was paired with a corresponding empathetic response. To ensure psychological validity, these responses were reviewed by a psychological expert who evaluated whether the replies contained essential human aspects like empathy and understanding. Additionally, feedback was provided regarding the potential impact of these responses on user satisfaction and emotional well-being.

3.6 Model Fine-Tuning

For the task of developing an empathetic chatbot, we fine-tuned the `TinyLlama/TinyLlama-1.1B-Chat-v1.0` model. This model is used to handle emotional responses.

3.6.1 Data Loading and Preprocessing

We began by loading our conversational dataset, stored in JSON format, which contains user messages along with the associated emotional labels and empathetic assistant responses.

```
with open("data/empathetic_data.json", "r", encoding="utf-8") as f:
    raw_data = json.load(f)
```

Listing 3.2: Loading raw data from JSON file

Each example was then formatted to match the expected prompt-response structure of the TinyLlama chat model. Specifically, the data was wrapped using special tokens indicating the system, user, and assistant messages, as shown below:

```
def format_example(example):
    return {
        "text": f"<|system|>\nYou are an empathetic assistant.\n<|user|>\n{
example['user_message']} (feeling: {example['emotion']})\n<|assistant|>\n{
example['empathetic_response']}"
    }
```

Listing 3.3: Formatting data to prompt style

3.6.2 Tokenizer and Model Initialization

The pretrained tokenizer and model were loaded from the HuggingFace model hub. The tokenizer's padding token was set to the end-of-sequence token to maintain consistency during training.

```
model_name = "TinyLlama/TinyLlama-1.1B-Chat-v1.0"
tokenizer = AutoTokenizer.from_pretrained(model_name)
tokenizer.pad_token = tokenizer.eos_token

model = AutoModelForCausalLM.from_pretrained(model_name)
```

Listing 3.4: Loading pretrained tokenizer and model

3.6.3 Tokenization of Dataset

The formatted dataset was tokenized with truncation and padding to a maximum sequence length of 512 tokens. This process converts textual data into token IDs and attention masks required for model training.

```
def tokenize(example):
    return tokenizer(example["text"], truncation=True, padding="max_length",
max_length=512)
```

```
tokenized_dataset = dataset.map(tokenize, batched=True)
tokenized_dataset.set_format(type="torch", columns=["input_ids", "attention_mask"])
```

Listing 3.5: Tokenizing dataset examples

3.6.4 Training Setup

Training parameters were configured using HuggingFace's `TrainingArguments`. Due to hardware constraints, a batch size of 1 was used. The model was trained for 3 epochs, with logging and checkpoint saving enabled for monitoring progress.

```
training_args = TrainingArguments(
    output_dir="tinyllama-finetuned",
    per_device_train_batch_size=1,
    num_train_epochs=3,
    logging_steps=10,
    save_steps=100,
    save_total_limit=2,
    fp16=False,
    logging_dir="./logs",
    report_to="none",
    remove_unused_columns=False
)
```

Listing 3.6: Configuring training arguments

3.6.5 Fine-Tuning Procedure

The HuggingFace `Trainer` API was used to streamline the fine-tuning process, handling batching, optimization, and checkpointing.

```
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_dataset,
    tokenizer=tokenizer,
    data_collator=DataCollatorForLanguageModeling(tokenizer, mlm=False)
)

trainer.train()
```

Listing 3.7: Training the model using HuggingFace Trainer

3.6.6 Model Saving

After training completed, the fine-tuned model and tokenizer were saved locally for future inference and deployment.

```
trainer.save_model("tinyllama-finetuned")
tokenizer.save_pretrained("tinyllama-finetuned")
```

Listing 3.8: Saving the fine-tuned model and tokenizer

This section describes the core implementation steps taken to adapt the TinyLlama chat model to an empathetic conversational agent via supervised fine-tuning on domain-specific data.

3.7 UI/UX Design

3.7.1 Objective of UI/UX

The UI/UX design aims to provide a smooth, intuitive, and empathetic experience. It focuses on clarity, simplicity, and personalized interactions for users, while giving administrators easy access to analytics and essential tools.

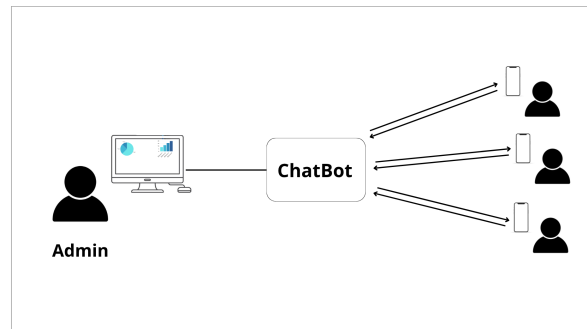


Figure 3.4: Intermediate solution between customers and stakeholders

3.7.2 Tools Used

The interfaces were designed using FIGMA⁶

3.7.3 Wireframes

Below are some of the wireframes that illustrate the early-stage design of the application.



Figure 3.5: Wirframe of our application

⁶<https://www.figma.com/>

3.7.4 Admin Dashboard (UI)

The admin dashboard is designed to allow administrators to monitor key analytics and manage content efficiently.

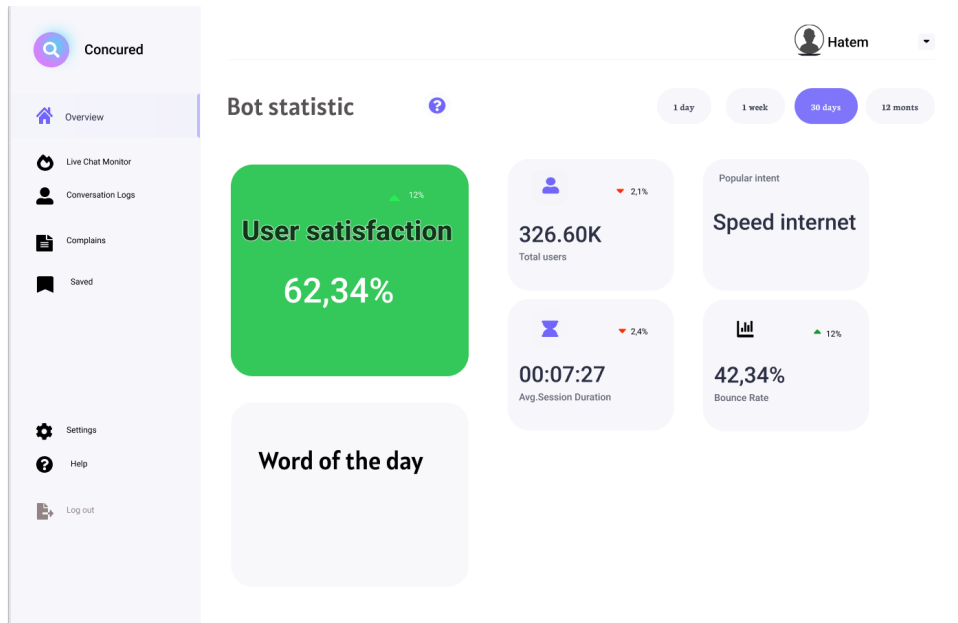


Figure 3.6: Admin dashbord

3.7.5 User Experience

The user experience was designed with empathy and personalization at its core. This is reflected in the interface's simplicity, fluidity, and intuitive structure, allowing users to interact effortlessly. Beyond clear and instant chatbot communication, the app includes thoughtful features such as emoji-based satisfaction feedback, easy complaint/report submission, motivational messages, and real-time tracking of submitted complaints all contributing to a transparent and supportive experience.

The application also fosters a strong sense of community. Users are not alone, they can participate in shared quizzes, earn bonuses, and view messages from others reinforcing that they're part of a connected and encouraging environment.

3.8 Challenges and Solutions

3.8.1 Model Size and Computation Cost

Transformers like BERT and RoBERTa are computationally expensive, especially during fine-tuning.

3.8.2 Prompt Design for Empathy

.Crafting prompts that consistently generate empathetic responses across various LLMs required extensive experimentation. We found that few-shot prompting combined with role-setting significantly improved the quality of responses.

3.8.3 Balancing Response Accuracy and Warmth

In some cases, model-generated responses were technically correct but emotionally cold. We solved this by post-processing responses and incorporating expert review to ensure alignment with empathetic standards.

In this chapter, we detailed the implementation of the core components of our empathetic chatbot system:

- Fine-tuned a BERT model for intent classification on a domain-specific dataset.
- Designed an (upcoming) emotion detection module based on transformer models to capture user sentiment.
- Integrated large language models with tailored prompts to enhance chatbot responses.
- Ensured psychological quality of responses through expert evaluation.

This implementation lays the foundation for the system’s evaluation, which we discuss in the following chapter.

3.9 Test and Evaluation

We tested the chatbot’s performance in two key areas: intent recognition and empathetic response.

Table 3.2 shows test questions used to assess the chatbot’s ability to understand user intents. Each row includes:

Test Question	Expected Intent	Predicted Intent	Correct (/)
Hello there!	greeting	greeting	1
Bye, see you later	goodbye	goodbye	1
Thank you for the help	thanks	thanks	1
When will fiber be available?	ask.fiber	ask.fiber	1
I want to pay my outstanding balance immediately. What are my payment options and steps?	billing.payments	billing.payments	1
My internet keeps disconnecting every hour, I want to file a complaint and get support.	complaints	network.connection _i ssues	0
Please provide a detailed list of all current internet offers including data limits, prices, and validity periods.	offers.view _{all}	offers.view _{all}	1
Why is my internet speed so slow even though my plan is unlimited? What can I do?	network.connection _i ssues	network.connection _i ssues	1
Can you tell me the process and contact information to escalate my unresolved technical issues to a human agent?	escalation	support.contact _i nfo	0

Table 3.2: Test Questions with Expected and Predicted Intents

Result :

- **Test Question:** The user’s input.
- **Expected Intent:** What the chatbot should identify.
- **Predicted Intent:** What the chatbot actually identified.

- **Correct (/):** Whether the chatbot was correct.

Key points:

- The chatbot correctly identified most off all intents in these test cases.
- It works well with different types of questions, from simple greetings to detailed payment or complaint requests.
- This shows the chatbot is reliable for handling common customer questions.

User Question	Response	Empathy Score (1-5)
My internet is not working and I need it urgently for a meeting. Please help!	Sorry for the inconvenience. Please restart your router and if the issue continues give me more details.	2
I've already tried restarting the router twice. It's still not working.	I'm sorry to hear that. Please restart the router and if the problem persists, provide more details and let's solve it together.	3
Why do I pay every month if this service keeps failing?	I understand how frustrating this can be. Please be sure that we are doing our best for a better experience. Share more about your problem and we'll find a solution together.	4
You guys never fix anything. I'm tired of complaining.	I understand how frustrating this can be. Please be sure that we are doing our best for a better experience.	3
I am so happy by the new offers!	Stay connected and empowered with these amazing internet offers! IDOOM 4G LTE: Get 150 GB for just 4500 DA – perfect for streaming, studying, or working online! (Valid for 30 days)	2

Table 3.3: Empathy Evaluation of Responses

To evaluate the empathetic quality of chatbot responses, we designed a second test using emotionally charged user questions including urgency, frustration, and satisfaction. We then analyzed the chatbot's responses to see how well they acknowledged user emotions and needs.

Using ChatGPT's advanced language capabilities, we assigned an empathy score from 1 (low empathy) to 5 (high empathy) to each response. These scores helped us identify whether the chatbot can respond in a human-like and emotionally aware manner, which is critical for enhancing user satisfaction and trust.[3.3](#)

3.10 Conclusion

This chapter presented the complete implementation process of the application. It covered the selection of appropriate technologies and frameworks, the integration of the language model, and the development of key features using Flutter. The UI/UX design and wireframes were carefully created to deliver a fluid and user-friendly experience. Additionally, a simple chatbot interface and an admin dashboard for analytics were implemented. This phase established a strong and functional base for the application, ready to support future enhancements.

General Conclusion

Conclusion

This thesis presented a rule-based, context-aware chatbot designed to improve how companies, especially in the telecom sector, handle client questions. By combining the client's profile, the context of the request, and the question itself, the system is able to better understand the user's intent and give more accurate and helpful answers.

The chatbot works automatically and continuously (24/7), helping both clients and staff. It reduces the time employees spend on repeated questions and improves overall client satisfaction. A dashboard also helps staff monitor and learn from real interactions, giving insight into common issues and needs.

The solution was tested in a real telecom company in Algeria and showed good results in saving time, improving support quality, and enhancing the client experience.

Future Work

This work opens the door to several future improvements:

- **Natural Language Understanding (NLU):** The current chatbot is rule-based. In the future, machine learning or deep learning techniques can be added to help understand more complex questions.
- **Personalization:** Future versions can make responses even more personalized by learning from past interactions and user feedback.
- **Multilingual Support:** Adding support for other languages, such as Tamazight or English, can help the chatbot serve a wider range of users.
- **Voice Interaction:** Integrating voice-based communication would make the chatbot more accessible and user-friendly.
- **Integration with Other Systems:** The chatbot can be connected to other company services (CRM, billing, etc.) to provide more complete and dynamic answers.

These improvements can make the chatbot smarter, more flexible, and more useful in different real-world situations.

Annexes

البطاقة التقنية للمشروع

Hatem Fatima Zahra	الاسم و اللقب Votre prénom et nom Your first and last Name
SmartConnect Algeria	الاسم التجاري للمشروع Intitulé de votre projet Title of your Project
	الصفة القانونية للمشروع Votre statut juridique Your legal status
0776961732	رقم الهاتف Votre numéro de téléphone Your phone number
hatemfatimazahra@gmail.com	البريد الالكتروني Votre adresse e-mail Your email address
Tiaret - Tiaret	مقر مزاولة النشاط (الولاية - البلدية) Votre ville ou commune d'activité Your city or municipality of activity

البطاقة التقنية للمشروع

--	--

القيمة المقترحة أو العرض المقدم

Value Proposition

تحديد المشكل الذي يواجهه الزبون

• ضعف التواصل مع الزبائن • تأخير الردود أو غيابها التام • صعوبة وصول الزبائن الى المعلومات	• عدم توفر الدعم على مدار الساعة • تجارب سلبية مع الدعم الفني (الاحباط بسبب عدم الفهم أو عدم الاحترام أو عدم الكفاءة) • فقدان الثقة • تكلفة عالية لتشغيل الدعم
ما هي المشكلة التي تريد حلها؟	
ما هي البيانات المتوفرة لديك التي تدل على وجود المشكلة المحددة؟	
ما هي المشاريع الأخرى التي استهدفت نفس المشكلة والتي جرى تنفيذها؟	
ماهي أهداف مشروعك و/أو نتائج المتوقعة؟	أهداف المشروع • تحسين وتخصيص تجربة العملاء • تقديم دعم فوري، انساني ومتعاطف • الاستماع الدائم والنشاط لاستعلامات وشكاوى الزبائن • تحسين جودة التواصل بين المؤسسات والزبائن • تمكين المؤسسات من تقديم دعم على مدار الساعة (24/7) • إتاحة الوصول السهل للمعلومات • تحسين رفاهية المواطنين • تحسين جودة تقديم الخدمة النتائج المتوقعة: • تحقيق رضا وثقة ولاء العملاء بالمؤسسة • تعزيز الصورة الرقمية للمؤسسات وسمعتها • خفض التكاليف التشغيلية • رفع فعالية فريق الدعم البشري • زيادة الإيرادات • جذب فرص شراكة مع مؤسسات عمومية و خاصة • في قطاعات مختلفة • دعم التحول الرقمي في الجزائر

القيمة المقترحة أو العرض المقدم

Value Proposition

القيمة المقترحة وفق المعايير التالية

القيمة المبتكرة أو الجديدة	دمج الذكاء العاطفي في روبوت محادثة ذكي و إنساني، بحيث لا يقتضي فقط يفهم استفسارات المستخدم، بل يتعرف على مشاعره ويقدم ردودا متعاطفة، يدعم اللغة العربية.
القيمة بالتخصيص	توفير شات بوت محلي، موجه خصيصا لحل مشاكل المواطن الجزائري والمؤسسة الجزائرية بطريقة معنوية وسريعة وعاطفية. تخصيص chatbot وفقا لاحتياجات كل مؤسسة من خلال إعداد قاعدة بيانات خاصة بها وتخصيص الحوار حسب طبيعة كل مستخدم
القيمة بالسعر	يوفر المشروع المقترح قيمة سعرية عالية، حيث يملئ المؤسسات أداة ذكية وعالية الجودة لتحسين تجربة المستخدمين وإدارة وتحليل بيانات التفاعل معهم كما يوفر حلا اقتصاديا مقارنة بتكاليف توظيف فريق دعم بشري كامل، كما أنه يقلل من الأخطاء البشرية ويحافظ على جودة الردود دون تكلفة إضافية على المدى الطويل.
القيمة بالتصميم	يتميز المشروع بتصميم ذكي وموثق، حيث يستفيد المستخدم من تطبيق محمول سهل الاستخدام وسلس في التنصف، في حين يتمكّن المسؤولون من خلال واجهة مكتب إدارية من متابعة التحليلات والتفاعلات لحظة بلحظة، مما يجعل التجربة فعالة، واضحة، وذات جاذبية بصرية عالية
القيمة بالأداء العالي	يوفر الروبوت أداء متفوقا من حيث السرعة، الدقة، والموثوقية، بالإضافة إلى فهم عاطفي للمستخدم، مما يتيح تواجدا فعالا دون انقطاع وعلى مدار الساعة. ويضم تجربة استخدام مستمرة وخالية من التأخير أو الأخطاء الناتجة عن التعب البشري
القيمة بالخدمة الشاملة	الروبوت يُقدم تجربة دعم ذكية ومتكاملة تعتمد على الفهم العاطفي والتفاعل الفوري مع المستخدمين كما يُزوّد الشركات بتحليلات دقيقة لسلوك العملاء واهتماماتهم، ما يُساعد في تحسين الخدمات، تطوير المنتجات، وزيادة رضا الزبائن وولائهم. تعزيز جودة العلاقة بين المؤسسة والعملاء
قيم أخرى	القيمة بالتكامل (Integration Value) يمكن ربط الروبوت مع أنظمة إدارة العملاء (CRM) وقواعد البيانات القيمة بالاستدامة (Sustainability Value) اعتماد روبوت محادثة ذكي يُقلل من استهلاك الموارد البشرية والورقية، ما يساهم في تحسين كفاءة المؤسسة ودعم مبادرات التحول الرقمي والاستدامة

Customer Segments أو الزبائن العملاء شرائح

Géographique الجغرافية	Démographique (B2C)	Démographique (B2B)	Psychographique العوامل النفسية و الشخصية	Comportemental السلوكيات
Continent القارة	Age العمر	Secteur القطاع	Classe sociale لطبقة الاجتماعية	Usage استخدام
Pays الدولة	Sexe الجنس	Nombre d'employés عدد العمال في القطاع	Niveau de vie المستوى المعيشي	Loyauté الوفاء
Région الجهة	Revenus annuel متوسط الدخل	Maturité de l'entreprise نضج المؤسسة	Valeurs القيم	Intérêt اهتمام
Département الولاية	Etat matrimonial الحالة الاجتماعية	Situation financière الحالة المالية للمؤسسة	Personnalité الشخصية	Passion الهواية و شغف
Ville البلدية أو البلدية	Niveau d'étude المستوى الدراسي	Détention/ actionariat الملكية/ المساهمة	Convictions المعتقدات	Sensibilité حساسيات
Quartier حي	Profession المهنة	Valorisation/ capitalisation boursière التقييم / القيمة السوقية	Présence digitale et sur les réseaux sociaux استعمال التكنولوجيا في التواصل	Habitude de consommation عادة الاستهلاك
Climat المناخ	Culture الثقافة	Business model نموذج الأعمال	Centres d'intérêts مراكز الاهتمام	Mode de paiement طرق الدفع
	Religion الدين	Secteur servi القطاع الذي يخدمه		Connaissance المعرفة
	Langue اللغة	Technologie utilisée التكنولوجيا المستخدمة		Nature de la demande طبيعة الطلب
		Format du produit ou packaging شكل المنتج أو التعبئة والتغليف		Fréquence d'achat عدد مرات الطلب على السلعة

قنوات التوزيع CHANNELS

بيع الخدمة مباشرة إلى الشركات (B2B) والمؤسسات الحكومية (B2G), من خلال تقديم خدمة تطوير روبوت محادثة ذكي مخصص حسب مجال ونشاط كل شركة أو مؤسسة.	المبيعات المباشرة
	تجار الجملة
شراكات مع شركات خدمات رقمية أو وكالات تسويق رقمي	الموزعون
مشاركة في معارض الابتكار والمؤتمرات	توزيع التجزئة

العلاقة مع العملاء Customer Relationship

تولي أهمية قصوى لبناء علاقات مثبته وطويلة الأمد مع عملائنا من خلال: - الواصل المستمر والمخصص: - نعمد على تواصل مباشر وشخصي مع كل عميل لفهم احتياجاته وتطلعاته، وضمان تطوير حلول تتماشى مع نشاطه وسياقه. - الدعم الفني المتواصل: - نوفر دعماً تقنياً فورياً قبل، أثناء، وبعد إطلاق الروبوت لضمان حسن الأداء والاستجابة السريعة لأي طارئ. - تحديثات وتطويرات دورية، نحرص على تحسين وتحديث روبوتات المحادثة بشكل مستمر - التفاعل المستند إلى البيانات: - نعمد على تحليلات سلوك المستخدمين والتغذية الراجعة لتحسين أداء الروبوت وتقديم توصيات للتعلم. - شراكة استراتيجية: - نعمل مع عملائنا كشركاء، ونسعى إلى خلق قيمة حقيقية تساهم في تطوير أعمالهم وتحقيق أهدافهم.	كيف تحير علاقاتك مع العملاء؟
Monday CRM: لإدارة الصفقات، وتنظيم مراحل التواصل مع العملاء ضمن سير عمل مرن وسهل التتبع	ماهية أهم البرامج التي ستعتمد عليها في إدارة العلاقة مع الزبونMicrosoft Dynamics Monday CRM Zoho CRM

الشركاء الأساسيون Key Partners

الشركاء	معلومات حول الشركاء	طبيعة الشراكة
الشريك الأول		-
الشريك الثاني	- الشريك ليس منافس و إنما هي عملية ترتيب متبادل المنفعة - يكون لنا معه مصلحة مشتركة في تطوير منتجات جديدة - إستراتيجية مصممة لتقليل المخاطر ، والتي قد ترتبط بإدخار منتج جديد إلى السوق - علاقتنا مع الشريك هي علاقات بين المشتري والمورد	
الشريك الثالث	- الشريك ليس منافس و إنما هي عملية ترتيب متبادل المنفعة - يكون لنا معه مصلحة مشتركة في تطوير منتجات جديدة - إستراتيجية مصممة لتقليل المخاطر ، والتي قد ترتبط بإدخار منتج جديد إلى السوق - علاقتنا مع الشريك هي علاقات بين المشتري والمورد	

هيكل التكاليف structure Costs

43500 DA	تكاليف التعريف بالمنتج أو المؤسسة d'établissement
10,000 DA	تكاليف الحصول على العدادات (الماء- الكهرباء) Frais d'ouverture de compteurs (eaux-gaz-....)
200,000 DA	تكاليف (التكوين- برامج الاعلام الالي المختصة) Logiciels, formations
50,000 - 100,000 DA	Dépôt marque, brevet, modèle تكاليف براءة الاختراع و الحماية الصناعية و التجارية
100,000 DA - 300,000 DA	Droits d'entrée تكاليف الحصول على تكنولوجيا أو ترخيص استعمالها
100,000 DA - 300,000 DA	Achat fonds de commerce ou parts شراء الأصول التجارية أو الأسهم
30,000 -45,000 DA	Droit au bail الحق في الإيجار
15,000 DA	Frais de dossier رسوم إيداع الملفات
40,000 DA	Frais de notaire ou d'avocatتكاليف الموثق- المحامي
100,000 DA	Enseigne et éléments de communication تكاليف التعريف بالعلامة و تكاليف قنوات الاتصال
	Achat immobilier شراء العقارات
40,000 DA	الأعمال والتحسينات الاماكن Travaux et aménagements

هيكل التكاليف structure Costs

100,000 DA	Matériel الألات- المركبات- الاجهزة
50,000 DA	Matériel de bureau تجهيزات المكتب
100,000 DA	Stock de matières et produits تكاليف التخزين
1000,000 DA	Trésorerie de départ التدفق النقدي (الصندوق) الذي تحتاجه في بداية المشروع

المجموع:

نفقاتك أو التكاليف الثابتة الخاصة بمشروعك

15 000 - 30 000 DA	Assurances التأمينات
5 000 - 15 000 DA	Téléphone, internet الهاتف و الانترنت
5 000 - 15 000 DA	Autres abonnements اشتراكات أخرى
	Carburant, transports الوقود و تكاليف النقل
	Frais de déplacement et hébergement تكاليف التنقل و المبيت
5 000 - 15 000 DA	Eau, électricité, gaz فواتير الماء - الكهرباء - الغاز
	Mutuelle التعاضدية الاجتماعية
40000 DA	Fournitures diverses لوازم متنوعة
	Entretien matériel et vêtements صيانة المعدات والملابس
	Nettoyage des locaux تنظيف المباني
5 000 - 15 000 DA	Budget publicité et communication ميزانية الإعلان والاتصالات
40000 DA	

المجموع:

**مصادر الإيرادات
Revenue Stream**

	Apport personnel ou familial المساهمة الشخصية أو العائلية
	Apports en nature (en valeur) التبرعات العينية
	Prêt n° 1 (nom de la banque) قرض رقم 1 اسم البنك
	Prêt n° 2 (nom de la banque) قرض رقم 2 اسم البنك
	Prêt n° 3 (nom de la banque) قرض رقم 3 اسم البنك
	Subvention n° 1 (libellé) منحة 1
	Subvention n° 2 (libellé) منحة 2
	Autre financement (libellé) تمويل آخر

المجموع:

رقم الأعمال

Votre chiffre d'affaires de la première année

بيع المنتج في السنة الأولى

متوسط أيام العمل في الشهر	بيع المنتج في السنة الأولى
20	الشهر 1Mois
20	الشهر 2Mois
20	الشهر 3Mois
20	الشهر 4Mois
20	الشهر 5Mois
20	الشهر 6Mois
20	الشهر 7Mois
20	الشهر 8Mois
20	الشهر 9Mois
20	الشهر 10Mois
20	الشهر 11Mois
20	الشهر 12Mois

المجموع:

رقم الأعمال

Votre chiffre d'affaires de la deuxième année

بيع المنتج في السنة الثانية

متوسط أيام العمل في الشهر	بيع المنتج في السنة الثانية
20	الشهر 1Mois
20	الشهر 2Mois
20	الشهر 3Mois
20	الشهر 4Mois
20	الشهر 5Mois
20	الشهر 6Mois
20	الشهر 7Mois
20	الشهر 8Mois
20	الشهر 9Mois
20	الشهر 10Mois
20	الشهر 11Mois
20	الشهر 12Mois

المجموع:

رقم الأعمال

بيع المنتج في السنة الثالثة

Votre chiffre d'affaires de la troisième année

متوسط أيام العمل في الشهر	بيع المنتج في السنة الثالثة
20	الشهر 1Mois
20	الشهر 2Mois
20	الشهر 3Mois
20	الشهر 4Mois
20	الشهر 5Mois
20	الشهر 6Mois
20	الشهر 7Mois
20	الشهر 8Mois
20	الشهر 9Mois
20	الشهر 10Mois
20	الشهر 11Mois
20	الشهر 12Mois

المجموع:

تطور حجم رقم الأعمال في السنة

- النسبة المئوية للزيادة في حجم الأعمال بين السنة 1 والسنة 2؟
- النسبة المئوية للزيادة في حجم الأعمال بين السنة 2 والسنة 3؟

حاجتك لرأس المال العامل

يوم 30	متوسط مدة الاعتمادات الممنوحة للعملاء بالأيام Durée moyenne des crédits accordés aux clients en jours
يوم 30	متوسط مدة ديون الموردين بالأيام Durée moyenne des dettes fournisseurs en jours

رواتب الموظفين و مسؤولين الشركة

50000 DA	رواتب الموظفين Salaires employés
70000 DA	صافي أجور المسؤولين Rémunération nette dirigeant

Business Model Canvas				
Designed for:		Designed by:		Date:
Version:				
Key Partners Who are our Key Partners? Who are our key suppliers? Which Key Resources are we acquiring from partners? Which Key Activities do partners perform? MOTIVATIONS FOR PARTNERSHIPS: Optimization and economy, Reduction of risk and uncertainty, Acquisition of particular resources and activities	Key Activities What Key Activities do our Value Propositions require? Our Distribution Channels? Customer Relationships? Revenue streams? CATEGORIES: Production, Problem Solving, Platform/Network Key Resources What Key Resources do our Value Propositions require? Our Distribution Channels? Customer Relationships? Revenue Streams? TYPES OF RESOURCES: Physical, Intellectual (brand patents, copyrights, data), Human, Financial	Value Propositions What value do we deliver to the customer? Which one of our products and services are we offering to each Customer Segment? Which customer needs are we satisfying? CHARACTERISTICS: Newness, Performance, Customization, "Getting the Job Done", Design, Brand/Status, Price, Cost Reduction, Risk Reduction, Accessibility, Convenience/Usability	Customer Relationships What type of relationship does each of our Customer Segments expect us to establish and maintain with them? Which ones have we established? How are they integrated with the rest of our business model? How costly are they? Channels Through which Channels do our Customer Segments want to be reached? How are we reaching them now? How are our Channels integrated? Which ones work best? Which ones are most cost-efficient? How are we integrating them with customer routines?	Customer Segments For whom are we creating value? Who are our most important customers? Is our customer base a Mass Market, Niche Market, Segmented, Diversified, Multi-sided Platform
Cost Structure What are the most important costs inherent in our business model? Which Key Resources are most expensive? Which Key Activities are most expensive? IS YOUR BUSINESS MORE: Cost Driven (leanest cost structure, low price value proposition, maximum automation, extensive outsourcing), Value Driven (focused on value creation, premium value proposition). SAMPLE CHARACTERISTICS: Fixed Costs (salaries, rents, utilities), Variable costs, Economies of scale, Economies of scope		Revenue Streams For what value are our customers really willing to pay? For what do they currently pay? How are they currently paying? How would they prefer to pay? How much does each Revenue Stream contribute to overall revenues? TYPES: Asset sale, Usage fee, Subscription Fees, Lending/Renting/Leasing, Licensing, Brokerage fees, Advertising FIXED PRICING: List Price, Product feature dependent, Customer segment dependent, Volume dependent DYNAMIC PRICING: Negotiation (bargaining), Yield Management, Real-time-Market		

Business Model Canvas				
Designed for:		Designed by:		Date:
Version:				
Key Partners Puplic Instituions Universities Companies Assciations	Key Activities Active listening 24/7 Deliver empathetic context aware responses Personalize interactions Enhance communication Rebuild trust Reduce operational costs Deliver analytics fr administrators Key Resources NLP and LLM Collecting database Developers and AI engineers Server infrastructure for hosting	Value Propositions Integrating emotional intelligence into a smart, human-like chatbot that not only understands user queries, but also recognizes their emotions and provides empathetic responses. Supports the Arabic language. Providing a local chatbot specifically designed to solve the problems of Algerian citizens and Algerian institutions in a professional, rapid, and compassionate manner. Customizing the chatbot to the needs of each institution by creating its own database and tailoring the dialogue to the nature of each user.	Customer Relationships Automated support through the chatbot Feedback mechanisms for continuous improvement Channels Mble application Web interface	Customer Segments Citizens Students Serve providers Cmpanies
Cost Structure Development and maintenance costs Server and hosting expenses Marketing and promotional costs Employee salaries and training		Revenue Streams Selling the mobile application Subscription model for administrators Premium models to citizen		

Bibliography

Bibliography

- [1] C. Waghmare and C. Waghmare, “Deploy chatbots in your business,” *Introducing Azure Bot Service: Building Bots for Business*, pp. 31–60, 2019 (cit. on p. 14).
- [2] M. D. Delis, I. Hasan, and M. Iosifidi, “On the effect of business and economic university education on political ideology: An empirical note,” *Journal of Business Ethics*, vol. 155, pp. 809–822, 2019 (cit. on p. 14).
- [3] M. Gunarto, P. Purwanto, D. Amanah, and D. A. Harahap, “Creating student loyalty through the value of context-based customer education,” *Management Scientific Journal*, vol. 12, no. 1, pp. 14–30, 2022 (cit. on p. 14).
- [4] S. Henne, V. Mehlin, E. Schmid, and S. Schacht, “Components of digital assistants in higher education environments,” in *2022 IEEE 28th International Conference on Engineering, Technology and Innovation (ICE/ITMC) & 31st International Association For Management of Technology (IAMOT) Joint Conference*, IEEE, 2022, pp. 1–8 (cit. on p. 14).
- [5] K. Sevnarayan, “The implementation of telegram as a pedagogical tool to enhance student motivation and interaction,” *Journal of Education Technology*, vol. 7, no. 1, 2023 (cit. on p. 14).
- [6] O. Talbi and A. Ouared, “Goal-oriented student motivation in learning analytics: How can a requirements-driven approach help?” *Education and Information Technologies*, vol. 27, no. 9, pp. 12 083–12 121, 2022 (cit. on p. 14).
- [7] A. Ouared, M. Amrani, and P.-Y. Schobbens, “Learning analytics solution for monitoring and analyzing the students’ behavior in sql lab work,” 2023 (cit. on p. 14).
- [8] A. Shevat, *Designing bots: Creating conversational experiences*. " O'Reilly Media, Inc.", 2017 (cit. on p. 14).
- [9] J.-B. Aujogue and A. Aussem, “Hierarchical recurrent attention networks for context-aware education chatbots,” in *2019 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2019, pp. 1–8 (cit. on p. 14).
- [10] D. Khurana, A. Koli, K. Khatte, and S. Singh, “Natural language processing: State of the art, current trends and challenges,” *Multimedia tools and applications*, vol. 82, no. 3, pp. 3713–3744, 2023 (cit. on pp. 16, 17).
- [11] Y. Vasiliev, *Natural Language Processing with Python and spaCy: A Practical Introduction*. No Starch Press, 2020 (cit. on p. 16).
- [12] R. Sharnagat, “Named entity recognition: A literature survey,” *Center For Indian Language Technology*, vol. 1, p. 1, 2014 (cit. on p. 17).
- [13] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, et al., “Large language models: A survey,” *arXiv preprint arXiv:2402.06196*, 2024. [Online]. Available: <https://arxiv.org/abs/2402.06196> (cit. on pp. 18, 19, 22).

- [14] Anonymous, “Evaluation of open and closed-source llms for low-resource language with zero-shot, few-shot, and chain-of-thought prompting,” *Natural Language Processing Journal*, vol. 10, p. 100 124, 2025 (cit. on p. 18).
- [15] D. R. Vasiliev, *Transformers for Natural Language Processing: Build and Train State-of-the-Art Natural Language Processing Models Using Transformer’s Architecture*. Packt Publishing, 2020 (cit. on p. 20).
- [16] M. Wisdom, *Understanding the ml pipeline for nlp tasks*, Accessed: 2025-06-22, 2022. [Online]. Available: https://medium.com/@asjad_ali/understanding-the-nlp-pipeline-a-comprehensive-guide-828b2b3cd4e2 (cit. on p. 20).
- [17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018 (cit. on p. 20).
- [18] T. B. Brown et al., “Language models are few-shot learners,” *arXiv preprint arXiv:2005.14165*, 2020 (cit. on p. 20).
- [19] S. Minaee, N. Kalchbrenner, H. Ali, et al., “Transfer learning for natural language processing: A survey,” *arXiv preprint arXiv:2401.00001*, 2024 (cit. on p. 20).
- [20] A. Khurana, A. Patel, and D. Shah, “Llms in production: Architectures and use cases,” *arXiv preprint arXiv:2307.00001*, 2023 (cit. on p. 20).
- [21] H. Touvron et al., *Llama: Open and efficient foundation language models*, arXiv:2302.13971 [cs.CL], 2023. [Online]. Available: <https://doi.org/10.48550/arXiv.2302.13971> (cit. on pp. 21, 22).
- [22] Y. Gao et al., “Retrieval-augmented generation for large language models: A survey,” *arXiv preprint arXiv:2312.10997*, 2023. [Online]. Available: <https://arxiv.org/abs/2312.10997> (cit. on p. 22).
- [23] P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha, “A systematic survey of prompt engineering in large language models: Techniques and applications,” *arXiv preprint arXiv:2402.07927*, 2024. [Online]. Available: <https://arxiv.org/abs/2402.07927> (cit. on p. 23).
- [24] T. Debnath, M. N. A. Siddiky, M. E. Rahman, P. Das, and A. K. Guha, “A comprehensive survey of prompt engineering techniques in large language models,” *TechRxiv*, 2024. [Online]. Available: <https://www.techrxiv.org/users/898487/articles/1274333-a-comprehensive-survey-of-prompt-engineering-techniques-in-large-language-models> (cit. on p. 23).
- [25] J. Decety and P. L. Jackson, “The functional architecture of human empathy,” *Behavioral and Cognitive Neuroscience Reviews*, vol. 3, no. 2, pp. 71–100, 2004 (cit. on p. 25).
- [26] Y. Luo and et al., “Assessing empathy in large language models with real-world physician-patient interactions,” *medRxiv*, 2024. [Online]. Available: <https://arxiv.org/abs/2405.16402v1> (cit. on p. 25).
- [27] E. L. Deci and R. M. Ryan, “The “what” and “why” of goal pursuits: Human needs and the self-determination of behavior,” *Psychological Inquiry*, vol. 11, no. 4, pp. 227–268, 2000. DOI: [10.1207/S15327965PLI1104_01](https://doi.org/10.1207/S15327965PLI1104_01) (cit. on p. 26).
- [28] R. Sajja, Y. Sermet, M. Cikmaz, D. Cwiertny, and I. Demir, “Artificial intelligence-enabled intelligent assistant for personalized and adaptive learning in higher education,” 2023. [Online]. Available: <https://arxiv.org/abs/2309.10892> (cit. on p. 26).
- [29] S. Maity and A. Deroy, “Generative ai and its impact on personalized intelligent tutoring systems,” 2024. [Online]. Available: <https://arxiv.org/abs/2410.10650> (cit. on p. 26).

- [30] K.-J. Laak and J. Aru, “Ai and personalized learning: Bridging the gap with modern educational goals,” 2024. [Online]. Available: <https://arxiv.org/abs/2404.02798> (cit. on p. 26).
- [31] S. Wu et al., “A comprehensive exploration of personalized learning in smart education: From student modeling to personalized recommendations,” 2024. [Online]. Available: <https://arxiv.org/abs/2402.01666> (cit. on p. 26).
- [32] J. Goetz, S. Kiesler, and A. Powers, “Matching robot appearance and behavior to tasks to improve human-robot cooperation,” in *Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication (ROMAN)*, IEEE, 2003, pp. 55–60. DOI: [10.1109/ROMAN.2003.1251796](https://doi.org/10.1109/ROMAN.2003.1251796) (cit. on p. 26).
- [33] A. Kobsa, “Privacy-enhanced web personalization,” in *The Adaptive Web*, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds., Springer, 2007, pp. 628–670. DOI: [10.1007/978-3-540-72079-9_21](https://doi.org/10.1007/978-3-540-72079-9_21) (cit. on p. 26).
- [34] P. Salovey and J. D. Mayer, “Emotional intelligence,” *Imagination, Cognition and Personality*, vol. 9, no. 3, pp. 185–211, 1990 (cit. on p. 27).
- [35] GeeksforGeeks, *Introduction to ucd (user centered design)*, Accessed: 2025-06-01, 2022. [Online]. Available: <https://www.geeksforgeeks.org/introduction-to-ucd-user-centered-design/> (cit. on p. 29).
- [36] F. A. Tellez and J. Gonzalez-Tobon, “Empathic design as a framework for creating meaningful experiences,” *Conference Proceedings of the Academy for Design Innovation Management*, vol. 2, no. 1, pp. 909–919, 2019, Accessed: 2025-06-01. [Online]. Available: https://libres.uncg.edu/ir/asu/f/Tellez_Fabio_and%20Gonzalez-Tobon_2019_Empathic%20Design.pdf (cit. on p. 29).
- [37] Y. Zhou, M. Zhou, J. Luo, and A. Zhao, “From prompting to partnering: Personalization features for human-llm interactions,” *arXiv preprint arXiv:2503.00681*, 2025. [Online]. Available: <https://arxiv.org/abs/2503.00681> (cit. on p. 29).
- [38] J. Chen, D. Santos, M. Tran, and S. Lee, “Designing adaptive user interfaces for mhealth applications targeting chronic disease: A user-centered approach,” *arXiv preprint arXiv:2405.08302*, 2024. [Online]. Available: <https://arxiv.org/abs/2405.08302> (cit. on p. 29).
- [39] Y. Zhou, M. Zhou, J. Luo, and A. Zhao, “From prompting to partnering: Personalization features for human-llm interactions,” *arXiv preprint arXiv:2503.00681*, 2025. [Online]. Available: <https://arxiv.org/abs/2503.00681> (cit. on p. 29).
- [40] Y. Fan, Y. Shi, and K. N. Truong, “Practices and challenges of using think-aloud protocols in industry: An international survey,” *Journal of Usability Studies*, vol. 15, no. 2, pp. 55–77, Feb. 2020. [Online]. Available: https://uxpajournal.org/wp-content/uploads/sites/7/pdf/JUS_Fan_Feb2020.pdf (cit. on p. 30).