SmokeCtrl: AI-powered support for quitting tobacco.

Project-III (CS57003) report submitted to

Indian Institute of Technology Kharagpur

in partial fulfilment for the award of the degree of

Master of Technology

in

Computer Science and Engineering

by

Hardik Pravin Soni

(20CS30023)

Under the supervision of

Professor Jayanta Mukhopadhyay

Department of Computer Science and Engineering

Indian Institute of Technology Kharagpur

Autumn Semester, 2024-25

November 7, 2024

DECLARATION

I certify that

(a) The work contained in this report has been done by me under the guidance of my supervisor.

(b) The work has not been submitted to any other Institute for any degree or diploma.

(c) I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

(d) Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Date: November 7, 2024
Place: Kharagpur

(Hardik Pravin Soni)
(20CS30023)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

KHARAGPUR - 721302, INDIA

CERTIFICATE

This is to certify that the project report entitled "**SmokeCtrl: AI-powered support for quitting tobacco.**" submitted by **Hardik Pravin Soni** (Roll No. 20CS30023) to Indian Institute of Technology Kharagpur towards partial fulfilment of requirements for the award of degree of Master of Technology in Computer Science and Engineering is a record of bona fide work carried out by him under my supervision and guidance during Autumn Semester, 2024-25.

Jayanta Mukhopadhyay

Professor Jayanta Mukhopadhyay
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
Kharagpur - 721302, India

Date: November 7, 2024

Place: Kharagpur

Abstract

Name of the student: **Hardik Pravin Soni** Roll No: **20CS30023**
Degree for which submitted: **Master of Technology**
Department: **Department of Computer Science and Engineering**
Thesis title: **SmokeCtrl: AI-powered support for quitting tobacco.**
Thesis supervisor: **Professor Jayanta Mukhopadhyay**
Month and year of thesis submission: **November 7, 2024**

SmokeCtrl: A Flutter-based mobile app leveraging Llama 3.2 to support and guide tobacco cessation for a healthier, smoke-free life.

SmokeCtrl is a mobile application developed using **Flutter**, designed specifically to assist patients in overcoming **tobacco addiction**. By leveraging the capabilities of **large language models (LLMs)**, SmokeCtrl provides *intelligent* and *tailored* responses to users' medical queries, supporting their journey towards cessation. The application features:

- A **simple, dynamic, and user-friendly UI** for easy navigation, offering **login** and **sign-up** functionalities.
- Back-end integration with **Spring** to manage user authentication and database operations.
- **Llama 3.2-based model inference**, offering interactive and responsive dialogue for users by generating relevant responses to prompts.
- **Retrieval-Augmented Generation (RAG)** setup using **LangChain**, with a *Chroma database* to support *context-aware* response generation.
- Fine-tuning of **Llama 3.2** using **Quantized Low-Rank Adaptation (qLoRA)** to optimize performance specifically for conversational and retrieval tasks.

Beyond assisting individual users, SmokeCtrl contributes to **societal well-being** by:

1. Encouraging **public health awareness** about tobacco's adverse effects, reaching a broad audience through **mobile accessibility**.

2. Reducing **healthcare costs** by promoting early interventions and preventive measures in tobacco cessation.
3. Supporting **healthcare providers** with valuable insights into patient progress, facilitating better-targeted support.
4. Enhancing **mental and emotional well-being** by providing round-the-clock access to resources and personalized motivation.
5. Strengthening **community health** by promoting a culture of wellness and offering a tool for family and friends to support users in their cessation journey.

Keywords: Tobacco cessation, Flutter application, large language model, retrieval-augmented generation, Quantized qLoRA

Acknowledgements

I express my sincere gratitude to my supervisor, Professor Jayanta Mukhopadhyay, for granting me the opportunity to work under his mentorship. Throughout my M.Tech project, he provided insightful guidance and constructive feedback. He was consistently supportive and understanding.

I am grateful for the chance to have worked on this project and to have learned from the people I have met along the way. This project has been challenging but rewarding, and I am proud of what I have achieved. I would also like to extend my warmest gratitude to my family and friends at IIT Kharagpur for their unwavering support and motivation throughout my journey.

Contents

Declaration i

Certificate ii

Abstract iii

Acknowledgements v

Contents vi

List of Figures ix

List of Tables x

Abbreviations xi

1 Introduction 1

1.1 Overview . . . 1
1.2 Problem Statement . . . 2
1.3 Research Questions . . . 3
1.4 Objective of the Thesis . . . 3
1.5 Scope . . . 4
1.6 Goals of the Thesis . . . 5
1.6.1 Development of the SmokeCtrl Application . . . 5
1.6.2 Implementation of Retrieval-Augmented Generation (RAG) . . . 5
1.6.3 Fine-tuning with Quantized Low-Rank Adaptation (qLoRA) . . . 5
1.6.4 Evaluation of User Engagement and Application Effectiveness 6
1.6.5 Analysis of Results and Impact . . . 6

2 Background 7

2.1 Introduction to Tobacco Cessation . . . 7
2.1.1 Importance of Tobacco Cessation . . . 7
2.1.2 Health Risks of Tobacco Use . . . 8
2.1.3 Challenges in Quitting Tobacco . . . 8

2.1.4 Need for Comprehensive Support . . . 8
2.1.5 Role of Digital Tools in Addiction Recovery . . . 8
2.1.6 Overcoming Obstacles . . . 8
2.2 Overview of Mobile Health (mHealth) Applications . . . 9
2.2.1 Evolution of mHealth for Addiction Support . . . 9
2.2.2 Benefits of mHealth Applications in Healthcare . . . 9
2.2.3 Current Applications for Tobacco Cessation . . . 9
2.2.4 Limitations of Existing Tobacco Cessation Apps . . . 10
2.3 Large Language Models (LLMs) . . . 10
2.3.1 Introduction to Large Language Models . . . 10
2.3.2 Role of NLP in Healthcare Applications . . . 10
2.3.3 Advantages of LLMs in Providing Personalized Responses . . . 11
2.3.4 Challenges of Integrating LLMs in Real-Time Applications . . . 11
2.4 Quantized Low-Rank Adaptation (qLoRA) . . . 11
2.4.1 Overview of Low-Rank Adaptation . . . 11
2.4.2 Introduction to Quantization Techniques . . . 12
2.4.3 Benefits of qLoRA in Mobile Applications . . . 12
2.4.4 Impact of qLoRA on Model Efficiency and Accuracy . . . 12
2.5 Retrieval-Augmented Generation (RAG) Framework . . . 12
2.5.1 Overview of RAG in NLP . . . 12
2.5.2 RAG Implementation in Healthcare Applications . . . 13
2.5.3 Chroma Database in Retrieval Mechanisms . . . 13
2.5.4 Enhancing Contextual Relevance Using RAG . . . 14
2.6 Mobile Development Frameworks and Technologies . . . 15
2.6.1 Introduction to Flutter for Cross-Platform Development . . . 15
2.6.2 Overview of Spring for Backend Support . . . 15
2.6.3 Database Management and Security Considerations . . . 15
2.6.4 Integrating LLMs with Mobile and Backend Systems . . . 16

3 Literature Review **17**
3.1 Transformers and Neural Network Architectures . . . 17
3.2 Large Language Models . . . 18
3.3 LLMs in Healthcare Applications . . . 18
3.4 Advancements in Model Fine-Tuning and Adaptation . . . 18
3.5 Retrieval-Augmented Generation (RAG) . . . 19
3.6 Mobile Frameworks for Cross-Platform Development . . . 19
3.7 Backend Support and Database Management . . . 19
3.8 User Engagement and Retention in mHealth Applications . . . 20
3.9 Evaluation Metrics for Healthcare LLMs . . . 20
3.10 Applications of AI in Behavioral Health and Addiction Support . . . 20

4 Methodology **21**

4.1 Mobile Application Development Framework . . . 21
4.2 Back-End Development and Data Management . . . 22
4.3 Integration of Large Language Models . . . 22
4.4 Inference through Retrieval-Augmented Generation (RAG) . . . 22
4.5 QLoRA:Efficient Finetuning of Quantized LLMs . . . 23
4.5.1 Background . . . 24
4.5.2 Block-wise k-bit Quantization . . . 24
4.5.3 Low-rank Adapter (LoRA) Finetuning . . . 24
4.5.4 QLoRA: Memory Optimization Techniques . . . 25
4.5.5 Guanaco Results . . . 27
4.6 Experiments . . . 31

5 Implementation **32**
5.1 Front-End and Schema Design . . . 32
5.2.2 Home Page . . . 32
5.2.3 Login and Sign-Up Page . . . 32
5.2.4 User ID Tracking and Registration Statistics . . . 33
5.2 Spring Back-End and API Endpoints . . . 34
5.2.1 Authentication Endpoints . . . 34
5.3 Retrieval-Augmented Inference using LangChain . . . 36
5.4 Fine-tuning Llama-3.2-1b-instruct using qLoRA . . . 37

6 Results and Observations **40**
6.1 Results and Observations . . . 40
6.1.1 ROGUE . . . 40
6.1.2 MMLU . . . 41
6.1.3 ARC . . . 41
6.1.4 SQuAD . . . 41
6.1.5 Hellaswag . . . 42
6.2 Benchmarks . . . 42
6.2.1 Base Pretrained LLaMA 3.2 Models . . . 42
6.2.2 Curated Fine-Tuned LLaMA 3.2 Models . . . 43

7 Conclusions **44**
7.1 Conclusion . . . 44

8 Future Work **46**

List of Figures

2.1 Retrieval Augmented Generation Workflow 13
2.2 Creation of Embedding using PDF Input [Source] 14
2.3 General Overview of Spring Back-end 16

4.1 Our re-parametrization. We only train A and B. (24) 25
4.2 Full Finetuning, LoRA & QLoRA 25
4.3 Mean Zero-Shot Accuracy across various benchmarks. 28

5.1 Screenshots of the application's Home and Sign-Up screens. 33
5.2 Database Schema Diagram 34
5.3 Context Dependency Diagram for Back-End Architecture 34
5.4 Screenshots of the application's Home and Sign-Up screens. 35
5.5 Screenshots of a Sample Response being generated. 38

List of Tables

4.1 Mean Perplexity (PPL) for various data types. 28
4.2 Mean 5-Shot MMLU Accuracy 29
4.3 Elo ratings from a competition between models. 30
4.4 Zero-shot Vicuna benchmark scores as a percentage of the score obtained by ChatGPT evaluated by GPT-4. We see that OASST1 models perform close to ChatGPT despite being trained on a very small dataset and having a fraction of the memory requirement of baseline models. 30

6.1 Benchmark results for Llama models across various tasks. 42
6.2 Benchmark results for various Llama 3.2 models across different tasks and configurations. 43
6.3 Benchmark results for the Experiments conducted by me during the course of Project. Note that the sce.v.1..3 are LLaMA 3.2 Fine-tuned first on **UltraChat-200k** (25) and then on Conversation Set-1, Conversation Set-2 and Combined Version respectively 43

Abbreviations

LLM	**L**arge **L**anguage **M**odel
NLP	**N**atural **L**anguage **P**rocessing
mHealth	**M**obile **H**ealth
RAG	**R**etrieval-**A**ugmented **G**eneration
qLoRA	**Q**uantized **L**ow-**R**ank **A**daptation
UI	**U**ser **I**nterface
API	**A**pplication **P**rogramming **I**nterface
JWT	**J**SON **W**eb **T**oken (used for authentication)
CRUD	**C**reate, **R**ead, **U**pdate, **D**elete (basic database operations)
GPU	**G**raphics **P**rocessing **U**nit
ML	**M**achine **L**earning
AI	**A**rtificial **I**ntelligence
SSL	**S**ecure **S**ockets **L**ayer (for secure data transmission)
HTTP	**H**ypertext **T**ransfer **P**rotocol
SDK	**S**oftware **D**evelopment **K**it
UX	**U**ser **E**xperience
DB	**D**atabase
OAuth	**O**pen **A**uthorization (often used for user authentication)
REST	**R**epresentational **S**tate **T**ransfer (API architectural style)

Chapter 1

Introduction

1.1 Overview

In recent years, there has been a growing interest in developing **intelligent, AI-driven healthcare applications** to tackle pressing public health issues like **tobacco addiction**. Significant technological advancements, especially in *natural language processing (NLP)* and mobile development frameworks, have facilitated the creation of tools that offer **personalized support** for health challenges such as addiction cessation.

The potential applications of **mobile-based cessation tools** like SmokeCtrl extend far beyond traditional interventions, utilizing the power of **large language models (LLMs)** to provide *real-time, contextually relevant responses* to user queries. These capabilities are strengthened by state-of-the-art **retrieval-augmented generation (RAG)** mechanisms, which retrieve the most relevant information from a database of responses, providing an enhanced user experience and helping users make *informed decisions* about their health.

A critical challenge in designing **mobile applications for healthcare** lies in creating a **user-friendly interface** that is both accessible and engaging for users.

SmokeCtrl addresses this through a **dynamic and intuitive UI** built with Flutter, making it easy for users to navigate and access resources. Additionally, *back-end support* provided by **Spring** ensures secure and efficient management of user data, offering a seamless user experience.

One of the major hurdles in AI-driven healthcare applications is ensuring model efficiency and accuracy. SmokeCtrl overcomes this challenge by employing **quantized Low-Rank Adaptation (qLoRA)** for fine-tuning **LLMs**, allowing for resource-efficient operation and improved task-specific performance. The integration of Llama 3.2 as the core model further enhances response generation, aligning with the goal of delivering **precise, contextually appropriate answers** in a manner accessible to a broad user base.

1.2 Problem Statement

Despite the availability of various resources for tobacco cessation, individuals struggling with addiction often lack personalized, accessible, and contextually relevant support to aid them in their journey. While advancements in artificial intelligence and mobile technology offer promising tools, there is still a need for a system that provides intelligent, tailored responses to specific medical queries related to tobacco cessation. For our thesis, we, therefore, ask this question:

> How can a mobile application, driven by large language models and retrieval-augmented generation, effectively support and enhance the tobacco cessation process for individual users?

1.3 Research Questions

In this study, we aim to investigate the potential of a mobile application powered by large language models to support tobacco cessation efforts. The following research questions guide our exploration of this objective:

1. How effective is a mobile application utilizing **large language models** in providing *personalized support* for tobacco cessation?
2. In what ways can **retrieval-augmented generation (RAG)** improve the relevance and contextual accuracy of responses provided to users in a healthcare setting?
3. How does the integration of **quantized low-rank adaptation (qLoRA)** impact the efficiency and accuracy of large language models within a mobile application for real-time interaction?
4. What are the primary factors influencing **user engagement and retention** within a mobile application aimed at supporting tobacco cessation?
5. How does access to round-the-clock, *AI-driven support* in a mobile application affect the success rates of individuals attempting to quit tobacco?

By addressing these research questions, we seek to evaluate the feasibility and effectiveness of *AI-driven, mobile-based support* for tobacco cessation, as well as its potential impact on **user engagement** and **cessation outcomes**.

1.4 Objective of the Thesis

The purpose of this thesis is to present the development and evaluation of a mobile application, **SmokeCtrl**, designed to support tobacco cessation by utilizing

large language models. This includes a description of the app's architecture and features, an analysis of how effectively it provides *personalized support* to users, a discussion of the user engagement data collected, and conclusions regarding its potential impact on aiding individuals in their journey toward tobacco cessation.

1.5 Scope

This thesis will focus on the development and assessment of the **SmokeCtrl** mobile application, which utilizes **large language models (LLMs)** to support individuals in their journey toward tobacco cessation. The application is built with the Flutter framework for front-end development, while the back-end relies on **Spring** for data management and user authentication. The large language model, Llama 3.2, is integrated within the app to provide *personalized, contextually relevant responses* to user queries, further enhanced by a **Retrieval-Augmented Generation (RAG)** system utilizing a *Chroma database*.

Throughout the study, we will evaluate the application's effectiveness in supporting tobacco cessation through **user engagement analysis** and by measuring the relevance and accuracy of the model responses. Additionally, we will analyze the impact of **quantized Low-Rank Adaptation (qLoRA)** on the app's performance and efficiency in generating responses. This thesis will also consider the role of a user-friendly interface in facilitating accessibility and retention for users.

By examining these elements, this research aims to assess the potential of *AI-driven mobile applications* as a viable solution for tobacco cessation and to explore the broader implications of personalized digital support in addiction recovery.

1.6 Goals of the Thesis

The main objective of this thesis is to develop and evaluate the **SmokeCtrl** mobile application as an AI-driven tool to support tobacco cessation. This has been fulfilled by achieving the following goals:

1.6.1 Development of the SmokeCtrl Application

The **SmokeCtrl** application has been designed and developed using Flutter for front-end architecture and **Spring** for back-end data management. The large language model Llama 3.2 is integrated within the app to provide intelligent, contextually relevant responses to user queries. The application includes key features such as a user-friendly interface, secure authentication, and real-time response generation.

1.6.2 Implementation of Retrieval-Augmented Generation (RAG)

To enhance response relevance, the application integrates a **Retrieval-Augmented Generation (RAG)** mechanism utilizing a *Chroma database* that stores key information. This approach allows the model to retrieve contextually significant data, improving response accuracy and aligning with user needs.

1.6.3 Fine-tuning with Quantized Low-Rank Adaptation (qLoRA)

The large language model has been optimized with **Quantized Low-Rank Adaptation (qLoRA)** to improve response efficiency and resource utilization. This fine-tuning process focuses on achieving high performance in generating real-time responses suitable for mobile deployment.

1.6.4 Evaluation of User Engagement and Application Effectiveness

The application's impact on user engagement and retention has been assessed through data collection and analysis of user interaction patterns. This evaluation examines the app's effectiveness in providing personalized support for tobacco cessation.

1.6.5 Analysis of Results and Impact

The outcomes of the evaluation are presented and analyzed, with a focus on how the app's intelligent response system, user-friendly design, and personalized support influence the cessation journey. This includes an assessment of the potential benefits of AI-driven mobile applications in addiction recovery.

Chapter 2

Background

In this chapter, we will explore key concepts in mobile application development and advanced AI techniques relevant to tobacco cessation support. Only the foundational principles necessary to understand the application's design and functionality are discussed (Refer to Chapter 4 for more details).

2.1 Introduction to Tobacco Cessation

2.1.1 Importance of Tobacco Cessation

Tobacco cessation is a critical area in public health, as tobacco use remains one of the leading causes of preventable diseases and deaths worldwide. Quitting tobacco is challenging due to the addictive nature of nicotine and the behavioral dependencies that often accompany long-term use.

2.1.2 Health Risks of Tobacco Use

Tobacco use poses significant health risks, including heart disease, lung cancer, and respiratory illnesses, making cessation a public health priority.

2.1.3 Challenges in Quitting Tobacco

Nicotine's addictive properties and the psychological dependence it creates make quitting tobacco difficult for many users.

2.1.4 Need for Comprehensive Support

Successful cessation often requires a blend of behavioral support, counseling, and, when needed, pharmacological aids to manage cravings and withdrawal symptoms.

2.1.5 Role of Digital Tools in Addiction Recovery

Mobile applications and digital tools have become valuable in tobacco cessation, offering accessible, personalized support directly on users' devices.

2.1.6 Overcoming Obstacles

By providing real-time, tailored interventions, digital tools help users overcome common obstacles in cessation, improving motivation and increasing the likelihood of successful quit attempts.

2.2 Overview of Mobile Health (mHealth) Applications

2.2.1 Evolution of mHealth for Addiction Support

mHealth applications have evolved significantly, transforming from basic informational tools to sophisticated, interactive platforms that provide tailored support for addiction recovery. Early mobile apps mainly offered general health tips, whereas modern mHealth solutions leverage AI and personalization to offer guidance and coping strategies, making them a valuable asset for tobacco cessation.

2.2.2 Benefits of mHealth Applications in Healthcare

mHealth applications offer a range of benefits in healthcare, such as accessibility, convenience, and real-time support, making them particularly effective in reaching users wherever they are. For tobacco cessation, these apps provide a non-judgmental space where users can access personalized content, track their progress, and receive timely encouragement and reminders to stay committed to their goals.

2.2.3 Current Applications for Tobacco Cessation

Several mHealth applications specifically designed for tobacco cessation are available today, offering features like goal-setting, motivational content, and peer support. Apps such as Quit Genius and Smoke Free are popular options, integrating behavioral science and habit tracking to help users make gradual lifestyle changes and overcome addiction.

2.2.4 Limitations of Existing Tobacco Cessation Apps

Despite their benefits, many current tobacco cessation apps face limitations, such as a lack of personalization, inadequate response to relapse situations, and limited integration of evidence-based strategies. These gaps highlight the need for enhanced mHealth solutions that leverage AI to deliver more adaptive and contextually relevant support to users.

2.3 Large Language Models (LLMs)

2.3.1 Introduction to Large Language Models

Large language models (LLMs) are advanced AI systems capable of processing vast amounts of data to generate human-like text. These models play a crucial role in healthcare by enabling personalized interactions, which is particularly beneficial in *tobacco cessation* support applications, where customized guidance can make a significant impact on user motivation and success.

2.3.2 Role of NLP in Healthcare Applications

Natural Language Processing (NLP) in healthcare allows applications to understand and respond to user queries intelligently. In tobacco cessation, NLP enables apps to interpret individual challenges and provide responses that address specific aspects of addiction, offering encouragement, advice, and resources tailored to each user's unique journey.

2.3.3 Advantages of LLMs in Providing Personalized Responses

LLMs provide a high level of *personalization*, adjusting responses based on user input and engagement history. This ability is critical in tobacco cessation apps, as it allows users to feel understood and supported, improving adherence to cessation plans and promoting better health outcomes through individualized guidance.

2.3.4 Challenges of Integrating LLMs in Real-Time Applications

Integrating LLMs in real-time applications poses challenges such as *latency* and **resource consumption**, which are critical in mobile healthcare. Real-time responses are essential in cessation support, where timely interaction can make a difference in a user's ability to handle cravings and stay on track with their goals.

2.4 Quantized Low-Rank Adaptation (qLoRA)

2.4.1 Overview of Low-Rank Adaptation

Low-Rank Adaptation (LoRA) is a method that adjusts specific neural network layers, allowing for resource-efficient performance without compromising accuracy. This technique is particularly beneficial for *tobacco cessation applications*, where efficient model deployment on mobile devices can provide real-time support without overwhelming system resources.

2.4.2 Introduction to Quantization Techniques

Quantization reduces the precision of model weights, making models more memory-efficient and accessible on mobile platforms. This reduction is valuable in tobacco cessation apps, where delivering quick, accurate responses on low-resource devices is essential to ensure user engagement and support at critical moments.

2.4.3 Benefits of qLoRA in Mobile Applications

Quantized Low-Rank Adaptation (qLoRA) optimizes large models by reducing their memory and computational demands. In mobile tobacco cessation applications, this adaptation supports real-time interaction, helping users manage their cravings and access encouragement immediately, enhancing the app's effectiveness.

2.4.4 Impact of qLoRA on Model Efficiency and Accuracy

By using **qLoRA**, models achieve a balance between *efficiency* and *accuracy*, crucial for mobile apps that need rapid response times. This balance allows tobacco cessation apps to maintain high-quality support while remaining accessible to users with limited device capabilities.

2.5 Retrieval-Augmented Generation (RAG) Framework

2.5.1 Overview of RAG in NLP

Retrieval-Augmented Generation (RAG) combines generative AI with information retrieval to generate contextually relevant responses. In tobacco cessation

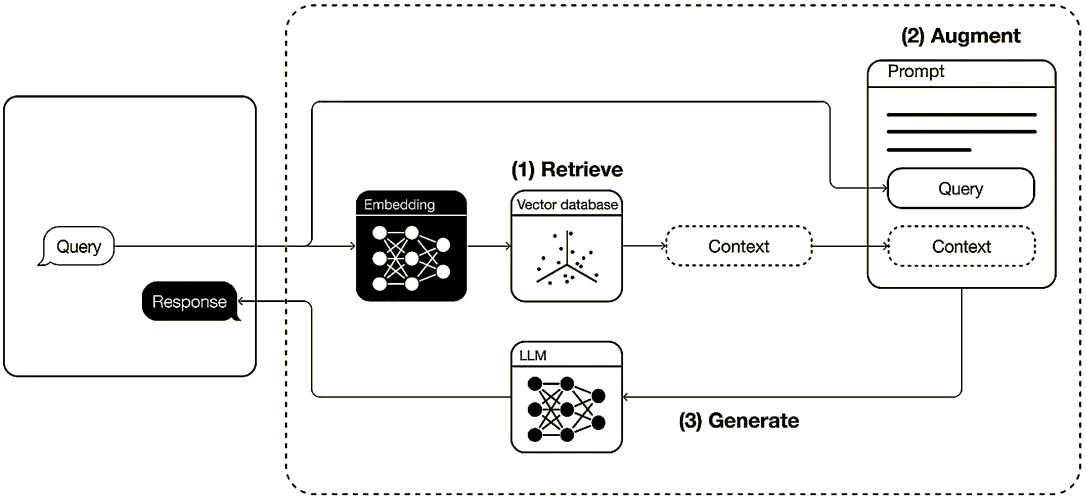


FIGURE 2.1: Retrieval Augmented Generation Workflow

apps, RAG enables the model to draw on stored information about addiction and cessation strategies, helping users receive responses grounded in accurate, evidence-based resources.

2.5.2 RAG Implementation in Healthcare Applications

In healthcare applications, **RAG** enhances response accuracy by retrieving relevant data before generating an answer. For tobacco cessation, this process ensures users receive precise information about addiction, withdrawal, and coping techniques, making the support more effective and trustworthy.

2.5.3 Chroma Database in Retrieval Mechanisms

The **Chroma database** stores embeddings that represent relevant cessation-related content, facilitating quick retrieval of supportive material. This retrieval process is

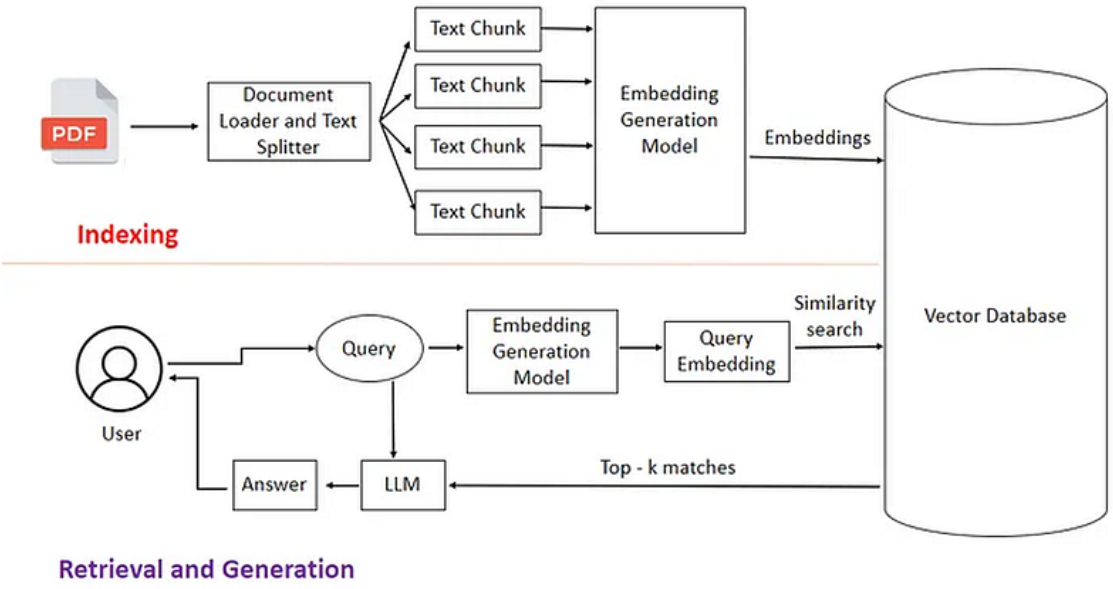


FIGURE 2.2: Creation of Embedding using PDF Input [Source]

essential in providing contextually meaningful responses to users at moments when personalized support is needed to stay on track with quitting.

2.5.4 Enhancing Contextual Relevance Using RAG

RAG enhances *contextual relevance* by combining user-specific information with high-quality resources, helping tobacco cessation apps provide guidance that aligns with each user's experiences. This contextual accuracy is essential for reinforcing user motivation and supporting long-term behavior change.

2.6 Mobile Development Frameworks and Technologies

2.6.1 Introduction to Flutter for Cross-Platform Development

Flutter is a powerful framework for creating cross-platform applications, allowing seamless deployment on iOS and Android from a single codebase. For tobacco cessation apps, Flutter enables a unified experience across devices, helping more users access critical cessation resources without compatibility issues.

2.6.2 Overview of Spring for Backend Support

Spring offers secure and efficient back-end support, ideal for handling sensitive healthcare data in tobacco cessation apps. With Spring, developers can manage user authentication and store cessation progress data securely, enhancing both user trust and data privacy.

2.6.3 Database Management and Security Considerations

Effective **database management** and *security* are essential in applications dealing with sensitive health data, such as tobacco cessation. Secure data storage and access control ensure that user information, including progress tracking and personal preferences, remains confidential and protected.

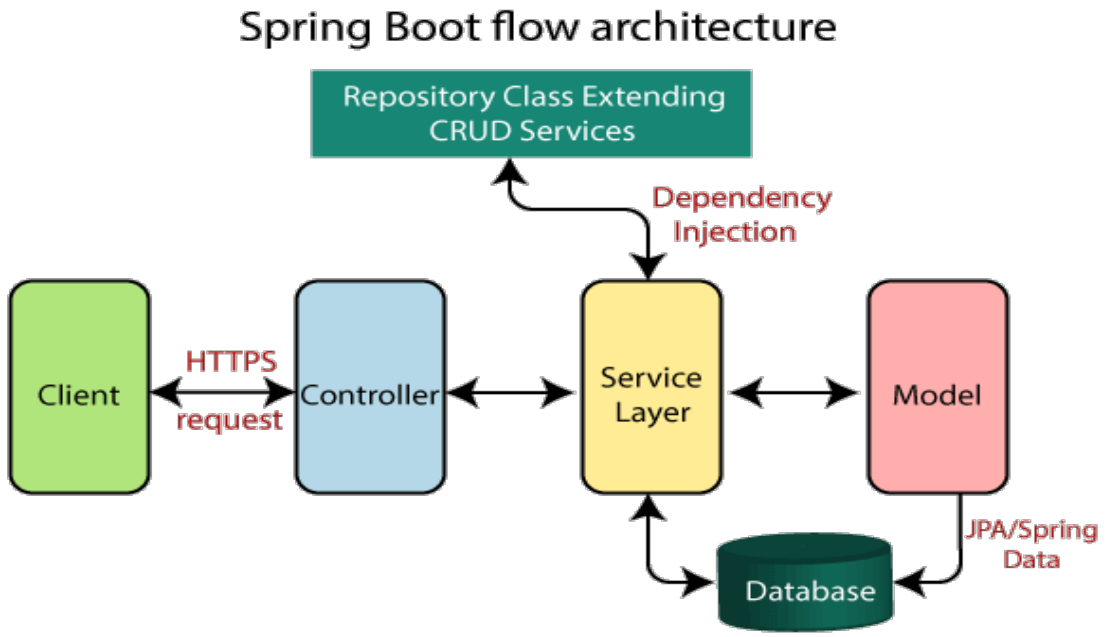


FIGURE 2.3: General Overview of Spring Back-end

2.6.4 Integrating LLMs with Mobile and Backend Systems

Integrating **LLMs** with mobile and backend systems allows applications to deliver real-time, personalized responses for tobacco cessation. Proper integration ensures smooth communication between the app, the back-end, and the LLM, enabling consistent support that encourages users to overcome cravings and maintain progress.

Chapter 3

Literature Review

The development of our AI-driven tobacco cessation support system builds upon a rich foundation of research in the fields of **transformers**, **large language models (LLMs)**, and **personalized healthcare support**. This literature review explores foundational work in each of these areas and highlights contributions that inform the development of our project.

3.1 Transformers and Neural Network Architectures

The introduction of **transformer models** by Vaswani et al. ([1]) has significantly impacted NLP, enabling models to process language with unprecedented accuracy and speed. Further improvements in transformers have been made with BERT ([2]), which utilizes bidirectional training, and GPT ([3]), which emphasizes autoregressive generation. These architectures have been instrumental in developing modern LLMs, including BERT-based models for healthcare ([4]).

3.2 Large Language Models

Large language models such as GPT-3 (5) and T5 (6) demonstrated the capability of LLMs to generate coherent and contextually relevant text across various domains. The development of more refined models, including OpenAI's ChatGPT (7) and Google's LaMDA (8), further improved conversational capabilities, setting the stage for applications in personalized healthcare, including tobacco cessation.

3.3 LLMs in Healthcare Applications

In healthcare, LLMs like BioBERT (9) and ClinicalBERT (10) are tailored specifically to medical literature, providing accurate responses to complex medical queries. These models offer insights into symptom management, treatment advice, and health behavior support, making them valuable assets for addiction and cessation support.

3.4 Advancements in Model Fine-Tuning and Adaptation

Techniques such as Low-Rank Adaptation (LoRA) (11) and Quantized LoRA (qLoRA) (12) enable efficient model fine-tuning, enhancing response accuracy while minimizing computational demands. Recent research on PEFT (Parameter-Efficient Fine-Tuning) (13) explores adaptation methods that are resource-efficient, which is particularly relevant in mobile applications like tobacco cessation.

3.5 Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) combines retrieval mechanisms with generative models, as demonstrated in Lewis et al.'s work (14). RAG has shown promise in providing contextually enriched responses in healthcare, which is essential for delivering meaningful advice in cessation apps. Chroma-based retrieval (**?**) supports efficient information storage and retrieval, critical for on-demand response generation in mobile applications.

3.6 Mobile Frameworks for Cross-Platform Development

Flutter (16) and **React Native** (17) are two popular frameworks that simplify cross-platform app development, reducing the time needed to build robust mobile healthcare applications. Flutter, in particular, has gained popularity for its performance on both iOS and Android, making it a suitable choice for developing healthcare applications like SmokeCtrl.

3.7 Backend Support and Database Management

Robust backend frameworks such as **Spring** (18) and Django (19) provide essential support for secure data management and authentication in healthcare applications. These frameworks offer scalable solutions to store and manage large volumes of user data, ensuring data security and compliance with health information privacy standards.

3.8 User Engagement and Retention in mHealth Applications

Studies in **mHealth** highlight the importance of engagement and retention metrics in healthcare applications (20). Strategies like personalized content and regular reminders have been shown to improve user commitment in cessation programs. By incorporating these strategies, our project aims to maintain high engagement levels and support sustained behavior change.

3.9 Evaluation Metrics for Healthcare LLMs

Evaluation metrics for LLMs in healthcare, including accuracy, relevance, and user satisfaction, are critical in assessing model performance (21). Metrics such as precision, recall, and F1 scores provide insights into how well the model meets the needs of healthcare users in sensitive applications like tobacco cessation.

3.10 Applications of AI in Behavioral Health and Addiction Support

AI has been widely researched for its role in behavioral health, with models addressing mental health, addiction, and behavior change (22). Studies have shown that AI-driven interventions can significantly impact addiction management, with personalized, accessible support improving the likelihood of cessation success.

Chapter 4

Methodology

This chapter outlines the methodology employed in developing **SmokeCtrl**, a mobile application for *tobacco cessation* that leverages the power of **LLM's** to provide intelligent support to users. The methodology includes the selection of frameworks, data handling strategies, model configurations, and inference techniques used to implement and fine-tune the application. We have employed a comprehensive methodology that incorporates the use of the **Flutter** framework, **PostgreSQL** for local data storage, and **Spring boot** framework for the server and **Python** Code for Running the Model. The methodology encompasses the methods, tools, and techniques used to address this challenge.

4.1 Mobile Application Development Framework

Flutter was chosen as the primary development framework due to its *cross-platform capabilities*, allowing SmokeCtrl to operate seamlessly on both Android and iOS. Flutter's widget-based architecture enables the creation of an interactive, user-friendly interface, while its robust community and package ecosystem ensure efficient

development with a wide array of functionalities. This cross-platform approach minimizes development time and ensures consistent user experience across devices.

4.2 Back-End Development and Data Management

The **back-end** was implemented using **Spring Boot**, which facilitates secure and scalable *data management* and *authentication*. Spring Boot's dependency injection capabilities, coupled with its embedded server, streamline back-end deployment. Key back-end functionalities include endpoints for **user authentication**, session management, and logging user activities. Authentication endpoints, including Sign-Up and Login, ensure secure user access, and a dedicated table tracks registration statistics, allowing for efficient monitoring of user engagement and activity.

4.3 Integration of Large Language Models

The core of SmokeCtrl's AI functionality is built on **Llama 3.2**, a large language model designed to provide personalized and contextually relevant responses. The model is initialized to handle prompts and generate responses in real time, with specific configurations for *asynchronous execution* using Python's asyncio library. This asynchronous setup allows efficient handling of user queries, providing immediate support without blocking main execution threads.

4.4 Inference through Retrieval-Augmented Generation (RAG)

The **Retrieval-Augmented Generation (RAG)** framework, implemented with **LangChain**, plays a pivotal role in enhancing response relevance. The *Chroma*

database is used for storing sentence embeddings generated via HuggingFace's models, enabling efficient retrieval of contextually pertinent information. LangChain's architecture supports a similarity search, retrieving the most relevant documents based on user queries, ensuring that responses are both *accurate* and *contextually enriched*.

4.5 QLoRA:Efficient Finetuning of Quantized LLMs

To optimize **Llama 3.2** for mobile application deployment, **Quantized Low-Rank Adaptation (qLoRA)** ([23]) was employed. This fine-tuning approach enhances resource efficiency while maintaining model accuracy, making it well-suited for memory-constrained environments. Key parameters, such as *alpha* and *rank*, were adjusted in `LoraConfig` to achieve a balance between task-specific performance and computational efficiency. This optimization enables *SmokeCtrl* to provide responsive, polite, and accurate support, aiding patients in their tobacco cessation journey without overtaxing device resources.

The innovations introduced by qLoRA ([23]) include:

1. **4-bit NormalFloat (NF4)**: A new data type designed to be theoretically optimal for representing normally distributed weights, reducing precision requirements without sacrificing performance.

2. **Double Quantization**: An additional quantization layer that compresses the quantization constants themselves, significantly reducing the average memory footprint.

3. **Paged Optimizers**: Memory management techniques that address memory spikes during training, supporting smoother and more efficient optimization processes.

Using qLoRA, the *Guanaco* model family was developed, demonstrating the potential of this approach. Below, I outline the core concepts and steps followed in fine-tuning **Llama 3.2**.

4.5.1 Background

4.5.2 Block-wise k-bit Quantization

- The input data type is commonly rescaled into the target data type range through normalization by the absolute maximum of the input elements, which are usually structured as a tensor.
- For example, quantizing a 32-bit floating-point (FP32) tensor into an Int8 tensor with a range of $[-127, 127]$:

$$X^{\ln 8} = \text{round}\left(\frac{127}{\text{absmax}(X^{\text{FP32}})} X^{\text{FP32}}\right) = \text{round}(c^{\text{FP32}}.X^{\text{FP32}}) \tag{4.1}$$

  where c is the quantization constant or quantization scale.

- De-quantization is the inverse:

$$\text{dequant}(c^{FP32}, X^{ln8}) = \frac{X^{ln8}}{c^{FP32}} = X^{FP32} \tag{4.2}$$

4.5.3 Low-rank Adapter (LoRA) Finetuning

LoRA: Only train A and B. (Figure from LoRA) With the full model parameters W which remain fixed, LoRA augments a linear projection through an additional factorized projection. Given a projection $X.W = Y$:

$$Y = XW + sXL_1L_2 \tag{4.3}$$

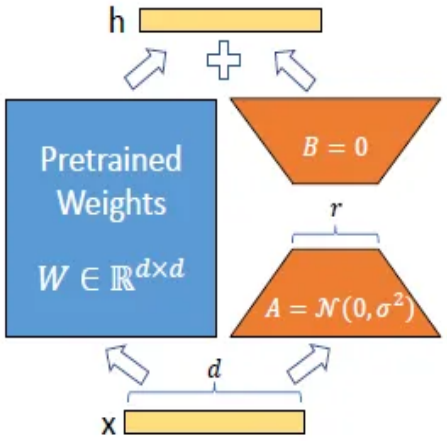


FIGURE 4.1: Our re-parametrization. We only train A and B. ([24])

4.5.4 QLoRA: Memory Optimization Techniques

QLoRA further reduces the **memory requirement** and achieves high-fidelity **4-bit finetuning** using two techniques — **4-bit NormalFloat (NF4) quantization** and **Double Quantization**. Additionally, *Paged Optimizers* are introduced to prevent memory spikes.

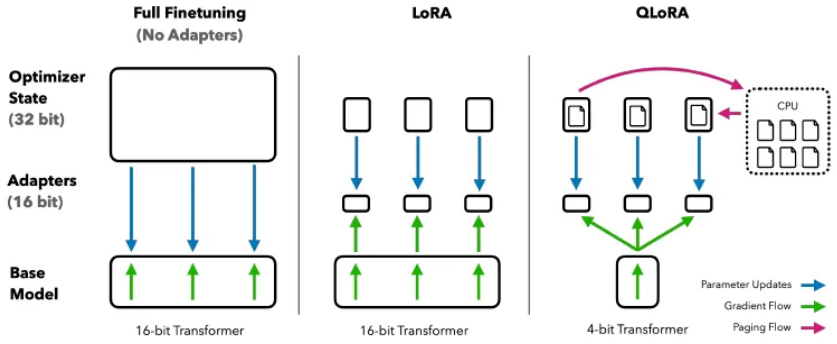


FIGURE 4.2: Full Finetuning, LoRA & QLoRA

4-bit NormalFloat Quantization

The **NormalFloat (NF4)** data type builds on *Quantile Quantization*. However, the process of quantile estimation can be computationally expensive. Since pretrained neural network weights typically follow a **zero-centered normal distribution**

with a standard deviation σ, all weights can be scaled to fit within a single fixed distribution. This is achieved by adjusting σ such that the distribution conforms to the desired range, typically $[-1, 1]$:

- Estimate the $2^k + 1$ quantiles of a theoretical $N(0, 1)$ distribution to obtain a k-bit quantile quantization data type tailored for normal distributions.
- Normalize these values into the range $[-1, 1]$.
- Quantize an input weight tensor by rescaling it into the $[-1, 1]$ range through *absolute maximum* normalization.

Formally, the 2^k values q_i of the data type are estimated as follows:

$$q_i = Q_X\left(\frac{i}{2^k}\right) \tag{4.4}$$

where $Q_X(\cdot)$ is the quantile function of the standard normal distribution $N(0, 1)$.

Double Quantization (DQ)

Double Quantization applies quantization in two steps. The constants c_2^{FP32} from the initial quantization serve as inputs for a second quantization. This secondary process produces **quantized quantization constants**, labeled as c_2^{FP8}, and a second level of quantization constants, c_1^{FP32}. In this step, 8-bit floating points are used, with a **blocksize** of 256 for enhanced memory efficiency.

Paged Optimizers

Paged Optimizers leverage NVIDIA's unified memory feature, enabling **automatic page-to-page transfers** between the CPU and GPU. When GPU memory is fully

utilized, data is automatically *evicted to CPU RAM* and subsequently transferred back to GPU memory as required during the optimizer update step.

QLoRA for a Single Linear Layer

With the components defined above, QLoRA applied to a single linear layer in the quantized base model with a single **LoRA** ([24]) adapter is as follows:

$$\mathbf{Y}^{\text{BF16}} = \mathbf{X}^{\text{BF16}} \cdot \text{doubleDequant}(c_1^{\text{FP32}}, c_2^{\text{k-bit}}, \mathbf{W}^{\text{NF4}}) \quad + \mathbf{X}^{\text{BF16}} \cdot \mathbf{L}_1^{\text{BF16}} \cdot \mathbf{L}_2^{\text{BF16}}$$

$$\begin{aligned} \text{doubleDequant}(c_1^{\text{FP32}}, c_2^{\text{k-bit}}, \mathbf{W}^{\text{NF4}}) &= \text{dequant}\left(\text{dequant}(c_1^{\text{FP32}}, c_2^{\text{k-bit}}), \mathbf{W}^{\text{4bit}}\right) \\ &= \mathbf{W}^{\text{BF16}} \end{aligned} \tag{4.5}$$

- **NF4** is utilized for W, with a blocksize of 64 for finer quantization precision.
- **FP8** is used for c_2, with a blocksize of 256 to conserve memory.

To summarize, **QLoRA** features both a **storage data type** (usually 4-bit NormalFloat) and a **computation data type** (16-bit BrainFloat). During forward and backward passes, QLoRA dequantizes the storage data type to the computation data type, but only *computes weight gradients* for the LoRA parameters, which use 16-bit BrainFloat.

4.5.5 Guanaco Results

4-bit NormalFloat (NF4) outperforms traditional 4-bit Floating Point (FP4) in terms of performance.

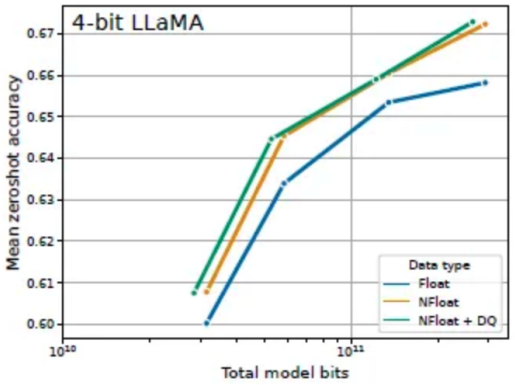


FIGURE 4.3: Mean Zero-Shot Accuracy across various benchmarks.

The performance of LLaMA models is evaluated using different 4-bit data types on WinoGrande, HellaSwag, PIQA, Arc-Easy, and Arc-Challenge. NF4 demonstrates significant bit-for-bit accuracy improvements compared to FP4. Additionally, while Double Quantization (DQ) offers minimal gains in accuracy, it provides better memory management for larger models (33B/65B) on GPUs with limited memory (24/48GB).

TABLE 4.1: Mean Perplexity (PPL) for various data types.

Data Type	**Mean PPL**
Int4	34.34
Float4 (E2M1)	31.07
Float4 (E3M0)	29.48
NFloat4 + DQ	27.41

The table shows the mean perplexity (PPL) for different data types across Pile Common Crawl datasets, evaluated on models ranging from 125M to 13B parameters (OPT, BLOOM, LLaMA, and Pythia). The NF4 data type matches the performance of 16-bit full fine-tuning and LoRA fine-tuning on academic benchmarks while effectively reducing memory usage without compromising performance.

The results highlight that 4-bit QLoRA with NF4 matches the performance of 16-bit fine-tuning on well-established academic benchmarks. Furthermore, NF4 outperforms FP4, while DQ facilitates a reduced memory footprint without performance

degradation.

Table 4.2: Mean 5-Shot MMLU Accuracy

LLaMA Size	**7B**		**13B**		**33B**		**65B**		**Mean**
Dataset	**Alpaca**	**FLAN v2**	**Alpaca**	**FLAN v2**	**Alpaca**	**FLAN v2**	**Alpaca**	**FLAN v2**	
BFloat16	38.4	45.6	47.2	50.6	57.7	60.5	61.8	62.5	53.0
Float4	37.2	44.0	47.3	50.0	55.9	58.5	61.3	63.3	52.2
NFloat4 + DQ	39.0	44.5	47.5	50.7	57.3	59.2	61.8	63.9	53.1

The above table enlists mean 5-shot MMLU test accuracy for LLaMA 7-65B models finetuned with adapters on Alpaca and FLAN v2 for different data types. Overall, NF4 with double quantization (DQ) matches BFloat16 performance, while FP4 is consistently one percentage point behind both.

Summarizing all the inferences drawn from these results:

- 4-bit NormalFloat (NF4) outperforms 4-bit Floating Point (FP4).
- NF4 significantly enhances performance over FP4 and Int4, while double quantization minimizes memory usage without sacrificing performance.
- 4-bit QLoRA using NF4 achieves performance equivalent to 16-bit full fine-tuning and LoRA fine-tuning on established academic benchmarks.
- NF4 demonstrates greater effectiveness than FP4, with double quantization maintaining performance levels.

Guanaco: QLoRA Trained on OASST1 Achieves State-of-the-Art Chatbot Performance

The Guanaco 65B model, fine-tuned on a modified OASST1 dataset, achieves leading performance among open-source models according to both automated and human evaluations.

Table 1 shows Elo ratings averaged over 10,000 random initial orderings. The winner of each match is determined by GPT-4, which evaluates and selects the better

TABLE 4.3: Elo ratings from a competition between models.

Model	Size	Elo
GPT-4	-	1348 ± 1
Guanaco 65B	41 GB	1022 ± 1
Guanaco 33B	21 GB	992 ± 1
Vicuna 13B	26 GB	974 ± 1
ChatGPT	-	966 ± 1
Guanaco 13B	10 GB	916 ± 1
Bard	-	902 ± 1
Guanaco 7B	6 GB	879 ± 1

response for prompts from the Vicuna benchmark. Guanaco 65B and 33B models achieve the most wins after GPT-4, with Guanaco 13B scoring higher than Bard.

The Guanaco 65B model is the top-performing open-source chatbot, achieving performance close to ChatGPT.

TABLE 4.4: Zero-shot Vicuna benchmark scores as a percentage of the score obtained by ChatGPT evaluated by GPT-4. We see that OASST1 models perform close to ChatGPT despite being trained on a very small dataset and having a fraction of the memory requirement of baseline models.

Model / Dataset	Params	Model bits	Memory	ChatGPT vs Sys	Sys vs ChatGPT	Mean	95% CI
GPT-4	-	-	-	119.4%	110.1%	114.5%	2.6%
Bard	-	-	-	93.2%	96.4%	94.8%	4.1%
Guanaco	65B	4-bit	41 GB	96.7%	101.9%	99.3%	4.4%
Alpaca	65B	4-bit	41 GB	63.0%	77.9%	70.7%	4.3%
FLAN v2	65B	4-bit	41 GB	37.0%	59.6%	48.4%	4.6%
Guanaco	33B	4-bit	21 GB	96.5%	99.2%	97.8%	4.4%
Open Assistant	33B	16-bit	66 GB	73.4%	85.7%	78.1%	5.3%
Alpaca	33B	4-bit	21 GB	67.2%	79.7%	73.6%	4.2%
FLAN v2	33B	4-bit	21 GB	26.3%	49.7%	38.0%	3.9%
Vicuna	13B	16-bit	26 GB	91.2%	98.7%	94.9%	4.5%
Guanaco	13B	4-bit	10 GB	87.3%	93.4%	90.4%	5.2%
Alpaca	13B	4-bit	10 GB	63.6%	76.7%	69.4%	4.2%
HH-RLHF	13B	4-bit	10 GB	55.5%	69.1%	62.5%	4.7%
Unnatural Instr.	13B	4-bit	10 GB	50.6%	69.8%	60.5%	4.2%
Chip2	13B	4-bit	10 GB	49.2%	69.3%	59.5%	4.7%
Longform	13B	4-bit	10 GB	48.9%	76.0%	53.6%	5.6%
Self-Instruct	13B	4-bit	10 GB	38.0%	60.5%	49.1%	4.6%
FLAN v2	13B	4-bit	10 GB	32.4%	61.2%	47.0%	3.6%
Guanaco	7B	4-bit	5 GB	84.1%	89.8%	87.0%	5.4%
Alpaca	7B	4-bit	5 GB	57.3%	71.2%	64.4%	5.0%
FLAN v2	7B	4-bit	5 GB	33.3%	56.1%	44.8%	4.0%

Compared to GPT-4, Guanaco 65B and 33B exhibit an estimated win probability of 30%, based on Elo ratings from human evaluators assessing pairwise responses on the Vicuna benchmark. These findings underscore the effectiveness of 4-bit QLoRA for training state-of-the-art chatbots that approach the performance of ChatGPT.

Notably, the Guanaco 33B model can be fine-tuned on 24 GB consumer GPUs in under 12 hours.

4.6 Experiments

In this section, we outline the experiments to be performed as part of the evaluation. The experiments will focus on assessing the performance of various fine-tuned versions of the Llama 3.2 model on several benchmark tasks. The tasks include `MMLU`, `ROUGE`, `ARC`, `SQyAD`, and `HellaSwag`. For each of these tasks, we will perform inference on the following models:

1. **Llama 3.2:** The base Llama 3.2 model will be used as a control to assess the impact of fine-tuning.
2. **Llama 3.2 UltraChat-200k Fine-tuned:** This version of the Llama 3.2 model has been fine-tuned on the UltraChat dataset.
3. **Llama 3.2 UltraChat-200k + First Scenario (v1):** This fine-tuned version combines the UltraChat dataset with the first scenario, denoted as version 1 (v1).
4. **Llama 3.2 UltraChat-200k + Second Scenario (v2):** In this case, the Llama 3.2 UltraChat model is further fine-tuned with the second scenario, denoted as version 2 (v2).
5. **Llama 3.2 UltraChat-200k + First + Second Combined Scenario (v3):** This version incorporates both the first and second scenarios, creating a combined fine-tuned model, denoted as version 3 (v3).

The results will be in **Results and Observations** (6) Section. For Scenario-related details regarding the conversations involved in these experiments, please refer to the Appendix.

Chapter 5

Implementation

5.1 Front-End and Schema Design

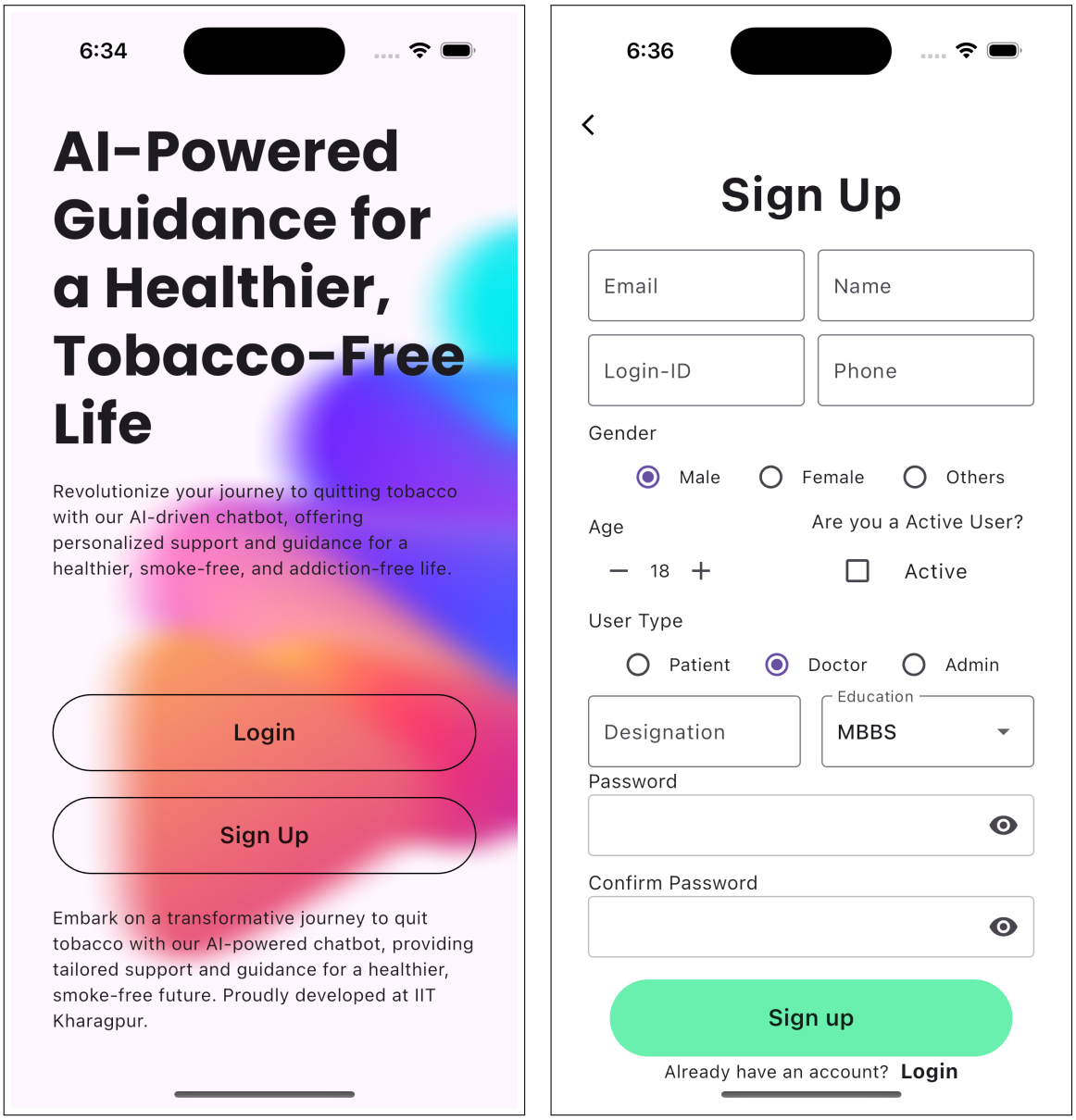
The front-end of the mobile application is designed to be intuitive, dynamic, and user-friendly, ensuring smooth navigation. Upon launching the application, users are presented with a **Home** screen that offers options for both **Login** and **Sign-Up**.

5.2.2 Home Page

The Home Page features a bold title to introduce the system to patients, along with a brief description to explain its purpose. The affiliation with IIT Kharagpur is also displayed to build user trust. Links to the login and sign-up pages are provided for easy access.

5.2.3 Login and Sign-Up Page

New users must register by providing a unique **login ID**, **password**, **age**, and **user type**. Registered users can log in by entering their login ID and password.



(a) Home Screen (b) Sign-Up Screen

FIGURE 5.1: Screenshots of the application's Home and Sign-Up screens.

5.2.4 User ID Tracking and Registration Statistics

- To track user activities and maintain up-to-date registration stats, a separate table, `reg_stats`, logs all users registered each day.
- The `user_id` is dynamically updated for each new registration to ensure a unique identifier.

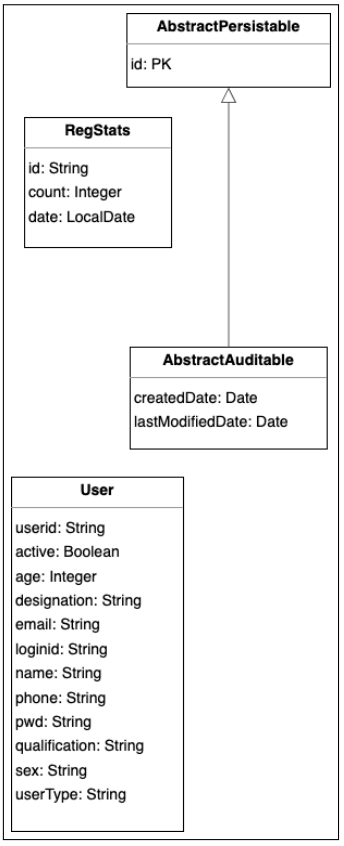


FIGURE 5.2: Database Schema Diagram

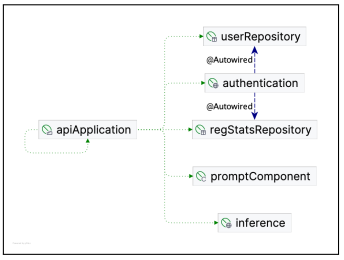


FIGURE 5.3: Context Dependency Diagram for Back-End Architecture

5.2 Spring Back-End and API Endpoints

The **SmokeCtrl** application integrates advanced large language models within a mobile interface to provide intelligent responses to medical queries. This setup involves configuring the development environment, managing data flows, and handling model integration and inference.

5.2.1 Authentication Endpoints

Two core API endpoints manage authentication: **Sign-Up** for user registration and **Login** for user verification, with key functionalities as follows:

Sign-Up Endpoint:-

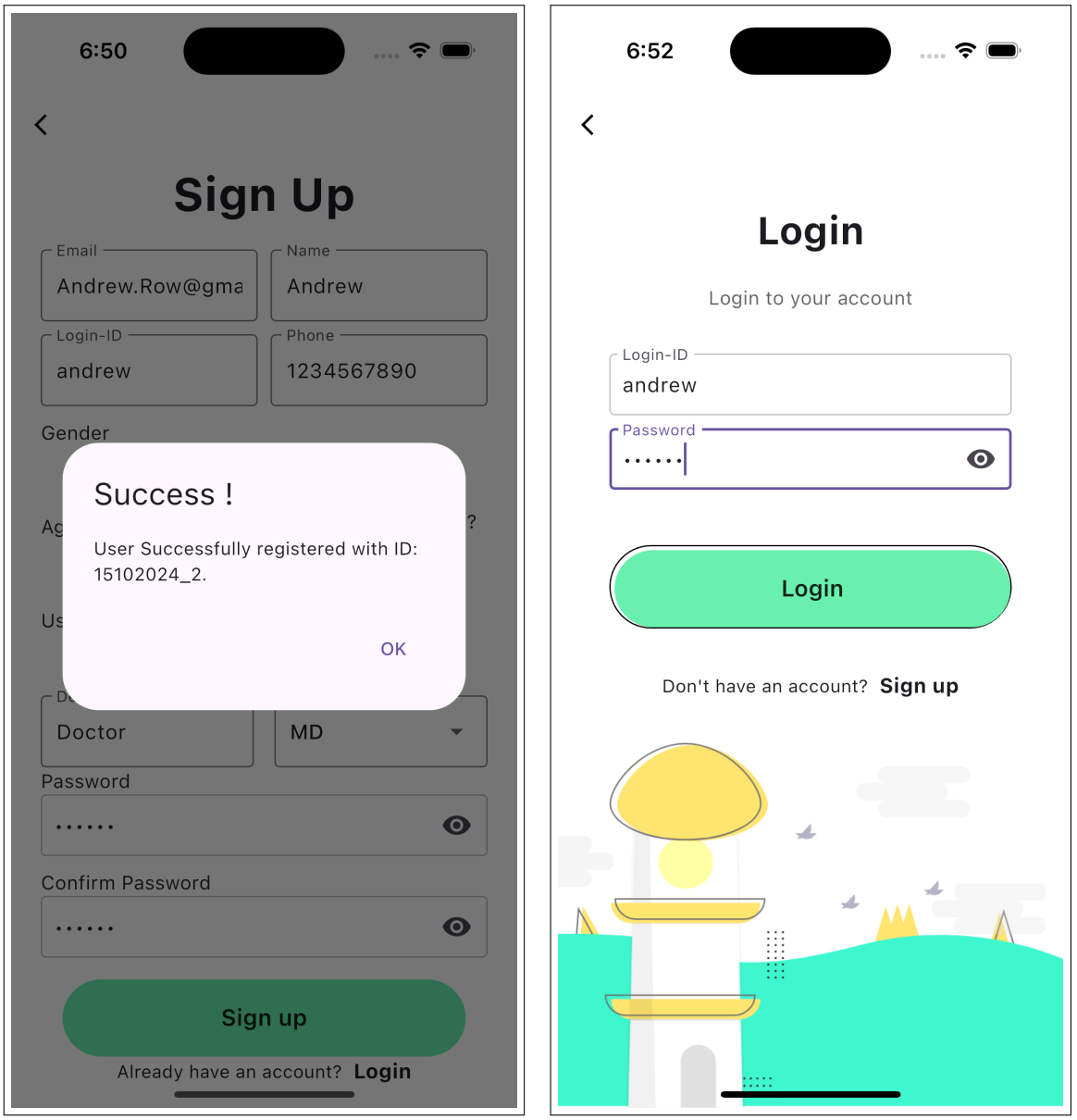
- Handles user registration with data validation.
- Ensures unique `login-id` to prevent duplicates.
- Enforces client-side validation for data integrity.

Login Endpoint:-

- Authenticates users by validating credentials.
- Sends essential user data (e.g., age, gender, education, user type, phone number) upon successful authentication via a REST response.

Endpoints for Llama 3.2 Inference

- **Purpose and Description:** This script provides an endpoint for interacting with the **Llama 3.2** model, designed to generate responses based on user-provided prompts.



(a) Successful Sign-up (b) Login in Action

FIGURE 5.4: Screenshots of the application's Home and Sign-Up screens.

- **Argument Parsing:** Uses ARGPARSE to manage command-line arguments, allowing users to specify parameters such as the prompt, model path, and embedding model path. Flags like `-e` for re-creating embeddings and `-v` for verbose output enhance the script's flexibility and usability.

- **Embedding Handling:** When the `-e` flag is activated, the script generates embeddings using the specified embedding model and documents in a designated directory, a process essential for preparing the model for accurate response generation.
- **Model Initialization:** Configures the Llama 3.2 model by setting paths for both the model and embedding files, and specifies the embedding model name, ensuring correct model initialization for response generation.
- **Asynchronous Execution:** Leverages Python's `asyncio` library to run the model asynchronously, enabling efficient operation handling without blocking the main execution thread.
- **Output:** Produces the model's response to a specified prompt, providing an interactive command-line interface for direct user interaction with Llama 3.2.

5.3 Retrieval-Augmented Inference using LangChain

The `LLModel` class in this Python module employs the **LangChain** framework to implement a RAG (Retrieval-Augmented Generation) setup for generating informed responses from a designated data source. Key components are outlined below:

- **Model Component Initialization**: The `LLModel` class is initialized with paths to the `LlamaCpp` model and a `Chroma` database, which stores embeddings generated by HuggingFace's SENTENCE-TRANSFORMERS/ALL-MPNET-BASE-V2, facilitating efficient retrieval.
- **Chroma Database Configuration**: Configured within `LangChain`, `Chroma` serves as the retrieval database, holding vector representations of data that enable rapid identification of relevant context for incoming queries.

- **Callback Management**: A `CallbackManager` with `StreamingStdOutCallbackHandler` provides real-time output management, crucial for debugging and performance monitoring.

- **Model and Chain Setup**: The `LlamaCpp` model is configured with parameters such as GPU layers, batch size, and context length to optimize prompt processing and leverage **Apple's Metal GPU**. The model is integrated with **LLMChain** (`LangChain`) through a `PromptTemplate` that structures input into sections (system, user, assistant) for context-aware responses.

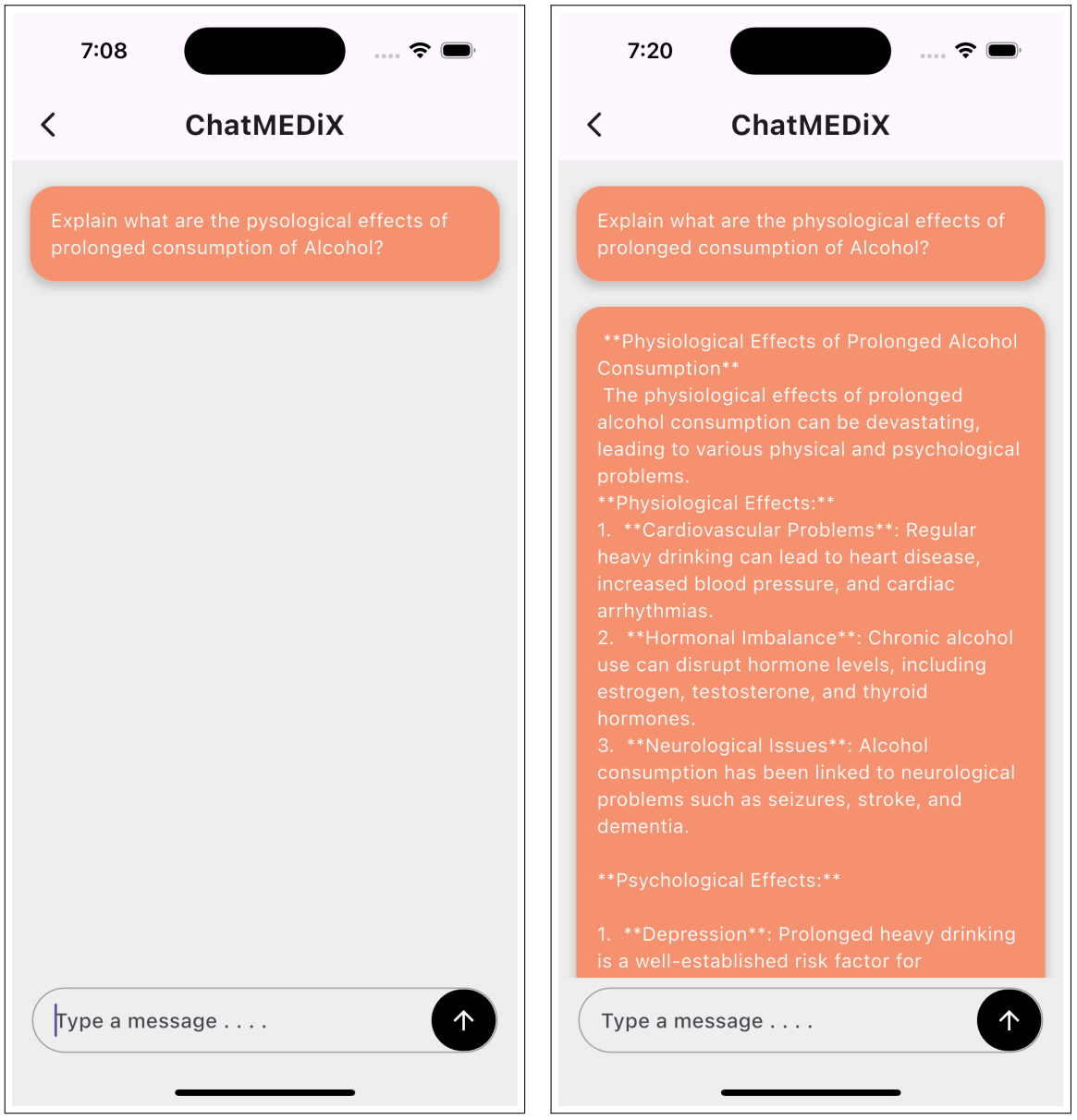
- **Execution Flow**: The `run` method initiates the RAG process by performing a similarity search in the `Chroma` database, retrieving the most relevant documents based on the query's embeddings. The retrieved context and user query are then fed into **LangChain**, where `LlamaCpp` generates a response that is both accurate and contextually enhanced.

- **Asynchronous Execution**: The model operates asynchronously to support non-blocking execution, optimizing scalability for concurrent query processing.

5.4 Fine-tuning Llama-3.2-1b-instruct using qLoRA

- **qLoRA Fundamentals**: qLoRA (Query-based Low-Rank Adaptation) is employed to apply low-rank adaptation selectively to transformer projections (e.g., `q_proj`, `k_proj`, `v_proj`, `o_proj`). This technique focuses on the *attention mechanism*, ensuring efficiency by modifying only the essential components.

- **Key Parameter Configuration**: The `LoraConfig` specifies two key parameters: *lora_alpha*, which adjusts the strength of adaptation, and `r` (rank), which governs the matrix rank and helps balance memory utilization with task-specific optimization.



(a) Asking a Query (b) Generated Response

FIGURE 5.5: Screenshots of a Sample Response being generated.

- **Model Initialization**: The pre-trained **Llama 3.2 model** and tokenizer are loaded to prepare the model for applying low-rank adaptation to specific transformer layers. This setup is crucial for the subsequent application of qLoRA.

- **Dataset Preparation**: The dataset is structured with roles and conversation

IDs to maintain coherent dialogue flow. Preprocessing ensures the data is formatted appropriately for accurate adaptation during training.

- **Training with `SFTTrainer`**: The training process is carried out using `SFTTrainer`, where parameters such as learning rate and batch size are adjusted. *qLoRA* ensures that only the essential projections are updated, optimizing resource use and improving performance for the targeted tasks.

Chapter 6

Results and Observations

6.1 Results and Observations

This section presents an overview of the performance metrics utilized to evaluate the effectiveness of the Llama 3.2 and their Fine-Tuned Adversaries, specifically focusing on ROGUE, MMLU, ARC, SQuAD, and Hellaswag.

6.1.1 ROGUE

The ROGUE (Recall-Oriented Understudy for Gisting Evaluation) metric is employed primarily for assessing the quality of text summarization. It calculates the overlap between the n-grams of the generated summary and the reference summaries, providing insights into the model's ability to capture essential information concisely. Higher ROGUE scores indicate better performance in generating coherent and relevant summaries.

6.1.2 MMLU

The MMLU (Massive Multitask Language Understanding) benchmark evaluates a model's general language understanding capabilities across a diverse range of tasks. It encompasses multiple-choice questions that assess various aspects of reasoning, comprehension, and knowledge. The results on this benchmark highlight the model's effectiveness in handling complex queries, with higher scores suggesting superior performance in natural language understanding.

6.1.3 ARC

The ARC (AI2 Reasoning Challenge) metric is specifically designed to evaluate a model's reasoning capabilities. It consists of multiple-choice questions that require inference and knowledge application. The performance of the Llama 3.2 model on ARC demonstrates its aptitude for understanding and reasoning through textual information, contributing to its overall reliability in educational and knowledge-driven applications.

6.1.4 SQuAD

The SQuAD (Stanford Question Answering Dataset) benchmark is a widely recognized metric for evaluating question-answering systems. It measures the accuracy of the model's responses to questions based on a provided context. The Llama 3.2 model's results on SQuAD reflect its proficiency in extracting relevant information and generating accurate answers, indicating robust performance in comprehension tasks.

6.1.5 Hellaswag

Hellaswag is a benchmark designed to assess the ability of models to predict plausible continuations of a given context. It challenges the model to differentiate between coherent and incoherent statements, providing insights into its understanding of narrative structure and context. The performance metrics from Hellaswag illustrate the model's competence in handling contextual inference, which is critical for conversational applications.

In summary, the Llama 3.2 model exhibits promising results across these diverse evaluation metrics, indicating its potential for real-world applications in natural language processing and understanding. For our project, we will be analyzing ROGUE to measure how much the generated response resembles the response in

6.2 Benchmarks

6.2.1 Base Pretrained LLaMA 3.2 Models

Category	**Benchmark**	**# Shots**	**Metric**	**Llama 3.2 1B**	**Llama 3.2 3B**
General	MMLU	5	macro_avg/acc_char	32.2	58
	AGIEval English	3-5	average/acc_char	23.3	39.2
	ARC-Challenge	25	acc_char	32.8	69.1
Reading comprehension	SQuAD	1	em	49.2	67.7
	QuAC (F1)	1	f1	37.9	42.9
	DROP (F1)	3	f1	28.0	45.2
Long Context	Needle in Haystack	0	em	96.8	1

TABLE 6.1: Benchmark results for Llama models across various tasks.

Capability	Benchmark	# Shots	Metric	1B bf16	1B QLoRA	3B bf16	3B PTQ
General	MMLU	5	macro_avg/acc	49.3	49.0	63.4	60.5
Re-writing	Open-rewrite eval	0	micro_avg/rougeL	41.6	41.2	40.1	40.3
Summarization	TLDR9+ (test)	1	rougeL	16.8	16.8	19.0	19.1
Instruction following	IFEval	0	Avg	59.5	55.6	77.4	73.9
Math	GSM8K (CoT)	8	em_maj1@1	44.4	46.5	77.7	72.9
	MATH (CoT)	0	final_em	30.6	31.0	48.0	44.2
Reasoning	ARC-C	0	acc	59.4	60.7	78.6	75.6
	GPQA	0	acc	27.2	25.9	32.8	32.8
	Hellaswag	0	acc	41.2	41.5	69.8	66.3
Tool Use	BFCL V2	0	acc	25.7	23.7	67.0	53.4
	Nexus	0	macro_avg/acc	13.5	12.5	34.3	32.4
Long Context	InfiniteBench/En.QA	0	longbook_qa/f1	20.3	N/A	19.8	N/A
	InfiniteBench/En.MC	0	longbook_choice/acc	38.0	N/A	63.3	N/A
	NIH/Multi-needle	0	recall	75.0	N/A	84.7	N/A
Multilingual	MGSM (CoT)	0	em	24.5	24.4	N/A	N/A

TABLE 6.2: Benchmark results for various Llama 3.2 models across different tasks and configurations.

6.2.2 Curated Fine-Tuned LLaMA 3.2 Models

Category	MMLU	ARC	SQuAD	Hellaswag	Rogue-L	Rogue-2
Llama 3.2 (1B)	49.3	59.4	49.2	41.2	0.031356	0.0070013
Llama 3.2 (3B)	63.4	78.6	67.7	69.8	-	-
LLama 3.2 (1B) sce.v.1	47.5	55.2	46.7	39.1	0.041926	0.0098073
LLama 3.2 (1B) sce v.2	45.3	52.0	44.5	37.0	0.036292	0.0072468
LLama 3.2 (1B) sce v.c	46.4	53.1	45.5	38.0	0.038109	0.0084276

TABLE 6.3: Benchmark results for the Experiments conducted by me during the course of Project. Note that the sce.v.1..3 are LLaMA 3.2 Fine-tuned first on **UltraChat-200k** ([25]) and then on Conversation Set-1, Conversation Set-2 and Combined Version respectively

Chapter 7

Conclusions

7.1 Conclusion

The **SmokeCtrl** project represents a significant advancement in leveraging mobile technology and **artificial intelligence** for **tobacco cessation**. Utilizing a **Flutter-based interface** and a **Spring** back-end integrated with **large language models (LLMs)**, SmokeCtrl provides tailored support for individuals seeking to quit smoking. Its robust design ensures secure, personalized interactions, positioning it as both a cessation aid and an *educational resource* on addiction.

SmokeCtrl's adaptive responses and user-centric design distinguish it from traditional cessation methods, offering unique, evidence-based feedback based on user input. By incorporating *activity tracking* and *daily statistics*, the platform can dynamically adjust its services based on engagement patterns, enhancing its effectiveness.

The choice of a **cross-platform Flutter interface** maximizes accessibility, while the **Spring-based back-end** ensures scalability and efficient data handling. Defined authentication endpoints enable a secure, user-friendly experience, safeguarding *user data* and privacy throughout the cessation journey.

From a *data analysis perspective*, SmokeCtrl's logging of *user activity* provides insights that inform both ongoing improvements and the development of new features, reinforcing a commitment to *continuous enhancement* based on real user behavior.

While challenges were encountered, particularly in balancing *response accuracy* with computational efficiency, the successful deployment of SmokeCtrl highlights the project's potential impact on public health. Initial feedback underscores the app's intuitive *interface* and helpfulness, supporting its role as an empowering tool for long-term behavior change.

In summary, **SmokeCtrl** serves as a promising prototype for AI-driven health applications targeting *behavior modification* and *addiction support*. With ongoing data collection and analysis, SmokeCtrl will continue evolving, adapting to meet the diverse needs of users and further supporting *tobacco cessation* with empathy and sophistication.

Chapter 8

Future Work

To further enhance **SmokeCtrl's** capabilities and reach, we outline several potential areas for future development:

- **Enhanced Personalization Algorithms**: Expanding the AI's ability to adapt responses based on individual user profiles and behaviors can make **SmokeCtrl's** support even more tailored and impactful. Machine learning models trained on user feedback could refine response generation, offering advice that closely aligns with each user's cessation journey.
- **Predictive Analytics for Relapse Prevention**: By analyzing *user behavior* and *engagement patterns*, **SmokeCtrl** could predict when a user is at higher risk of *relapse*. This feature could trigger proactive interventions, such as reminders or supportive messages, helping users stay on track during challenging moments.
- **Expanded Multilingual Support**: Implementing additional *language options* would allow **SmokeCtrl** to reach a broader, more diverse user base. *Multilingual support* could make **SmokeCtrl** accessible to non-English-speaking populations, promoting *tobacco cessation* on a global scale.

- **Community Support Features**: Creating a *community space* where users can connect with others on similar journeys may add an element of *peer support*. This could include *discussion forums*, *group challenges*, or shared progress tracking, which could enhance *motivation* and *accountability*.

- **Advanced Health Data Tracking and Analytics**: Collecting more granular data on *user engagement* and *health outcomes* could allow for a deeper analysis of what strategies are most effective for different user profiles. This would enable **SmokeCtrl** to dynamically adjust its approach, improving *cessation success rates*.

- **Clinical Validation and Partnerships**: Collaborating with healthcare providers and researchers to validate **SmokeCtrl's** efficacy could enhance its *credibility* as a cessation tool. Additionally, partnerships with *clinics* or *wellness programs* could facilitate more comprehensive support networks, providing users with access to in-person resources if needed.

By pursuing these future directions, **SmokeCtrl** can evolve from a mobile application into a comprehensive, *data-driven platform* for *tobacco cessation*, offering continuous, *personalized support* for those striving for a smoke-free life.

Bibliography

[1] Vaswani, A., et al. (2017). *Attention is All You Need.*

[2] Devlin, J., et al. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.*

[3] Radford, A., et al. (2018). *Improving Language Understanding by Generative Pre-Training.*

[4] Alsentzer, E., et al. (2019). *Publicly Available Clinical BERT Embeddings.*

[5] Brown, T., et al. (2020). *Language Models are Few-Shot Learners.*

[6] Raffel, C., et al. (2020). *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.*

[7] OpenAI. (2022). *ChatGPT: Optimizing Language Models for Dialogue.*

[8] Thoppilan, R., et al. (2022). *LaMDA: Language Models for Dialogue Applications.*

[9] Lee, J., et al. (2019). *BioBERT: A Pre-trained Biomedical Language Representation Model for Biomedical Text Mining.*

[10] Huang, K., et al. (2020). *ClinicalBERT: A Pre-trained Language Representation Model for Clinical Notes.*

[11] Hu, E. J., et al. (2021). *LoRA: Low-Rank Adaptation of Large Language Models.*

[12] Dettmers, T., et al. (2023). *Quantized Low-Rank Adaptation for Efficient Model Fine-Tuning.*

[13] Houlsby, N., et al. (2019). *Parameter-Efficient Transfer Learning for NLP.*

[14] Lewis, P., et al. (2020). *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.*

[15] Reimers, N., and Gurevych, I. (2020). *Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation.*

[16] Google. (2020). *Flutter: Beautiful native apps in record time.*

[17] Facebook. (2015). *React Native: A Framework for Building Native Apps using React.*

[18] Johnson, R., et al. (2003). *The Spring Framework: Simplifying Java Development.*

[19] Django Software Foundation. (2005). *Django: The Web framework for perfectionists with deadlines.*

[20] Marcolino, M. S., et al. (2018). *The Impact of mHealth Interventions on Health Behavior: A Systematic Review.*

[21] Weitzman, E., and Kaci, L. (2019). *Metrics for Evaluating Machine Learning in Healthcare: A Review.*

[22] Rinaldi, A., et al. (2021). *AI for Behavioral Health: Applications, Challenges, and Prospects.*

[23] Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. (2023). *QLoRA: Efficient Finetuning of Quantized LLMs.* arXiv preprint arXiv:2305.14314. Retrieved from https://arxiv.org/abs/2305.14314.

[24] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2022). *LoRA: Low-Rank Adaptation of Large Language Models.* In *Proceedings of the International Conference on Learning Representations.* Retrieved from https://openreview.net/forum?id=nZeVKeeFYf9.

[25] N. Ding, Y. Chen, B. Xu, Y. Qin, Z. Zheng, S. Hu, Z. Liu, M. Sun, and B. Zhou, *Enhancing Chat Language Models by Scaling High-quality Instructional Conversations*, arXiv preprint arXiv:2305.14233, 2023.

[26] OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J. (2024). *GPT-4 Technical Report.* arXiv preprint arXiv:2303.08774. Retrieved from https://arxiv.org/abs/2303.08774.