SOFTWARE REQUIREMENTS SPECIFICATION

for

Intelligent FAQ Chatbot: A User-centric Approach using Large Language Models

Under the supervision of

Dr. B. SAI JYOTHI **M. Tech, Ph.D.**

Team: C2

21BQ1A12G5 – U. VYSHNAVI

21BQ1A12H7 – Y. KAVYA VASANTHI

21BQ1A12C9 – R. SAI DEEPIKA

21BQ1A12D1 – G. RAMYA SINDHU

Table Of Contents

1. Introduction

2. Overall Description

3. External Interface Requirements

4. System Features

5. Non-functional Requirements

1. Introduction

1.1 Purpose

The purpose of this SRS document is to outline the requirements for developing an FAQ Chatbot using Large Language Models (LLM) to provide automated and intelligent responses to Frequently Asked Questions (FAQ's). It aims to support customers, education and knowledge dissemination across various domains by leveraging Natural Language Processing (NLP) Techniques. It is designed to improve user experience, provide quick and accurate answers and enhance accessibility to information.

1.2 Intended Audience and Reading Suggestions

The intended audience for this FAQ chatbot includes students, customers, and businesses in education, healthcare, and e-commerce aiming to enhance customer support through intelligent automation, as well as researchers and developers exploring NLP and deep learning applications. For further understanding, recommended readings include Natural Language Processing with Deep Learning and Speech and Language Processing for foundational NLP and GPT concepts, and Deep Learning for Natural Language Processing for practical implementation. Readers should explore the design and implementation of FAQ chatbots, focusing on Natural Language Understanding (NLU), knowledge base integration, and user interaction strategies.

1.3 Project Scope

The project aims to develop an Intelligent FAQ Chatbots using Large Language Models (LLMs) to provide accurate and context-aware answers to frequently asked questions. The chatbot will focus on delivering seamless user interaction, ensuring quick and precise responses across various domains based on pre-defined and dynamically updated knowledge bases

1.4 References

- FAQ (Frequently Asked Questions) Chatbot for Conversation :

  https://www.researchgate.net/publication/345177573_FAQ_Frequently

  _Asked_Questions_ChatBot_for_Conversation_

- FAQ chatbot and inclusive learning in massive open online courses:

  https://www.sciencedirect.com/science/article/abs/pii/S0360131521002724

- AI text generators and Producers
  https://ieeexplore.ieee.org/document/9853785

- AI Chatbot for Answering FAQ’s
  https://ieeexplore.ieee.org/document/9908774

- About the datasets:

  For this project no precise datasets are available. To obtain the necessary information , data must be merged from several Sources.

- https://www.kaggle.com/datasets/saadmakhdoom/ecommerce-faq-chatbot-dataset/data

- https://www.kaggle.com/datasets/narendrageek/mental-health-faq-for-chatbot

- https://paperswithcode.com/search?q_meta=&q_type=&q=FAQ+datas

2. Overall Description

2.1 Project Perspective

This project aims to improve user engagement and support by leveraging advanced AI technology to provide accurate and context-aware responses in real-time. It addresses user inquiries efficiently, reduces the manual effort, and ensures consistent information delivery. The FAQ chatbot enhances user experience by ensuring accessibility and data confidentiality.

2.2 Project Functions

1. **Data Collection:** Gather a variety of frequently asked questions (FAQs) and their answers, along with conversation logs that reflect user queries and responses.
2. **Data Preprocessing:** Clean, preprocess, and format the collected data to make it suitable for the system. This may involve handling missing values, data normalization, tokenization and feature engineering.
3. **Model Training:** Fine-tune LLMs using domain-specific datasets. Divide the data into training and validation sets to ensure effective learning and generalization. Optimize hyperparameters such as learning rate and batch size for maximum performance. Train the chatbot to handle complex queries and generate accurate, personalized responses by incorporating historical and user interaction data.
4. **Prediction and Model Evaluation:** Deploy the trained chatbot to interpret user queries, identify intent, and provide contextually relevant responses. Personalize predictions using user-specific data like preferences and interaction history. Evaluate performance with metrics such as accuracy, precision while leveraging user feedback to enhance scalability and reliability
5. **Visualization:** Track the system performance through simple metrics like response accuracy, common queries, and user satisfaction, helping identify areas for improvement.
6. **Feedback Mechanism:** collect feedback from users to continuously improve the system's accuracy and efficiency.

2.3 User Classes and Characteristics

1. **End Users :** End Users include customers, employees, students, and website visitors who rely on the system to quickly access information and resolve queries related to their tasks. These users interact with the chatbot via a web or mobile interface, making it an accessible tool for a wide range of individuals**.** By typing queries in natural language, users receive instant, accurate, and contextually relevant responses tailored to their needs. The chatbot adapts to individual preferences and provides a personalized experience by analyzing user history and interaction patterns.

2. **Data Scientists and Analysts:** Data Scientists and Analysts mainly focus on gathering data, selecting relevant features, and evaluating the system performance. They ensure the model is accurate and can handle diverse queries effectively.
3. **Software Developers:** Developers are essential in building the interfaces and integrate the models into software applications, websites, and other systems. They handle maintenance and optimize the chatbot for a seamless user experience.

2.4 Operating Environment

- **Local Development Environments:** Local development environments such as Jupyter Notebooks or IDEs like PyCharm or Visual Studio Code to develop and test the prediction models.
- **Operating System:** Windows 8 or above, MacOS Catalina (10.15) or above and Linux OS can also be used.

2.5 Design And Implementation Constraints

- **Data Quality and Availability:** The FAQ Chatbot depends on high-quality, updated, and well-prepared datasets. Poor or outdated data can lead to inaccurate responses, so data cleaning and consistent updates are essential.
- **Data integration:** Integrating data from many sources can be a time-consuming and complex operation that may necessitate extensive technological skill.
- **Data and device security:** User data and interactions must be protected, even if public datasets are used. Ensure secure environments for model training and safeguard user interactions against misuse.
- **Scalability:** The FAQ chatbot must handle growing user traffic efficiently. Optimize processing, use scalable cloud solutions, and regularly monitor system performance to avoid slowdowns.

2.6 User Documentation

The FAQ Chatbot user documentation offers step-by-step guidance for all users, including customers, employees, students, and website visitors, on how to log in, input queries, and receive personalized, context-aware responses. Customers can access product or service information, employees can retrieve work-related data, students can ask academic questions, and website visitors can navigate the site easily. The documentation also explains how to provide feedback by rating or correcting chatbot responses in real time, contributing to system improvement. It includes troubleshooting support to ensure a seamless user experience, highlighting the chatbot's adaptive, real-time capabilities for efficient information retrieval.

2.7 Assumptions And Dependencies

- The project assumes access to diverse, high-quality FAQ datasets for training the chatbot. It also assumes that users have a basic understanding of interacting with chatbot interfaces in web or app-based environments.
- The chatbot relies on stable internet connectivity, cloud-based AI services , and compatibility with existing platforms like customer support tools or knowledge management systems for seamless integration.

3. External Interface Requirements

3.1 User Interface

The system provides an intuitive, feature-rich interface designed to enhance user interactions. A responsive chat widget integrated into websites or applications, allowing users to effortlessly interact with the chatbot for information retrieval and support. Users can engage with the chatbot to ask questions on a wide range of topics. The system effectively interprets user queries, even when dealing with complex or context-dependent data. The interface includes a feedback mechanism where users can rate responses or provide their opinions. This feedback helps improve the system's accuracy and effectiveness over time.

3.2 Hardware Interface

End-users can access the chatbot through a variety of standard devices, including desktops, laptops, tablets, and smartphones, ensuring broad compatibility and accessibility across different platforms and screen sizes.

3.3 Software Interfaces

- **Large Language Model (**LLM): The chatbot leverages advanced Large Language Model (LLM), such as GPT, to generate context-aware responses and accurately interpret user intent, allowing for more dynamic and personalized interactions.

- **Flask:** Flask is a lightweight Python web framework ideal for developing APIs and web applications quickly and efficiently.

- **Database Management:** The system interfaces with databases to efficiently store and manage user data, interaction history, and feedback, ensuring seamless data retrieval and continuous improvement of the chatbot's performance.

3.4 Communication Interfaces

Internet Connectivity: Requires stable internet connectivity for both the client and server, with a minimum bandwidth of 2 Mbps to ensure smooth interactions.

User Feedback Mechanism: Provides users with an easy way to submit feedback through the chatbot interface, including options to correct or suggest improvements to responses.

4. System Features

4.1 Data Ingestion

The model is fed with various data sources to ensure accurate and context-aware responses to user queries.

- Predefined FAQ Data
- External Knowledge Sources
- User Interaction Data

4.2 Data Preprocessing

- This process handles cleaning, transforming, and preparing the data for effective modelling.
- Feature engineering, tokenization, and text normalization are key steps to ensure the data is suitable for the LLM.
- The process aims to enhance data quality, feature relevance, and model performance.

4.3 Historical Data Analysis

- Historical data analysis helps identify patterns in user queries and interactions.
- It is essential for improving response strategies, identifying common user needs, and refining model performance.
- It is critical in understanding User Query Patterns, behavioural insights and improving conversational flow.

4.4 Prediction Result and Visualization

- The system visualizes the chatbot's response accuracy, user engagement metrics, and response time to assess performance.
- Interaction analytics provide insights into common queries and user behavior.
- Additionally, visualizations highlight how frequently the knowledge base is updated to improve the chatbot’s responses. These metrics enable continuous optimization of the chatbot.

5. Non-functional Requirements

5.1 Performance Requirements

Real-Time Prediction:

- The chatbot should provide responses in real-time, ensuring minimal latency for user interactions. The system must be optimized to handle multiple queries simultaneously without significant delays.

Prediction Accuracy:

- Prediction accuracy refers to the chatbot's ability to provide correct and relevant answers. To ensure high accuracy, the chatbot's training data should be regularly updated, the model's parameters fine-tuned, and advanced techniques like reinforcement learning used to improve responses.

- Continuous evaluation and fine-tuning of the model are required to maintain and enhance its ability to understand and answer queries over time.

5.2 Safety Requirements

Backup and recovery:

- Regular backups of the chatbot's knowledge base and training data should be implemented to ensure no loss of critical information.

- In case of system failure, recovery mechanisms must be in place to restore the chatbot to its operational state quickly.

Error Handling:

- The system should include robust error handling to manage unexpected inputs or failure scenarios, ensuring users receive fallback responses and the system remains stable.

5.3 Security Requirements

- The Proposed System will handle public and general queries, but any sensitive user data should be protected. Privacy guidelines must be followed to ensure that no personal information is stored without consent.

- While security is not a primary concern in public interactions, the system must adhere to basic data protection protocols to ensure user trust.

5.4 Software Quality Attributes

- **Maintainability**: The system’s code and model need to be easily maintainable to accommodate new features, updates, or improvements. Regular updates to the LLM and knowledge base will ensure it remains relevant.

- **Scalability**: The system should be designed to scale and handle an increasing number of user queries or interactions without performance degradation.

5.5 Business Rules

The system should comply with any company-specific rules or guidelines regarding its use, including adherence to ethical standards in AI deployment and content management.