Docu-Gen Project

Project Report submitted in the partial fulfilment

of

(Bachelor of Technology)
In
(Artificial Intelligence)

by

Dharmik Shetty (I073)

Under the supervision of

Dr.Radhika Chapaneri

(Professor, AI Department, MPSTME)

SVKM's NMIMS University

(Deemed-to-be University)

MUKESH PATEL SCHOOL OF TECHNOLOGY MANAGEMENT & ENGINEERING (MPSTME)

Vile Parle (W), Mumbai-56

(2023-24)

CERTIFICATE

This is to certify that the project entitled **Docu-Gen Bot**, has been done by Mr. Dharmik Shetty under my guidance and supervision & has been submitted in partial fulfilment of the degree of Bachelor of Technology in Artificial Intelligence of MPSTME, SVKM's NMIMS (Deemed-to-be University), Mumbai, India.

_____ _____

Project mentor (Name and Signature) Examiner (Name and Signature)

(Internal Guide)

Date

Place: Mumbai **(HoD)** (Name and Signature)

ACKNOWLEDGEMENT

Student Name: Dharmik Shetty
B.Tech. (AI)
Roll No.: I073

ABSTRACT

The project focuses on different topics starting from details about Document Generation Bot that provides a chat platform for interaction and providing generated document. The Document Generation part of the project that is meant for SREs(Site Reliability Engineers),this part of project includes in helping the SREs with LLM powered chatbot that has the capacity of generating documents that follow the predefined guidelines of the document which has been already setup. The work mostly includes in easing the process which involves customer to provide details to SRE regarding the their needs for VMs(Virtual Machines) and mentioning various specifications that go along with it. Noting down all the details via a chatbot and providing the end user with a document that ensures it follows proper guidelines and showcases all the necessary details is the end goal of the project.

Table of Contents

List of the figures

Abbreviations

Chapter 1

Introduction

1.1 Background

The rise of Large Language Models (LLMs) has revolutionized the field of natural language processing (NLP). These powerful AI models can process and generate text with remarkable fluency and understanding. This has opened exciting possibilities for developing interactive chatbots that can engage users in natural conversation and provide them with information or complete tasks.

1.2 Motivation and Scope

Traditional support channels often struggle to keep pace with the growing demand for user self-service solutions. Chatbots offer a promising alternative, providing users with immediate access to information and troubleshooting assistance. However, the effectiveness of chatbots can be hampered by limitations in understanding complex questions or adapting to nuanced language.

This project explores a methodology for developing a chatbot system that leverages LLMs to address these limitations. Furthermore, the project investigates how to utilize user interaction data from the chatbot to generate informative documents, such as FAQs and troubleshooting guides. The ultimate goal is to create a system that enhances user experience by providing efficient information access and streamlines knowledge capture.

1.3 Problem Statement

This project seeks to address the following challenges:

- **Limited User Experience:** Existing support channels may not offer convenient and interactive ways for users to find answers and troubleshoot problems.
- **Inefficient Knowledge Transfer:** Traditional methods of knowledge capture may not effectively capture and disseminate the insights gained from user interactions.
- **Limitations of Traditional Chatbots:** Current chatbots can be hindered in their ability to understand complex user queries or adapt to diverse language styles.

1.4 Salient Contribution

This project proposes a novel methodology that combines LLM-powered chatbot development with document generation based on user interaction analysis. This approach offers several potential benefits:

- Improved user experience through interactive and natural language chat interactions.
- Enhanced knowledge capture by identifying user pain points and creating targeted documentation.
- Reduced burden on traditional support channels by deflecting user inquiries towards the chatbot.

1.5 Organization of Report

The remainder of this report will delve into the proposed methodology. Section 2 outlines the key stages involved, including preparation, creation (focusing on chatbot development), extraction (analyzing chat history data), document generation, and ongoing discussions with SREs (Site Reliability Engineers). Section 3 will be populated with results and analysis upon project completion. Section 4 will explore the advantages, limitations, and potential applications of this methodology. Finally, Section 5 will provide a conclusion and discuss future work directions.

Chapter 2

Literature survey

The literature survey discusses on Chatbot application using Large Language Models [1] and frameworks in the domain that help in creating end application from it.Chatbots are getting a boost from LangChain and Chainlit. These frameworks work together to simplify chatbot creation. Chainlit handles the user interface and conversation flow, while LangChain lets you tap into the power of large language models (LLMs) for things like answering questions and generating text.

One key advantage is prompt engineering. By crafting effective prompts, you can guide the LLM towards the kind of responses you want for your chatbot. This is essential for making your chatbot behave the way you want it to and for creating natural conversations. Overall, LangChain and Chainlit are a powerful combination for building chatbots using which one can take unstructured information from LLM and store it in structured manner [2].They're flexible and can be used for a variety of purposes, from customer service to HR to casual conversation.

Another aspect of this chatbot systems is to make the system robust enough to provide outputs that is safe and reliable and does not lead to generation of harmful topics that do not fit original scope of the project. For this there are several approaches like using guardrails and presidio library [3,4,5] that are specifically focused on task of assessing the input and output of applications that generate content using LLMs.

Chapter 3

Methodology and Implementation

4.1 Block diagram

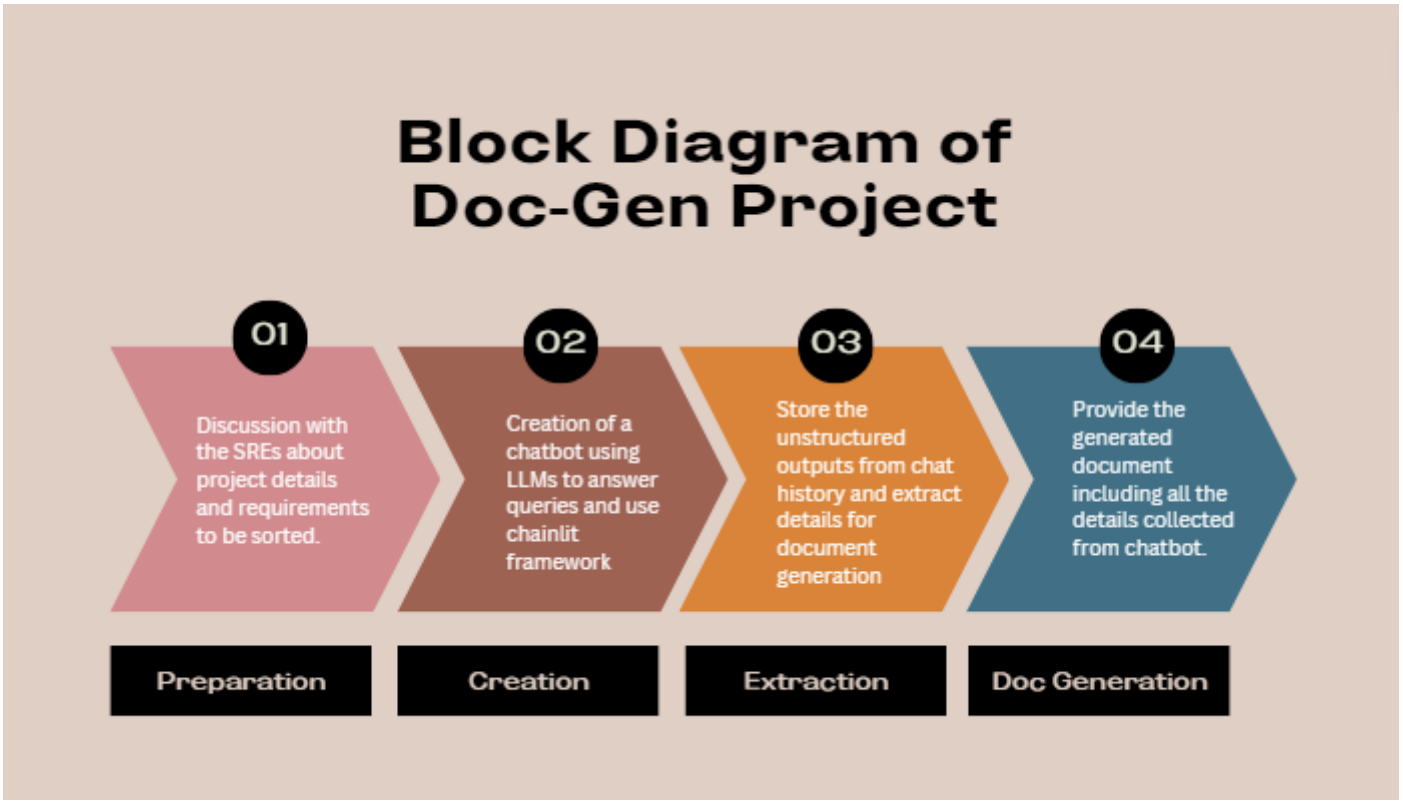

Figure1.Block Diagram

1. Preparation

- **Project Discussion:** We will convene with Site Reliability Engineers (SREs) to establish the project's objectives and requirements. This discussion will center around:
    - The chatbot's target functionality and intended audience.
    - The scope of queries the chatbot should handle.
    - Data sources that will inform the chatbot's responses.
    - Success metrics to gauge the chatbot's effectiveness.
- **Data Collection:** Following the defined requirements, we will gather the necessary data to train the chatbot and inform document generation. This may include:
    - Existing documentation relevant to the chatbot's domain, such as VMs, OS version and all different types of identifications.

2. Creation

- **LLM-based Chatbot Development:**
    - We will select an LLM model appropriate for the project's needs, such as question answering.
    - The Chainlit framework will be employed to construct the chatbot's user interface, facilitating a swift web-based deployment.
    - The LLM will be prompt engineered with necessary requirements for SREs.This entails feeding the data into the model and enabling it to learn patterns and associations between questions and corresponding answers.
    - Prompt engineering will be utilized to craft specific prompts that guide the LLM towards generating informative and relevant responses to user queries.

3. Extraction

- **Chat History Analysis:** Upon deploying the chatbot, we may extract crucial details from user interactions to inform document generation. Some potential techniques include:
    - Using langchain chat history to access data from the chatbot to store data by storing necessary requirements in JSON format.

4. Doc Generation

- **Content Creation:** Leveraging the extracted details and insights, the LLM will be utilized to generate a document that encapsulates the information gleaned from chat interactions. This document could encompass:
    - Frequently Asked Questions (FAQs) compiling common user queries and corresponding answers.
    - Troubleshooting guides outlining steps to resolve frequently encountered user issues.
    - Knowledge Base Articles consolidating information on specific topics relevant to the chatbot's domain.

5. SRE Discussion

Following each stage, we will return to the project objectives in conversation with the SREs. This iterative approach ensures the chatbot and generated documents remain aligned with the initial goals and requirements.

Additional Considerations

- **Chatbot Evaluation:** The chatbot's performance will be continuously monitored and evaluated. This may involve soliciting user feedback or assessing how effectively it addresses user queries.
- **Data Security:** We will ensure that all collected data, including chat history, adheres to data privacy regulations.

By adhering to these outlined stages and fostering open communication with the SREs, we can establish a robust chatbot and document generation system that enhances user experience and knowledge capture.

4.2 Software description, flowchart / algorithm

This chapter can comprise of actual implementation photos and their description.

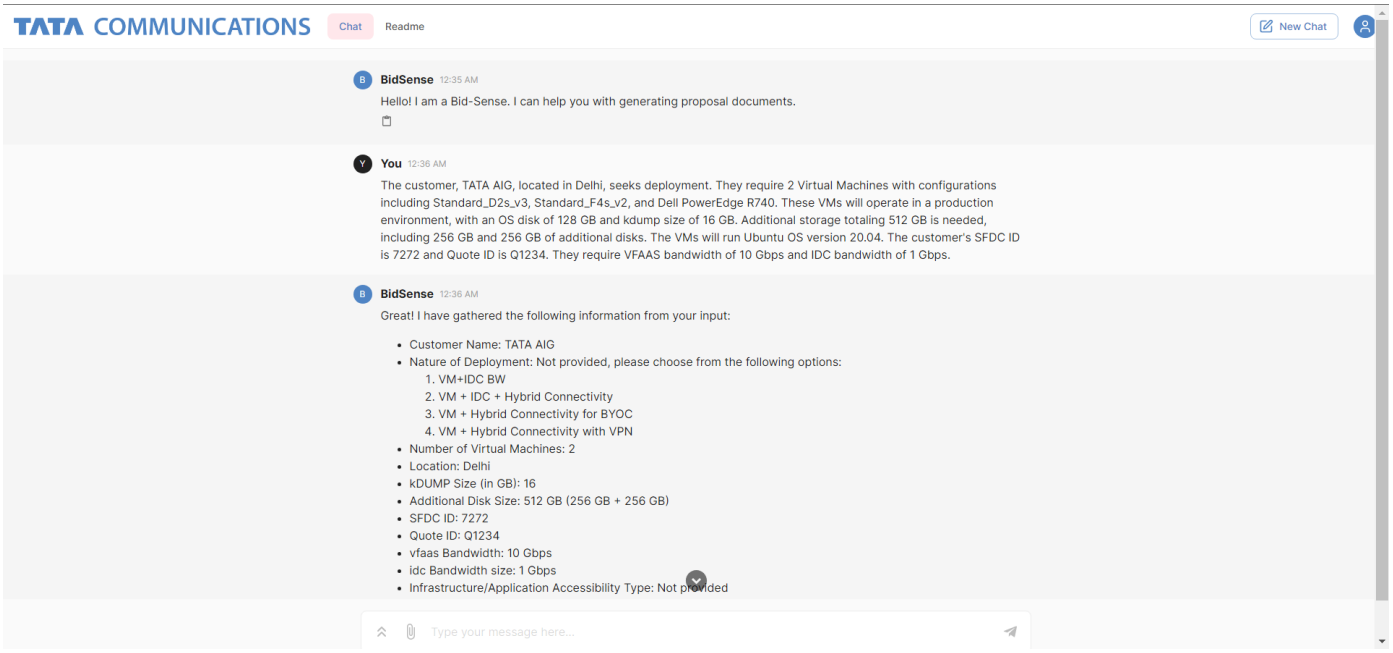

Figure2.Software Screenshot

Chapter 4

Results and Analysis

The outputs that are generated by the LLM with help of prompt is able to gather details from the end user and ask necessary questions if any input is missing or invalid . The chatflow follows the pattern that has been specifically guided by the prompt and the chainlit framework enables to upload file content as well so as to read the inputs from the end user and using this pass to document generation pipeline that helps in generating the document for the end user.

Chapter 5

Advantages, Limitations and Applications

1. Preparation

- **Project Discussion:** We will convene with Site Reliability Engineers (SREs) to establish the project's objectives and requirements. This discussion will center around:
    - The chatbot's target functionality and intended audience.
    - The scope of queries the chatbot should handle.
    - Data sources that will inform the chatbot's responses.
    - Success metrics to gauge the chatbot's effectiveness.
- **Data Collection:** Following the defined requirements, we will gather the necessary data to train the chatbot and inform document generation. This may include:
    - Existing documentation relevant to the chatbot's domain, such as manuals, FAQs, and troubleshooting guides.
    - If available, anonymized transcripts of past user chat interactions.

2. Creation

- **LLM-based Chatbot Development:**
    - We will select an LLM model appropriate for the project's needs, such as question answering.
    - The Chainlit framework will be employed to construct the chatbot's user interface, facilitating a swift web-based deployment.
    - The LLM will be trained on the assembled data (documents and potentially anonymized chat history). This entails feeding the data into the model and enabling it to learn patterns and associations between questions and corresponding answers.
    - Prompt engineering will be utilized to craft specific prompts that guide the LLM towards generating informative and relevant responses to user queries.

3. Extraction

- **Chat History Analysis:** Upon deploying the chatbot, we may extract crucial details from user interactions to inform document generation. Some potential techniques include:
    - Named Entity Recognition (NER): NLP techniques will be used to identify important entities within chat history, such as product names, error codes, or specific tasks.
    - Sentiment Analysis: We will gauge user sentiment (positive, negative, or neutral) from conversations to comprehend user satisfaction or areas of frustration.

4. Doc Generation

- **Content Creation:** Leveraging the extracted details and insights, the LLM will be utilized to generate a document that encapsulates the information gleaned from chat interactions. This document could encompass:
    - Frequently Asked Questions (FAQs) compiling common user queries and corresponding answers.
    - Troubleshooting guides outlining steps to resolve frequently encountered user issues.
    - Knowledge Base Articles consolidating information on specific topics relevant to the chatbot's domain.

5. SRE Discussion

Following each stage, we will return to the project objectives in conversation with the SREs. This iterative approach ensures the chatbot and generated documents remain aligned with the initial goals and requirements.

Additional Considerations

- **Chatbot Evaluation:** The chatbot's performance will be continuously monitored and evaluated. This may involve soliciting user feedback or assessing how effectively it addresses user queries.
- **Data Security:** We will ensure that all collected data, including chat history, adheres to data privacy regulations.

By adhering to these outlined stages and fostering open communication with the SREs, we can establish a robust chatbot and document generation system that enhances user experience and knowledge capture.

Advantages:

- **Improved User Experience:** The chatbot provides a convenient and interactive way for users to find answers and troubleshoot problems, potentially reducing frustration and improving overall satisfaction.
- **Reduced Support Burden:** By deflecting user inquiries, the chatbot frees up SREs and support staff to focus on more complex issues.
- **Knowledge Capture:** Analyzing chat history allows for the identification of common user issues and the creation of documents (FAQs, guides) that address these needs, effectively capturing and disseminating knowledge.
- **Scalability:** The LLM-based approach can handle a large volume of user interactions efficiently.
- **Cost-Effectiveness:** Chatbots can potentially reduce the need for human support staff, leading to cost savings.
- **Personalization:** LLMs can potentially personalize chatbot responses based on user history or context, leading to a more engaging experience.

Limitations:

- **Data Dependence:** The effectiveness of the system relies heavily on the quality and relevance of the training data. Poor quality data can lead to inaccurate or unhelpful responses.
- **Limited Understanding:** While LLMs are becoming more sophisticated, they may still struggle with complex questions, sarcasm, or nuanced language and providing consistent output
- **Bias:** Biases present in the training data can be reflected in the chatbot's responses. Careful data selection and mitigation strategies are crucial.
- **Security Concerns:** Data privacy and security must be considered when collecting and storing user interactions.
- **Development and Maintenance:** Building and maintaining the system requires technical expertise and ongoing effort.

Applications:

This methodology can be applied in various domains, including:

- **Customer Service:** Chatbots can answer frequently asked questions, troubleshoot common problems, and direct users to relevant resources.
- **Technical Support:** Chatbots can provide initial troubleshooting steps and escalate complex issues to human support.
- **Education:** Chatbots can answer student questions, offer personalized learning paths, and provide supplemental instruction.
- **Human Resources:** Chatbots can answer employee questions about benefits, policies, and procedures.
- **Healthcare:** Chatbots can offer basic health information, schedule appointments, and answer frequently asked medical questions (under appropriate guidance from medical professionals).

Chapter 6

Conclusion and Future Scope

The document generation chatbot for Site Reliability Engineers (SREs) is a cutting-edge technology that leverages the power of Large Language Models (LLMs) to assist SREs in generating high-quality documents. Currently, the chatbot uses the Langchain framework to gather requirements from the user, access chat history, and utilize templates to create documents, providing an interactive window for the end-user to interact with through the Chainlit framework. In the future, the chatbot will be enhanced to dynamically generate cloud architecture diagrams based on user input, perform more complex tasks within the document, and integrate with other systems and tools. This will enable SREs to visualize their infrastructure, make data-driven decisions, and extract information from unstructured data sources. The chatbot will also provide real-time updates and notifications, improving collaboration and decision-making. By automating the document generation process, the chatbot will increase efficiency, improve accuracy, and enhance collaboration, making it an invaluable tools for SREs. Including robust guardrails and PII detection methods would also be a key step that will have to be taken into implementation for safer and reliable application.

References

1) Topsakal, Oguzhan, and Tahir Cetin Akinci. "Creating large language model applications utilizing langchain: A primer on developing llm apps fast." *International Conference on Applied Engineering and Natural Sciences*. Vol. 1. No. 1. 2023.
2) Xieyang Liu, Michael, et al. "" We Need Structured Output": Towards User-centered Constraints on Large Language Model Output." *arXiv e-prints* (2024): arXiv-2404.
3) Rebedea, Traian, et al. "Nemo guardrails: A toolkit for controllable and safe llm applications with programmable rails." *arXiv preprint arXiv:2310.10501* (2023).
4) Liu, Yi, et al. "Prompt Injection attack against LLM-integrated Applications." *arXiv preprint arXiv:2306.05499* (2023).
5) Friebely, Alexander. *Analyzing the Efficacy of Microsoft Presidio in Identifying Social Security Numbers in Unstructured Text*. Diss. Utica University, 2022.