THINKAI:CHAT WITH YOUR NOTES AND IDEAS

A Project Report

Submitted to the APJ Abdul Kalam Technological University

in partial fulfillment of requirements for the award of degree

Bachelor of Technology

in

Computer Science and Engineering

by

NANDANA K V(KSD20CS077)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LBS COLLEGE OF ENGINEERING KASARAGOD

KERALA

December 2023

LBS COLLEGE OF ENGINEERING, KASARAGOD

MULIYAR – 671 542

DEPT. OF COMPUTER SCIENCE & ENGINEERING

Vision of Department

To be a renowned centre for education, research, and innovation in the frontier areas of Computer Science and Engineering.

Mission of Department

- Establish and maintain an operational environment to acquire, impart, create and apply knowledge in Computer Science and Engineering and inter-disciplinary areas
- Serve as a resource centre for innovation in design and development of software and hardware solutions
- Inculcate leadership qualities, professional ethics and a sense of social commitment

DEPT. OF COMPUTER SCIENCE & ENGINEERING

LBS COLLEGE OF ENGINEERING

KASARAGOD

2023 - 24

CERTIFICATE

This is to certify that the report entitled **THINKAI:CHAT WITH YOUR NOTES AND IDEAS** submitted by **NANDANA K V** (KSD20CS077) to the APJ Abdul Kalam Technological University in partial fulfillment of the B.Tech. degree in Computer Science and Engineering is a bonafide record of the project work carried out by them under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Dr.Anver S R
(Project Guide)
Associate Professor
Dept.of CSE
LBS College of Engineering
Kasaragod

Prof.Vinod George
(Project Coordinator)
Associate Professor
Dept.of CSE
LBS College of Engineering
kasaragod

Dr.Anver S R
Associate Professor and HOD
Dept.of CSE
LBS College of Engineering
Kasaragod

DECLARATION

We hereby declare that the project report **THINKAI:CHAT WITH YOUR NOTES AND IDEAS**, submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by us under the supervision of Dr.Anver S R .

This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources.

We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission.We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

NANDANA K V

Kasaragod
13-05-2024

ABSTRACT

This project encompasses the development of multiple AI-powered chatbot applications tailored for various tasks, including PDF document processing, website interaction, YouTube video summarization, and image-based question answering. The AI-Powered Chatbot for PDF Document Processing aims to revolutionize how users interact with complex PDF files, enabling efficient access, analysis, and extraction of valuable insights using advanced artificial intelligence techniques. In Chatting with Websites Using AI, the chatbot is designed to engage with websites intelligently, providing users with personalized assistance, information retrieval, and seamless interaction, enhancing the user experience. The YouTube Video Summarizer Using Chatbot leverages AI to summarize video content, enabling users to extract key information, insights, and highlights from videos efficiently and accurately. Additionally, the Image-based Question Answering feature allows users to ask questions based on images, with the chatbot providing relevant answers using state-of-the-art AI models. Together, these projects showcase the versatility and power of AI-powered chatbots in simplifying complex tasks, improving information access, and enhancing user engagement across different domains.

ACKNOWLEDGEMENT

We take this opportunity to express our deepest sense of gratitude and sincere thanks to everyone who helped us to complete this work successfully. We express our sincere thanks to Dr.Anver S R, Head of Department,Computer Science and Engineering, LBS College of Engineering Kasaragod for providing us with all the necessary facilities and support.

We would like to express our sincere gratitude to the Prof.Vinod George, department of Computer Science and Engineering, LBS College of Engineering Kasaragod for the support and co-operation.

We would like to place on record our sincere gratitude to our project guide Dr.Anver S R , Associate Professor, Computer Science and Engineering, LBS College of Engineering for the guidance and mentorship throughout this work.

Finally we thank our family, and friends who contributed to the succesful fulfilment of this project work.

NANDANA K V

Contents

Abstract **i**

Acknowledgement **ii**

List of Figures **vii**

1 INTRODUCTION **1**

2 PROBLEM STATEMENT **2**

3 MOTIVATION **3**

4 LITERATURE REVIEW **4**

4.1 Least-To-Most Prompting Enables Complex Reasoning in Large Language Models 4

4.2 Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions 5

4.3 Extractive Text Summarization using Word Vector Embedding 7

4.4 KBot: A Knowledge Graph Based ChatBot for Natural Language Understanding Over Linked Data 9

4.5 DocChat: An Information Retrieval Approach for Chatbot Engines Using Unstructured Documents 10

4.6 AI Assistant For Document Management Using Langchain and Pinecone 12

4.7 Design and Development of Retrieval-Based Chatbot Using Sentence Similarity 13

4.8 Creating Large Language Model Applications Utilizing LangChain: A Primer on Developing LLM Apps Fast 14

4.9 An Information Retrieval-based Approach for Building Intuitive Chatbots for Large Knowledge Bases 15
4.10 Evaluating Large Language Models for Document-grounded Response Generation in Information-Seeking Dialogues 17
4.11 A Comparison of Semantic Similarity Methods for Maximum Human Interpretability 18

5 PROJECT MANAGEMENT 20
5.1 SCRUM 20
5.1.1 User Stories 20
5.1.2 Sprint 1 21
5.1.3 Sprint 2 22
5.1.4 Sprint 3 23
5.1.5 Sprint 4 24

6 SYSTEM DESIGN 25
6.1 Input Text Preprocessing: 25
6.2 Embedding Generation: 25
6.3 Indexing and Search: 25
6.4 User Query Preprocessing: 26
6.5 Context Retrieval: 26
6.6 Overall Architecture 26
6.7 Detailed Architecture 27

7 METHODOLOGY 28
7.1 Developing an AI-Powered Chatbot for PDF Document Processing . . 28
7.1.1 User Interface Development: 28
7.1.2 Document Loading and Processing: 28
7.1.3 Text Embeddings and Vectorization: 29
7.1.4 Chatbot Functionality: 29
7.1.5 Sub-Question Generation: 29
7.1.6 Answer Generation using RAG: 29
7.1.7 User Interaction and Feedback: 29

7.1.8 Iterative Development and Testing: 29
7.1.9 Deployment and Integration: 30
7.1.10 Documentation and Reporting: 30
7.2 Retrieval Augmented Generation(RAG) 30
7.2.1 Input Query: 31
7.2.2 Indexing and Retrieval: 31
7.2.3 Retrieval powered via Similarity search: 32
7.2.4 Vector stores: 33
7.2.5 Query Translation: 34
7.2.6 Decomposition: 34
7.2.7 Generation: 35
7.3 Chatting with Websites Using AI 35
7.3.1 Technology Selection: 35
7.3.2 Application Architecture: 35
7.3.3 Implementation Steps: 36
7.3.4 Testing and Optimization: 36
7.3.5 Deployment and User Experience: 36
7.4 YouTube Video Summarizer Using Chatbot 37
7.4.1 Application Architecture: 37
7.4.2 Implementation Steps: 38
7.4.3 Testing and Optimization: 38
7.4.4 Deployment and User Experience: 38
7.5 Image-based Question Answering 39
7.5.1 Setting Up Google API Key: 39
7.5.2 Application Components: 39
7.5.3 Main Functionality: 39
7.5.4 User Interface: 39
7.5.5 Workflow Execution: 40
7.5.6 Error Handling: 40

8 RESULTS 41

8.1 Developing an AI-Powered Chatbot for PDF Document Processing . . 41

8.2 Chatting with Websites Using AI 42

8.3 YouTube Video Summarizer Using Chatbot 42

8.4 Image-based Question Answering 43

9 CONCLUSION 45

10 FUTURE WORKS 46

References 47

List of Figures

5.1 User Stories 21

5.2 Sprint 1 and Graph 21

5.3 Sprint 2 and Graph 22

5.4 Sprint 3 and Graph 23

5.5 Sprint 4 and Graph 24

6.1 Overall Architecture 26

6.2 Detailed Architecture 27

7.1 RAG motivation 30

7.2 Indexing and Retrieval 31

7.3 Retrieval powered via similarity search 32

7.4 Vectorstores 33

7.5 Decomposition 34

8.1 Home page 41

8.2 Chat with PDF 42

8.3 Chat with Websites 42

8.4 Youtube Video Summarizer 43

8.5 Chat with Image 43

Chapter 1

INTRODUCTION

We are embarking on a transformative journey to develop an innovative AI-driven project that encompasses the realms of PDF document processing, website interaction through AI, YouTube video summarization with a chatbot interface, and image-based question answering. Our endeavor is rooted in the profound potential of artificial intelligence to revolutionize how we engage with and extract insights from complex information sources. At the core of our initiative is the development of an AI-powered chatbot meticulously crafted to streamline the processing of PDF documents, simplifying the retrieval and analysis of crucial information vital for students, researchers, and professionals alike. This chatbot not only navigates through the intricacies of multiple PDF documents but also delves into the digital landscape by engaging in meaningful conversations with websites, leveraging AI algorithms to enhance information retrieval and user experience. In parallel, our project extends its reach to the dynamic realm of online video content, where our chatbot serves as a robust YouTube video summarizer, condensing lengthy videos into concise and informative summaries, empowering users with efficient content consumption. Furthermore, our chatbot's capabilities transcend text-based interactions, embracing image-based question answering, where it intelligently interprets and responds to inquiries based on visual cues, opening new avenues for intuitive information retrieval and user engagement. Through this comprehensive approach, we aim to redefine the boundaries of AI-driven tools, empowering users with efficient, intelligent, and user-friendly solutions that cater to diverse information needs in the digital era.

Chapter 2

PROBLEM STATEMENT

Develop an AI chatbot leverages the power of RAG and LangChain that efficiently processes PDF documents, interacts with websites, summarizes YouTube videos, and answers questions based on image inputs, offering enhanced functionality and user experience.

Chapter 3

MOTIVATION

Our project is driven by a deep-seated motivation to revolutionize the way people interact with information across multiple domains. By developing an AI-powered chatbot, we aim to streamline the often cumbersome process of processing PDF documents, engaging in meaningful conversations with websites using AI capabilities, summarizing YouTube videos using a chatbot, and answering questions based on image inputs. These tasks, while crucial in various fields such as research, education, and business, are often time-consuming and complex for users to handle manually. Our motivation stems from addressing these challenges head-on, empowering users with a sophisticated tool that leverages cutting-edge AI technologies. Through our chatbot, users can effortlessly extract insights from PDF documents, engage in intelligent conversations with websites to gather information efficiently, summarize lengthy YouTube videos into concise formats, and receive accurate answers to questions based on visual content. Ultimately, our project seeks to enhance productivity, streamline information retrieval processes, and empower users with a seamless and intuitive AI-driven experience across diverse tasks and platforms.

Chapter 4

LITERATURE REVIEW

4.1 Least-To-Most Prompting Enables Complex Reasoning in Large Language Models

The paper addresses the challenge of easy-to-hard generalization in natural language reasoning tasks, particularly focusing on the limitations of chain-of-thought prompting. While chain-of-thought prompting has shown impressive performance in various reasoning tasks, it tends to struggle when faced with problems that require solutions beyond the scope of the exemplars provided in the prompts. This limitation highlights a significant gap between human intelligence and machine learning capabilities, such as the ability of humans to learn from a few examples, explain their decisions, and solve novel and complex problems. The authors propose a novel prompting strategy called least-to-most prompting to overcome this challenge.

Least-to-most prompting is designed to break down complex problems into a series of simpler subproblems, which are then solved sequentially. The key idea is that solving each subproblem is aided by the answers obtained from previously solved subproblems. This strategy leverages the concept of progressive learning, where learners start with simpler concepts before moving on to more complex ones. The paper presents experimental results across tasks related to symbolic manipulation, compositional generalization, and math reasoning, demonstrating that least-to-most prompting enables generalization to more difficult problems than those seen in the prompts.

The study compares the effectiveness of least-to-most prompting with chain-of-thought prompting using the GPT-3 code-davinci-002 model. Remarkably, least-to-most prompting achieves significantly higher accuracy on tasks like compositional generalization, surpassing the performance of neural-symbolic models trained on extensive datasets. This achievement is particularly noteworthy as it showcases the potential of few-shot prompting strategies to handle complex reasoning tasks with minimal training data.

Furthermore, the paper discusses the significance of bridging the gap between human intelligence and machine learning capabilities. It acknowledges the success of deep learning but emphasizes the fundamental differences, such as humans' ability to learn from few examples, provide explanations for decisions, and tackle novel and challenging problems. The integration of chain-of-thought prompting with few-shot learning and self-consistency decoding represents a significant advancement in natural language processing, improving interpretability and performance on challenging tasks.

Overall, the paper contributes to the ongoing efforts in narrowing the disparity between human and machine intelligence by proposing an innovative prompting strategy that enhances the generalization ability of language models, particularly in solving complex reasoning problems beyond the scope of training data.

4.2 Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions

The paper introduces a novel approach called Interleaving Retrieval with Chain-of-Thought Reasoning (IRCoT) for addressing the challenges faced by large language models (LLMs) in multi-step question answering tasks, particularly in scenarios where required knowledge is not readily available within the model's parameters. The authors highlight the limitations of existing strategies, such as one-step retrieval based solely on the question, especially for complex multi-step reasoning questions that necessitate iterative retrieval and reasoning processes. IRCoT aims to augment chain-of-thought prompting for open-domain, knowledge-intensive tasks that demand intricate reasoning steps.

The motivation behind IRCoT stems from the observation that existing retrieval methods, while beneficial, often fall short in guiding the reasoning process effectively, leading to model hallucination and factual inaccuracies. The paper poses the question of how to enhance chain-of-thought prompting for tasks requiring complex multi-step reasoning, where retrieval and reasoning steps need to inform each other dynamically. For instance, retrieving partial knowledge based on the question's query may not suffice for questions involving iterative reasoning and multiple sources of information.

To address these challenges, IRCoT introduces an interleaving approach that intertwines retrieval with steps in a chain of thoughts (CoT). This interleaving allows for more relevant information retrieval guided by CoT reasoning, and vice versa, thereby improving the accuracy and relevance of both retrieval and reasoning steps. The authors illustrate this concept using examples that highlight the iterative nature of multi-step question answering, where each step of reasoning may depend on previously retrieved information and CoT sentences.

The proposed IRCoT framework involves alternating between extending CoT sentences and expanding retrieved information iteratively. This iterative process leverages the CoT reasoning to guide retrieval, ensuring that retrieved facts align with the ongoing reasoning steps, thus reducing model hallucination and improving factual accuracy. The paper evaluates IRCoT on multiple datasets, demonstrating significant improvements in retrieval effectiveness, downstream QA performance, reduction in factual errors, and compatibility with both large-scale and smaller-scale models without additional training.

In summary, IRCoT represents a significant advancement in the field of natural language processing, particularly in multi-step question answering tasks. By integrating retrieval and reasoning in an interleaved manner, IRCoT offers a promising solution to the challenges of knowledge-intensive, complex reasoning scenarios, showcasing improvements in both retrieval accuracy and downstream QA performance across various datasets and model scales.

4.3 Extractive Text Summarization using Word Vector Embedding

Extractive Text Summarization Using Word Vector Embedding, A. Jain et al. [1], 2017. The provided study focuses on text summarization, a vibrant area of research aimed at distilling relevant information from large documents across various domains such as finance, news media, academics, and politics.

The authors propose an approach for supervised extractive summarization, leveraging a combination of feature extraction and neural network techniques.

The study evaluates the effectiveness of their method using the Document Understanding Conferences 2002 dataset and compares it against various online extractive text summarizers. Text summarization, particularly based on either abstractive or extractive methods, is identified as a popular approach. The paper leans towards extractive summarization, which involves gathering relevant sentences from documents. The distinction between abstractive and extractive summarization methods is outlined, emphasizing the simplicity of the latter. The methodology section details the approach for text summarization as a binary classification problem. The explored text is categorized as either relevant for inclusion in the summary or irrelevant. The document is broken down into sentences, and features are extracted. These features are then used to train a neural network for predicting the inclusion of sentences in the summary.

The proposed method incorporates both standard features and word vector embedding-based features to enhance summarization accuracy. The paper identifies four major challenges in extractive text summarization, including the identification of important information, removal of irrelevant details, minimizing unnecessary information, and assembling relevant information into a coherent report.

The study evaluates its proposed method against challenges by employing a good set of features followed by a neural network for supervised extractive summarization. The inclusion of word vector embedding-based features contributes to higher accuracy, as demonstrated through testing against various online extractive text summarizers using the DUC 2002 dataset.

The paper concludes by summarizing the proposed methodology's effectiveness and suggests future research directions. The combination of feature extraction and neural networks for supervised extractive summarization proves promising, highlighting the potential for further advancements in the field.

4.4 KBot: A Knowledge Graph Based ChatBot for Natural Language Understanding Over Linked Data

KBot: A Knowledge Graph Based ChatBot for Natural Language Understanding Over Linked Data, A. Ait-Mlouk et al. [2], 2020. The paper addresses the growing significance of chatbots in leveraging linked data, specifically knowledge bases (KBs), to make structured data accessible and useful for end-users.

The authors highlight the challenges associated with building chatbots over linked data, emphasizing the importance of user query understanding, support for multiple knowledge bases, and handling multilingual aspects. The review of related works in the field is crucial to understanding the context and evolution of chatbots over linked data. T

The paper traces the historical evolution of chatbots from their inception in the 1960s with systems like Eliza, Parry, and Alice, which were primarily based on text conversation. It notes the significant progress made over the decades, leading to the development of sophisticated AI chatbots such as Siri, Cortana, Google Assistant, and others by major companies.

The overview provides context for the reader by highlighting the trajectory of chatbot development and its integration into various platforms and applications. In the context of linked data, the authors emphasize the primary goal of chatbot systems—to retrieve relevant information from one or multiple knowledge bases using natural language understanding (NLU) and semantic web technologies. They underscore the transformation of natural language into SPARQL queries as a key mechanism for achieving this goal.

The literature review acknowledges the progress in chatbot research within the linked data domain but identifies persistent challenges, including user query understanding, intent classification, multilingual support, handling multiple knowledge bases, and understanding analytical queries. The authors discuss the challenges faced by existing linked data chatbots, emphasizing the need for substantial training data, which is often expensive and challenging to obtain. They recognize the recent growth in linked data development and its impact on chatbot advancements in both research and industry. Despite this progress, the paper asserts that challenges such as user query understanding, intent classification, multilingual aspects, support for multiple knowledge bases, and analytical query comprehension persist in the field.

The literature review introduces the proposed solution, KBot, as a chatbot that addresses several challenges in the linked data domain. It emphasizes KBot's ability to compete with existing linked data chatbots in terms of performance. The authors outline key contributions, including the design and implementation of KBot, a machine learning model (SVM) for intent classification, an analytical queries engine for data exploration, and scalability features that allow the addition of new knowledge bases, support for multiple languages, and flexibility for diverse tasks.

In summary, the literature review provides a comprehensive overview of the historical evolution of chatbots, their integration with linked data, and the persisting challenges in the field. It sets the stage for the proposed solution, KBot, by establishing the context and underscoring the need for advancements in linked data chatbots.

4.5 DocChat: An Information Retrieval Approach for Chatbot Engines Using Unstructured Documents

The paper addresses a critical challenge in the development of chatbot engines, specifically focusing on the limitation of existing in engines that rely on predefined utterance-response (Q-R) pairs.

The study introduces "DocChat," a novel information retrieval approach that leverages unstructured docu- ments instead of Q-R pairs to respond to user utterances. To understand the context and significance of this novel approach, a literature review covers key themes related to chatbot development, existing methods, and challenges in the field. Building chatbot engines capable of natural language interaction with humans represents a formidable challenge in artificial intelligence. The paper acknowledges the complexity of this problem and emphasizes the rapid growth of social media platforms, community question answering (CQA) websites, and the vast amount of Q-R pairs that have become available.

The explosion of data-driven chatbot approaches is discussed as a response to this growing corpus of Q-R pairs. The paper categorizes existing methods for short text conversation (STC) into two main types: retrieval-based methods and generation-based methods. Retrieval-based methods involve matching the current utterance with existing Q-R pairs, and generation-based methods use an encoder- decoder framework to generate responses.

However, both approaches have drawbacks, such as intractability in collecting Q-R pairs for specific domains and limitations in the fluency and naturality of machine-generated text. To address the limitations of existing methods, the paper introduces ”DocChat” as a response retrieval approach based on unstructured documents. Unlike traditional Q-R pair-based methods, DocChat selects a response sentence directly from given documents by ranking all possible sentences using features designed at different levels of granularity.

This innovative approach aims to improve the adaptability of chatbot engines to various topics and ensures the fluency and naturality of responses since they are drawn from existing documents. The literature review highlights the promising results obtained through experiments, emphasizing the effectiveness of DocChat in both question-answering (QA) and chatbot scenarios. The approach’s adaptability and the natural fluency of responses are identified as key advantages. Additionally, the paper emphasizes the contributions of DocChat, positioning it as a solution that complements chatbot engines using Q-R pairs as their primary source of responses.

In summary, the literature review provides a comprehensive background on the challenges associated with chatbot development, the limitations of existing methods, and the introduction of DocChat as an innovative response retrieval approach. It sets the stage for the paper’s contributions and showcases the need for advancements in chatbot technology.

4.6 AI Assistant For Document Management Using Langchain and Pinecone

The paper presents an innovative project that leverages LangChain and the Large Language Model (LLM) to develop a chatbot specifically tailored for PDF document interactions. LangChain, a framework highlighted in the paper, streamlines the creation of chatbots and scalable AI/LLM applications, showcasing its potential to simplify AI development. The LLM Model, a massive language model, serves as a cornerstone for generating text, facilitating language translation, producing creative content, and offering insightful responses to user inquiries. This combination of LangChain and the LLM Model forms the foundation of the chatbot's functionality, empowering it to handle queries related to PDF files effectively.

The chatbot's training on a dataset of PDF files is a notable aspect of the project, enabling it to comprehend and respond intelligently to user queries regarding PDF content. By integrating Google Search capabilities, the chatbot enhances its information retrieval process, delivering comprehensive and informative answers to users. The utilization of Pinecone for storing PDF file vectors and embeddings further strengthens the project's infrastructure, ensuring efficient retrieval of related documents for seamless user interactions.

In terms of implementation, React JS plays a crucial role in developing the front-end interface for users to interact with the chatbot. This web-based interface offers a user-friendly platform for users to ask questions, receive responses, and engage with PDF content effortlessly. Through rigorous testing on a diverse range of PDF files, the chatbot demonstrates high accuracy and reliability, underscoring its effectiveness in handling PDF-related inquiries.

The integration of LangChain, LLM Model, Pinecone for vector storage, and React JS for front-end development reflects a comprehensive and well-rounded approach to chatbot development for PDF document processing. The paper's focus on practical implementation, technology integration, and testing outcomes provides valuable insights into the capabilities and potential applications of AI-driven chatbots in handling complex document formats like PDFs. Overall, the paper contributes to the growing body of research on AI-powered chatbots and their utility in enhancing user experiences with document-based interactions.

4.7 Design and Development of Retrieval-Based Chatbot Using Sentence Similarity

The paper presents a comprehensive exploration into the development of a Chatbot tailored for Prasad V Potluri Siddhartha Institute of Technology, designed to address diverse inquiries related to the college's facilities, procedures, policies, and more. This initiative aligns with the growing trend of integrating Chatbots across various sectors, driven by the increasing adoption of automation in business processes. The Chatbot's core functionality revolves around engaging users through text inputs and delivering responses using machine learning concepts, showcasing its versatility and adaptability in handling user queries effectively.

A crucial aspect of this project is the utilization of a retrieval approach, complemented by logic adapters, to process user inputs and generate contextually appropriate responses. This approach enhances the Chatbot's ability to understand and respond intelligently to a wide range of inquiries, contributing to a more seamless user experience. The incorporation of performance metrics, including performance, humanity, effect, and accessibility, underscores the project's focus on evaluating and optimizing the Chatbot's functionality and user interaction dynamics.

The paper also delves into the implementation details, highlighting the use of the Flask framework for developing the web-based Chatbot application. The front-end interface, developed using HTML, CSS, and internal JavaScript, provides a user-friendly platform for interacting with the Chatbot. Leveraging the Chatterbot Library, which is language-independent and facilitates conversational interactions, adds another layer of sophistication to the Chatbot's design and functionality.

A notable aspect of the paper is its comparative analysis between the existing web application and the enhanced version with the integrated Chatbot. This comparative study reveals a 20 percentage improvement in performance and a 5 percentage increase in accessibility, underscoring the positive impact of integrating the Chatbot into the college's digital ecosystem.

The paper's structured approach, from conceptualization to implementation and evaluation, provides valuable insights into the potential of Chatbots in educational institutions and the broader context of AI-driven conversational agents. It contributes to the ongoing discourse on leveraging machine learning and AI technologies to enhance user experiences, automate routine tasks, and streamline information dissemination processes within educational settings. Overall, the paper offers a comprehensive literature review and practical demonstration of deploying Chatbots for educational institutions, paving the way for future advancements and applications in this domain.

4.8 Creating Large Language Model Applications Utilizing LangChain: A Primer on Developing LLM Apps Fast

The rapid evolution of artificial intelligence (AI) over the past decade has ushered in a new era marked by transformative advancements in various domains. One of the pivotal developments within AI is the widespread adoption of Large Language Models (LLMs) due to their versatility in performing diverse tasks such as essay composition, code writing, explanation, and debugging. OpenAI's ChatGPT has particularly popularized the use of LLMs among millions of users, showcasing their immense potential across different applications. This study delves into the utilization of LLMs, with a specific focus on LangChain, an open-source software library designed to streamline the development of AI applications leveraging LLMs.

The introduction sets the stage by highlighting key milestones in AI evolution, including breakthroughs in deep learning, image and speech recognition, and reinforcement learning exemplified by DeepMind's AlphaGo and AlphaZero. The rise of generative models like BERT, GPT, and T5 has significantly enhanced natural language processing (NLP) capabilities, revolutionizing tasks such as machine translation, sentiment analysis, and text generation. These LLMs, characterized by their large parameter sizes and training on extensive datasets, have demonstrated human-like proficiency in generating contextually relevant and grammatically coherent text outputs.

Despite occasional limitations such as erroneous outputs, LLMs have rapidly gained traction for their prowess in various applications, especially with the introduction of OpenAI's ChatGPT series. This popularity surge has prompted a shift towards leveraging LLMs in diverse fields including education, research, customer service, content creation, healthcare, and entertainment. However, developing custom AI applications necessitates more than mere interaction through web interfaces, leading to the emergence of LangChain as a solution provider for rapid AI app development using LLMs.

The literature review elaborates on LangChain's capabilities, serving as a facilitator for developers to harness the power of LLMs effectively. By providing modular abstractions and customizable pipelines, LangChain empowers developers to create bespoke AI applications tailored to specific use cases. This paper serves as a primer for understanding LangChain's functionalities and its role in expediting the development of large language model applications, thereby contributing to the ongoing discourse on AI technology advancements and applications.

4.9 An Information Retrieval-based Approach for Building Intuitive Chatbots for Large Knowledge Bases

The paper introduces the significance of quickly finding relevant information in various application scenarios, highlighting the challenge of navigating vast data collections to address specific and often intricate problems. With advancements in automatic language processing, chatbots have emerged as a solution to streamline information search processes, offering a natural dialog-based interface for users to access the information they need efficiently.

The paper focuses on presenting a chatbot framework designed to answer questions related to services provided by public administration entities. This framework is tailored to support complex dialogs, provide hints, and offer recommendations, enhancing the user experience during interactions. The deployment of public chatbot services in two major German cities serves as a practical application of the framework, addressing inquiries regarding services, locations, and appointments. The architecture of the system is discussed, along with insights into the developed algorithms, providing a comprehensive view of the technical aspects involved.

The study sets the context by acknowledging the rising popularity of personal assistants or virtual agents across various domains. These digital assistants act as knowledgeable resources, assisting users with information and task resolution through interactive conversations. The shift towards chatbots as interactive search interfaces aims to enable longer dialogs, guide users through complex problem-solving, and accommodate natural language queries, enhancing the overall user experience compared to traditional search systems.

The development of chatbots brings about challenges such as handling diverse problem descriptions, domain-specific vocabulary, and informal language typical of user interactions. The need for extensive training data from specific domains poses a significant hurdle, particularly for broad scenarios encompassing numerous information objects. Information retrieval approaches offer a promising solution by extracting queries from natural language inputs, bridging the gap between user communication styles and formal knowledge bases.

Addressing colloquial language nuances and ensuring accuracy in natural language processing tools remain ongoing challenges, especially in formal domains like public administration. The paper outlines an approach combining information retrieval techniques and machine learning methods tailored for public administration services, emphasizing practical deployment and user feedback collection to evaluate the efficacy of the framework.

The subsequent sections of the paper delve into existing approaches, the proposed methodology, evaluation results, and future directions, providing a comprehensive analysis of the chatbot framework's development, deployment, and performance in real-world scenarios.

4.10 Evaluating Large Language Models for Document-grounded Response Generation in Information-Seeking Dialogues

The paper investigates the application of large language models (LLMs) like ChatGPT for generating document-grounded responses in information-seeking dialogues. It leverages the MultiDoc2Dial corpus, which comprises task-oriented dialogues across four social service domains, as a basis for evaluation. These dialogues are grounded in multiple documents that provide relevant information, presenting a challenging yet realistic scenario for response generation. Two methods are employed for generating dialogue completion responses using ChatGPT: Chat-Completion and LlamaIndex. The Chat-Completion method utilizes knowledge from ChatGPT's pre-training, while LlamaIndex extracts information from documents to enhance the response generation process.

Recognizing the limitations of automatic evaluation metrics in assessing the quality of document-grounded response generation by LLMs, the study conducts a human evaluation. This evaluation involves annotators rating the output of the shared task winning system, the two ChatGPT variants' outputs, and human responses. The evaluation sheds light on the performance of these systems in terms of relevance, trustworthiness, coherence, and fluency.

The paper highlights the importance of accessing domain-specific knowledge in task-oriented dialogue modeling to provide users with relevant and detailed information. It emphasizes the challenges associated with retrieving knowledge from diverse sources stored in multiple documents and formats, underscoring the need for dialogue systems to generate informed and coherent responses based on the retrieved knowledge and user queries.

The study compares traditional retrieval-augmented generation models with state-of-the-art LLMs like ChatGPT for knowledge-grounded response generation. It focuses on two prompting methods: Chat-Completion, which provides context to ChatGPT about the conversation topics and user queries, and LlamaIndex, which combines information extraction from documents with ChatGPT to generate grounded responses. The comparison aims to assess the suitability of these methods for the task of generating document-grounded responses in task-oriented dialogues.

The paper is structured to provide a comprehensive analysis, starting with an overview of the MultiDoc2Dial corpus and the DialDoc Shared Task, which forms the basis of the evaluation. It then delves into the description of the response generation methods, including the two novel methods leveraging ChatGPT. The experimental design is outlined, followed by the presentation of results from objective evaluations and human assessments. The paper concludes by summarizing the findings and discussing their implications.

4.11 A Comparison of Semantic Similarity Methods for Maximum Human Interpretability

The paper delves into the realm of text similarity computation, focusing on methods that go beyond mere lexical matching to incorporate semantic information. It emphasizes that purely word-focused similarity measures may yield less accurate results and lack human interpretability, especially in the context of short text comparisons. To address this, the paper introduces three distinct methods that not only consider the text's words but also integrate semantic information into their feature vectors, thus enhancing the accuracy and interpretability of similarity measures.

The methods proposed in the paper revolve around cosine similarity, a widely used metric in text analysis. The first method employs tf-idf (term frequency-inverse document frequency) vectors, a standard technique in information retrieval, to compute cosine similarity. The second method utilizes word embeddings, which capture semantic relationships between words, to calculate cosine similarity. Lastly, the paper introduces soft cosine similarity, which also leverages word embeddings but incorporates additional semantic information to enhance similarity calculations.

Through experimentation and evaluation, the paper compares the performance of these three methods in determining semantic similarities between short news texts. Among the methods explored, cosine similarity using tf-idf vectors emerges as the most effective in finding similarities and generating easily interpretable results. The similarity scores obtained from this method are not only accurate but also suitable for direct application in various information retrieval systems and other natural language processing tasks.

The study sets the stage by highlighting the significance of semantic similarity in various NLP applications, such as document classification, clustering, retrieval, translation, and summarization. It emphasizes the limitations of purely lexical matching approaches, especially when dealing with short texts, and advocates for methods that incorporate semantic features for more meaningful and accurate similarity computations.

The paper's main objective is to provide a comparative analysis of semantic similarity computation methods for short texts, with a focus on maximizing human interpretability. It explains the fundamental idea of computing text similarities by deriving feature vectors from documents and measuring distances between these features. The discussion encompasses distance metrics like Euclidean distance, Cosine distance, Jensen Shannon Distance, and Word Mover distance, highlighting their relevance in text similarity computation.

Overall, the paper contributes to the field by presenting practical methods that bridge the gap between lexical matching and semantic similarity, ultimately improving the efficiency and interpretability of text similarity measures, especially in the context of short texts.

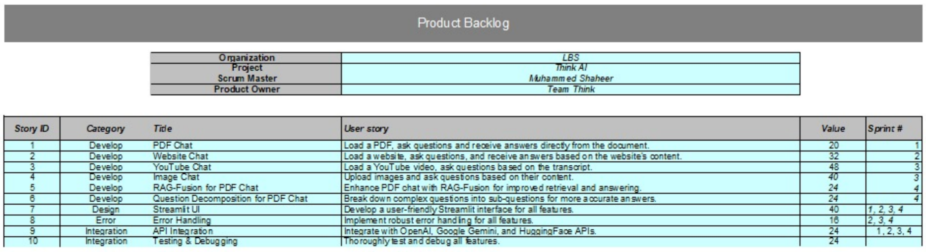
Chapter 5

PROJECT MANAGEMENT

Our project, an AI-Powered Chatbot for PDF Document Processing, Chatting with Websites Using AI, YouTube Video Summarizer Using Chatbot, and Image-Based Question Answering, requires a comprehensive project management approach. We will adopt an agile methodology, breaking the project into manageable sprints with clear goals for each phase. The initial sprint will focus on setting up the foundational components, including integrating Google's Generative AI model for chatbot interactions and developing algorithms for PDF document processing and website chatting. Subsequent sprints will tackle specific features such as YouTube video summarization and image-based question answering, leveraging technologies like LangChain and Chroma for efficient text processing and analysis. Regular meetings and progress reviews will ensure alignment with project objectives and timely adjustments to deliver a user-friendly and effective AI-powered solution for diverse information retrieval tasks.

5.1 SCRUM

5.1.1 User Stories



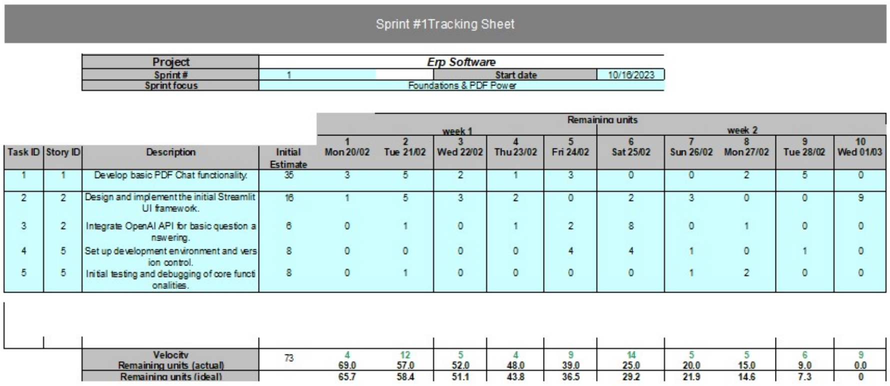
Product Backlog

Organization	*LBS*
Project	*Think AI*
Scrum Master	*Muhammed Shaheer*
Product Owner	*Team Think*

Story ID	*Category*	*Title*	*User story*	*Value*	*Sprint #*
1	Develop	PDF Chat	Load a PDF, ask questions and receive answers directly from the document.	20	1
2	Develop	Website Chat	Load a website, ask questions, and receive answers based on the website's content.	32	2
3	Develop	YouTube Chat	Load a YouTube video, ask questions based on the transcript.	48	3
4	Develop	Image Chat	Upload images and ask questions based on their content.	*40*	*3*
5	Develop	RAG-Fusion for PDF Chat	Enhance PDF chat with RAG-Fusion for improved retrieval and answering.	*24*	*4*
6	Develop	Question Decomposition for PDF Chat	Break down complex questions into sub-questions for more accurate answers.	*24*	*4*
7	Design	Streamlit UI	Develop a user-friendly Streamlit interface for all features.	40	*1, 2, 3, 4*
8	Error	Error Handling	Implement robust error handling for all features.	16	*2, 3, 4*
9	Integration	API Integration	Integrate with OpenAI, Google Gemini, and HuggingFace APIs.	24	1, 2, 3, 4
10	Integration	Testing & Debugging	Thoroughly test and debug all features.	24	

Figure 5.1: User Stories

5.1.2 Sprint 1



Sprint #1Tracking Sheet

Project	***Erp Software***		
Sprint #	1	Start date	10/16/2023
Sprint focus	Foundations & PDF Power		

				Remaining units									
				week 1					week 2				
Task ID	Story ID	Description	Initial Estimate	1 Mon 20/02	2 Tue 21/02	3 Wed 22/02	4 Thu 23/02	5 Fri 24/02	6 Sat 25/02	7 Sun 26/02	8 Mon 27/02	9 Tue 28/02	10 Wed 01/03
1	1	Develop basic PDF Chat functionality.	35	3	5	2	1	3	0	0	2	5	0
2	2	Design and implement the initial Streamlit UI framework.	16	1	5	3	2	0	2	3	0	0	9
3	2	Integrate OpenAI API for basic question a nswering.	6	0	1	0	1	2	8	0	1	0	0
4	5	Set up development environment and vers ion control.	8	0	0	0	0	4	4	1	0	1	0
5	5	Initial testing and debugging of core functi onalities.	8	0	1	0	0	0	0	1	2	0	0
		Velocity	73	4	12	5	4	9	14	5	5	6	9
		Remaining units (actual)		69.0	57.0	52.0	48.0	39.0	25.0	20.0	15.0	9.0	0.0
		Remaining units (ideal)		65.7	58.4	51.1	43.8	36.5	29.2	21.9	14.6	7.3	0

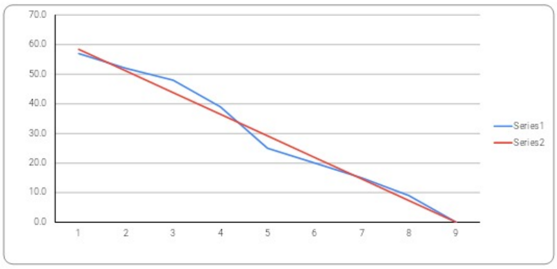
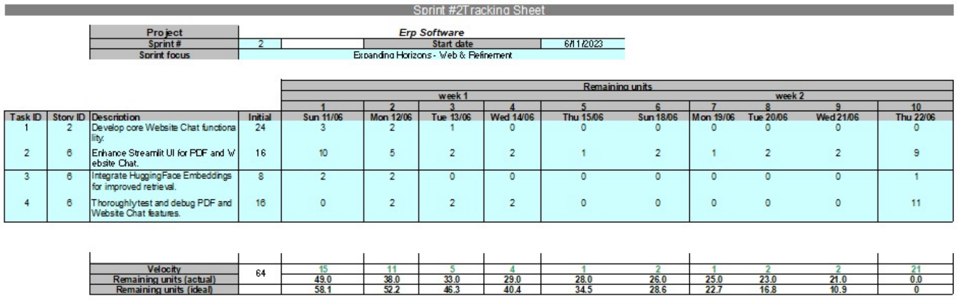


Figure 5.2: Sprint 1 and Graph

5.1.3 Sprint 2



Sprint #2Tracking Sheet

Project	*Erp Software*		
Sprint #	2	Start date	6/11/2023
Sprint focus	Expanding Horizons - Web & Refinement		

				Remaining units									
				week 1				week 2					
				1	2	3	4	5	6	7	8	9	10
Task ID	Story ID	Description	Initial	Sun 11/06	Mon 12/06	Tue 13/06	Wed 14/06	Thu 15/06	Sun 18/06	Mon 19/06	Tue 20/06	Wed 21/06	Thu 22/06
1	2	Develop core Website Chat functionality.	24	3	2	1	0	0	0	0	0	0	0
2	6	Enhance Streamlit UI for PDF and Website Chat.	16	10	5	2	2	1	2	1	2	2	9
3	6	Integrate HuggingFace Embeddings for improved retrieval.	8	2	2	0	0	0	0	0	0	0	1
4	6	Thoroughlytest and debug PDF and Website Chat features.	16	0	2	2	2	0	0	0	0	0	11

Velocity	64	15	11	5	4	1	2	1	2	2	21
Remaining units (actual)		49.0	38.0	33.0	29.0	28.0	26.0	25.0	23.0	21.0	0.0
Remaining units (ideal)		58.1	52.2	46.3	40.4	34.5	28.6	22.7	16.8	10.9	0

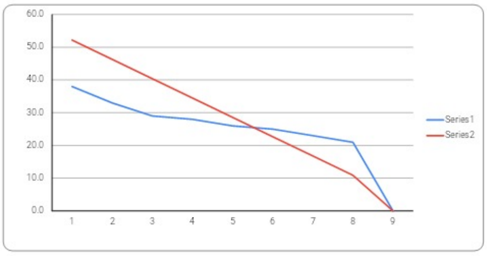
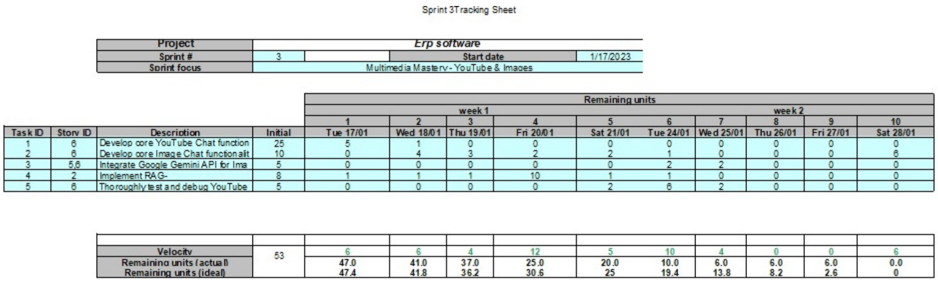


Figure 5.3: Sprint 2 and Graph

5.1.4 Sprint 3



Sprint 3Tracking Sheet

Project	*Erp software*			
Sprint #	3		Start date	1/17/2023
Sprint focus	Multimedia Mastery - YouTube & Images			

				Remaining units									
				week 1				week 2					
				1	2	3	4	5	6	7	8	9	10
Task ID	Story ID	Description	Initial	Tue 17/01	Wed 18/01	Thu 19/01	Fri 20/01	Sat 21/01	Tue 24/01	Wed 25/01	Thu 26/01	Fri 27/01	Sat 28/01
1	6	Develop core YouTube Chat function	25	5	1	0	0	0	0	0	0	0	0
2	6	Develop core Image Chat functionalit	10	0	4	3	2	2	1	0	0	0	6
3	5,6	Integrate Google Gemini API for Ima	5	0	0	0	0	0	2	2	0	0	0
4	2	Implement RAG-	8	1	1	1	10	1	1	0	0	0	0
5	6	Thoroughly test and debug YouTube	5	0	0	0	0	2	6	2	0	0	0

Velocity	53	6	6	4	12	5	10	4	0	0	6
Remaining units (actual)		47.0	41.0	37.0	25.0	20.0	10.0	6.0	6.0	6.0	0.0
Remaining units (ideal)		47.4	41.8	36.2	30.6	25	19.4	13.8	8.2	2.6	0

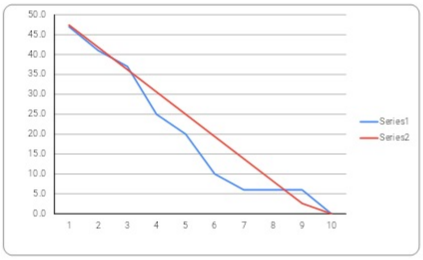
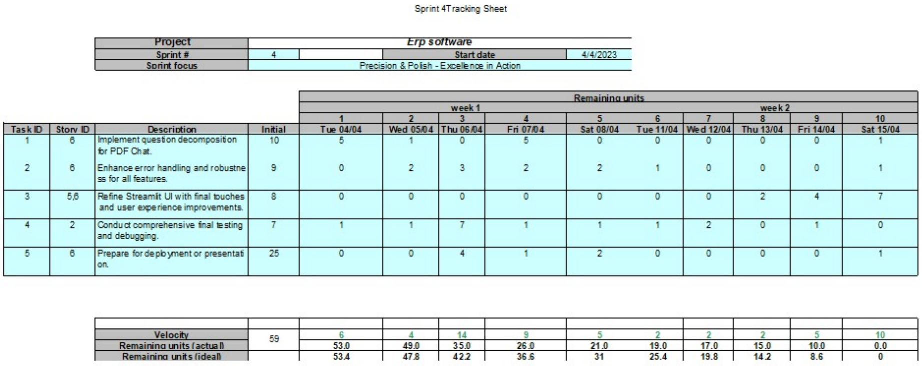


Figure 5.4: Sprint 3 and Graph

5.1.5 Sprint 4



Sprint 4Tracking Sheet

Project	Erp software			
Sprint #	4		Start date	4/4/2023
Sprint focus	Precision & Polish - Excellence in Action			

				Remaining units									
				week 1					week 2				
				1	2	3	4	5	6	7	8	9	10
Task ID	Story ID	Description	Initial	Tue 04/04	Wed 05/04	Thu 06/04	Fri 07/04	Sat 08/04	Tue 11/04	Wed 12/04	Thu 13/04	Fri 14/04	Sat 15/04
1	6	Implement question decomposition for PDF Chat.	10	5	1	0	5	0	0	0	0	0	1
2	6	Enhance error handling and robustness for all features.	9	0	2	3	2	2	1	0	0	0	1
3	5,6	Refine Streamlit UI with final touches and user experience improvements.	8	0	0	0	0	0	0	0	2	4	7
4	2	Conduct comprehensive final testing and debugging.	7	1	1	7	1	1	1	2	0	1	0
5	6	Prepare for deployment or presentation.	25	0	0	4	1	2	0	0	0	0	1

Velocity	59	6	4	14	9	5	2	2	2	5	10
Remaining units (actual)		53.0	49.0	35.0	26.0	21.0	19.0	17.0	15.0	10.0	0.0
Remaining units (ideal)		53.4	47.8	42.2	36.6	31	25.4	19.8	14.2	8.6	0

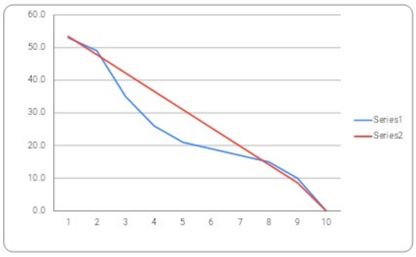


Figure 5.5: Sprint 4 and Graph

Chapter 6

SYSTEM DESIGN

6.1 Input Text Preprocessing:

Handles text extraction, cleaning, and formatting from various input sources like PDF documents, websites, and user queries.Utilizes techniques such as tokenization, stop-word removal, and normalization to prepare the text for further processing.

6.2 Embedding Generation:

Converts preprocessed text into vector embeddings using advanced embedding models like GoogleGenerativeAIEmbeddings and HuggingFaceEmbeddings. Represents text data in a high-dimensional vector space for semantic analysis and similarity calculations.

6.3 Indexing and Search:

Indexes the generated embeddings using vector stores like Chroma or FAISS for efficient storage and retrieval. Enables fast and accurate search operations to retrieve relevant information based on user queries.

6.4 User Query Preprocessing:

Preprocesses user queries by applying similar text preprocessing techniques to ensure compatibility with the indexed data. Normalizes and tokenizes user input for accurate matching and retrieval of relevant context.

6.5 Context Retrieval:

Retrieves contextually relevant information based on user queries and current conversation context. Utilizes retrieval mechanisms like ConversationalRetrievalChain or RAG (Retrieval Augmented Generation) to provide accurate responses.

6.6 Overall Architecture

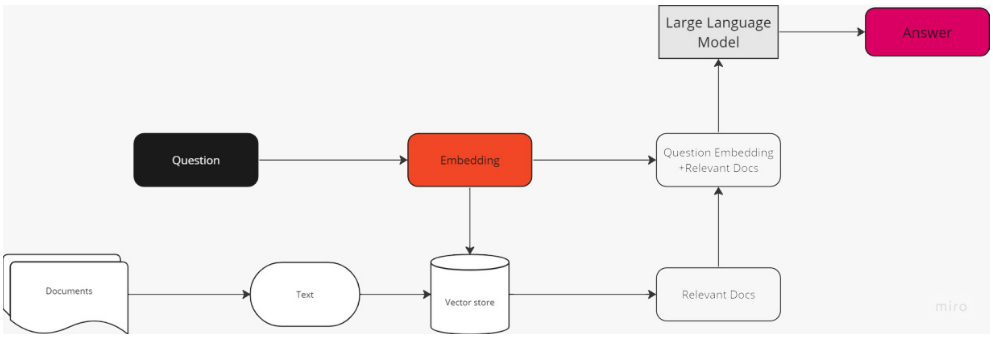


Figure 6.1: Overall Architecture

6.7 Detailed Architecture

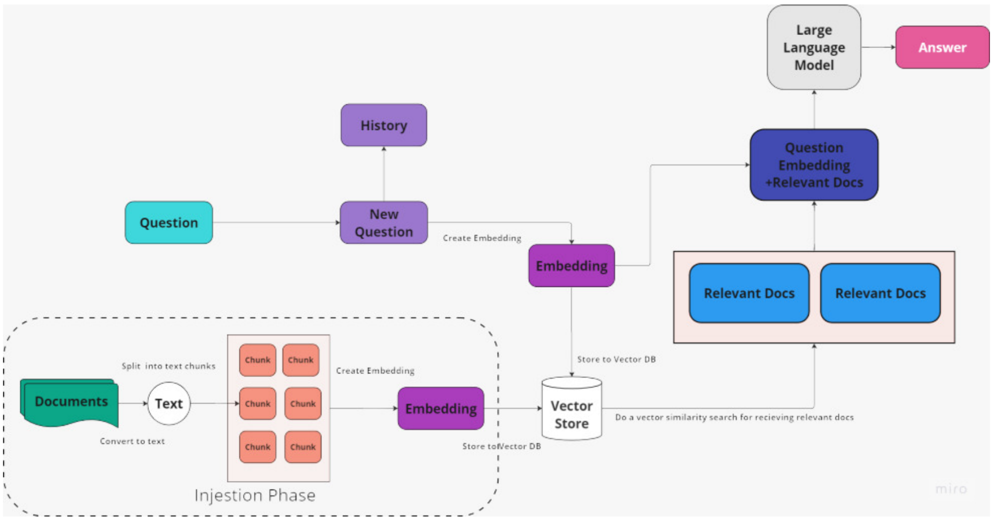


Figure 6.2: Detailed Architecture

Chapter 7

METHODOLOGY

7.1 Developing an AI-Powered Chatbot for PDF Document Processing

The first step in our methodology is to understand the problem statement, which involves developing an AI chatbot using Google Gemini Pro and LangChain. This chatbot utilizes FAISS and Google Embedding to streamline the processing of multiple PDF documents, aiming to improve efficiency and user experience. We begin by importing necessary libraries such as os, utils, streamlit (as st), and various components from LangChain and OpenAI libraries. The environment is configured using the dotenv package for handling environment variables.

7.1.1 User Interface Development:

The next phase involves developing the user interface using Streamlit. This includes setting page configurations, displaying a subheader for user instructions, and implementing file upload functionality for PDF documents.

7.1.2 Document Loading and Processing:

Upon uploading PDF documents, the system loads the documents using PyPDFLoader and splits them into manageable chunks using RecursiveCharacterTextSplitter.

7.1.3 Text Embeddings and Vectorization:

The text from the documents is then processed using GoogleGenerativeAIEmbeddings to generate embeddings, and a vector store is created using Chroma for efficient retrieval and analysis.

7.1.4 Chatbot Functionality:

The core functionality of the chatbot is implemented using ChatOpenAI, which allows users to ask questions related to the loaded documents. The chatbot uses a conversational retrieval chain to provide relevant answers based on the context and user queries.

7.1.5 Sub-Question Generation:

To enhance user interactions, the chatbot generates sub-questions related to the user's input question using a predefined template and AI-powered decomposition.

7.1.6 Answer Generation using RAG:

The chatbot utilizes the Retrieval Augmented Generation (RAG) approach to generate answers to user questions. It leverages background question-answer pairs and contextual information to provide accurate and informative responses.

7.1.7 User Interaction and Feedback:

The chatbot's responses and interactions with users are displayed in the Streamlit interface, allowing for real-time communication. Users can also view relevant document sources for the chatbot's answers.

7.1.8 Iterative Development and Testing:

Throughout the development process, iterative testing and refinement are carried out to ensure the chatbot's functionality, accuracy, and user-friendliness.

7.1.9 Deployment and Integration:

Once the chatbot is finalized, it can be deployed for use, integrated into existing systems or platforms, and further enhanced based on user feedback and usage analytics.

7.1.10 Documentation and Reporting:

Finally, comprehensive documentation is prepared, detailing the chatbot's architecture, functionality, usage instructions, and potential future enhancements. A report summarizing the development process, challenges faced, and key learnings is also generated for stakeholders and readers.

7.2 Retrieval Augmented Generation(RAG)

RAG, short for Retrieval Augmented Generation, is an innovative approach that combines the strengths of information retrieval and natural language generation to improve the quality and relevance of generated text. It is particularly useful in tasks such as question answering, content summarization, and conversational AI systems.

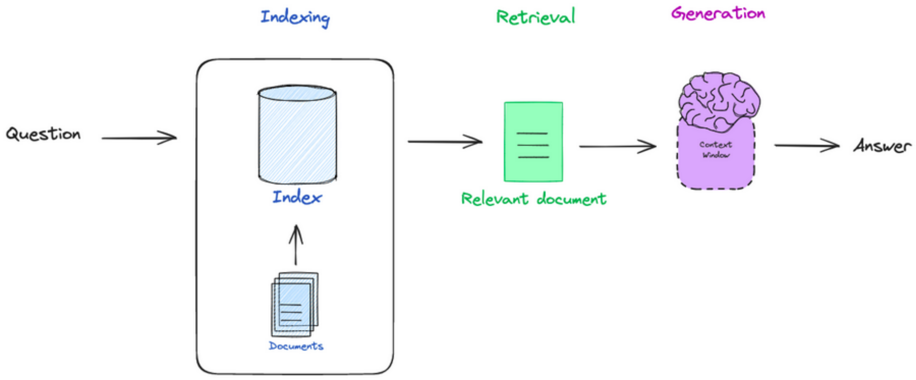


Figure 7.1: RAG motivation

7.2.1 Input Query:

The process starts with a user inputting a query or question into the system. This query can be in natural language or structured format.

7.2.2 Indexing and Retrieval:

RAG first indexes a large knowledge base or corpus of documents. When a user query is received, RAG retrieves relevant information from this indexed data based on the query's context and keywords.

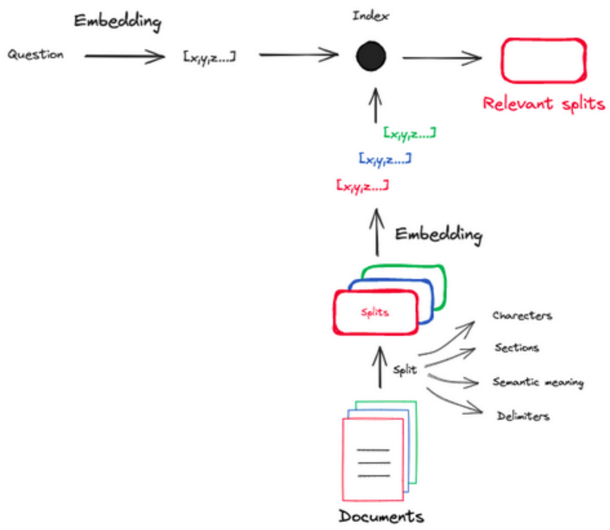


Figure 7.2: Indexing and Retrieval

7.2.3 Retrieval powered via Similarity search:

which refers to the process of retrieving relevant information or documents by searching for similarities between the input query or context and the stored data. This approach uses advanced techniques such as vector embeddings and similarity metrics to find the most relevant content based on similarity scores, improving the accuracy and effectiveness of information retrieval in RAG systems.

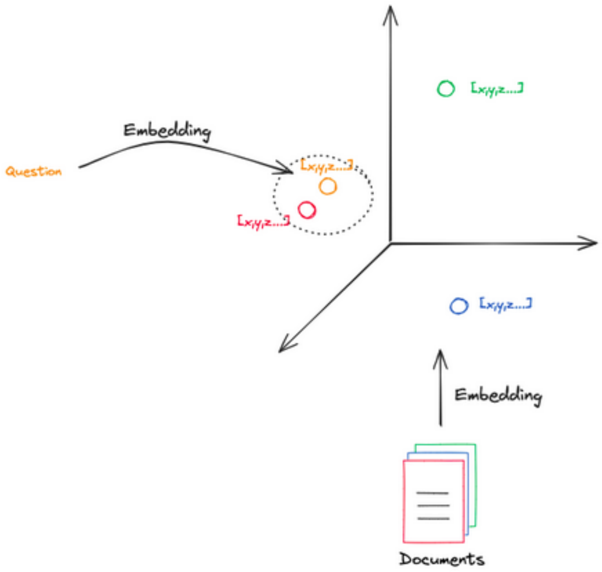


Figure 7.3: Retrieval powered via similarity search

7.2.4 Vector stores:

which are repositories or databases that store vector representations of text data. These vector representations are numerical representations of text that capture semantic and contextual information about the text. In RAG, vector stores are used for efficient retrieval of relevant information during the generation process. When a query is made, RAG searches the vector store to retrieve contextually relevant information, which is then used in the generation of responses or content.

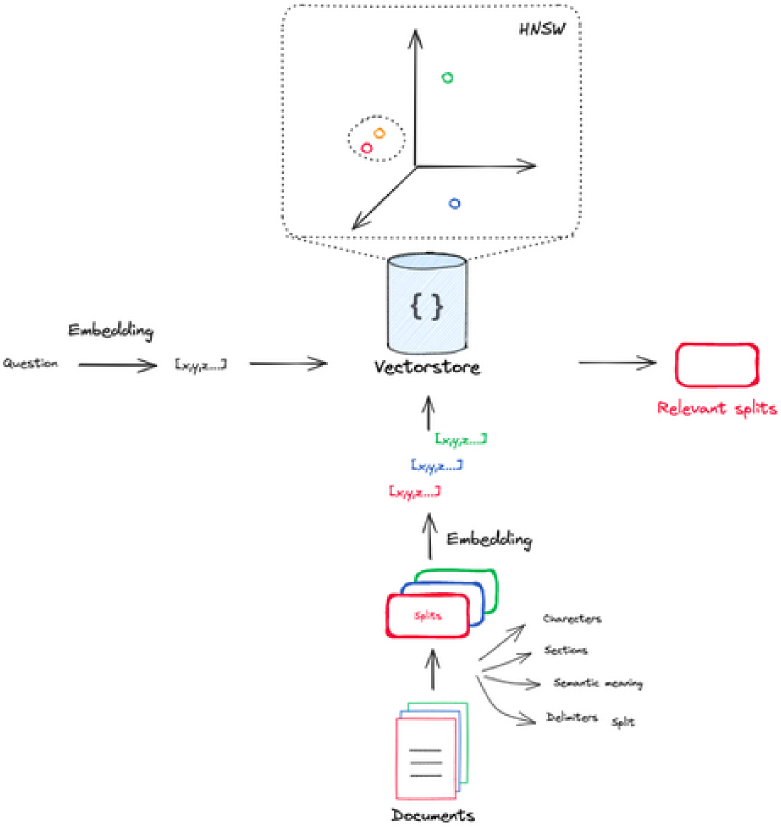


Figure 7.4: Vectorstores

7.2.5 Query Translation:

RAG translates the user query into a format that the system can understand and process efficiently. This translation step ensures that the query is interpreted accurately for retrieval and generation.

7.2.6 Decomposition:

If the query is complex, RAG may decompose it into smaller sub-questions or sub-problems. This decomposition strategy helps in retrieving more accurate and specific information.

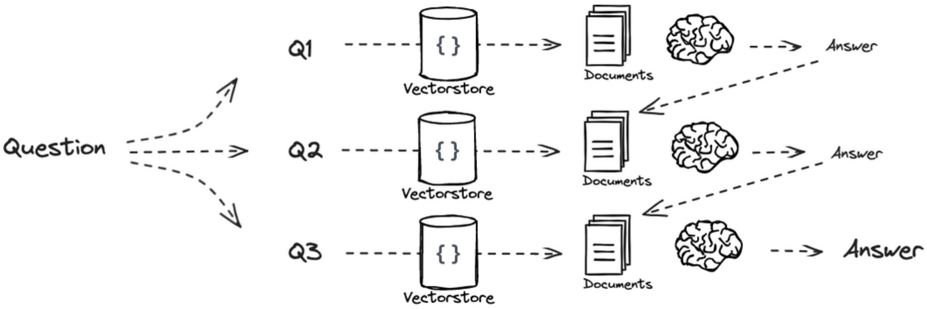


Figure 7.5: Decomposition

7.2.7 Generation:

Finally, RAG generates a response based on the retrieved information and the query's context. The generated response is coherent, contextually relevant, and aims to provide a comprehensive answer to the user's query.

Overall, RAG enhances the capabilities of AI systems by integrating advanced retrieval and generation techniques, ultimately improving the accuracy, relevance, and user experience in information retrieval and text generation tasks.

7.3 Chatting with Websites Using AI

The initial step in developing the application was to identify the problem of engaging in conversational interactions with websites efficiently. Users often struggle to find relevant information or engage in meaningful conversations with website content. Understanding the requirements involved analyzing the need for integrating natural language processing (NLP) capabilities, information retrieval, and conversational AI into a user-friendly interface. The goal was to create a chatbot that can interact with website content and provide relevant responses to user queries.

7.3.1 Technology Selection:

The selected technologies for this project include Streamlit for the user interface, LangChain for NLP capabilities, OpenAI for language models, Google Generative AI for embeddings, and Chroma for vector stores. These technologies were chosen based on their capabilities to handle text processing, information retrieval, and AI-driven conversation generation.

7.3.2 Application Architecture:

The application architecture consists of different components:

- WebBaseLoader: Retrieves website content for processing.

- RecursiveCharacterTextSplitter: Splits the retrieved text into manageable chunks.

- GoogleGenerativeAIEmbeddings and Chroma: Create a vector store from the text chunks to facilitate efficient retrieval and analysis.

- ChatOpenAI and ChatPromptTemplate: Utilized for conversational AI capabilities, allowing the chatbot to generate responses based on user queries and context.

- create history aware retriever and create retrieval chain: Create a retrieval chain that considers the conversation history and context, improving the relevance of retrieved information.

- create stuff documents chain: Generates responses to user queries based on the retrieved context and information.

7.3.3 Implementation Steps:

- User inputs a website URL.

- The application retrieves website content and processes it using LangChain and other libraries.

- The content is indexed and stored in a vector store for efficient retrieval.

- Users can then input queries or messages into the chat interface.

- The chatbot, powered by ChatOpenAI, retrieves relevant information from the indexed content and generates responses based on the context and conversation history.

- The conversation continues, with the chatbot providing relevant and coherent answers to user queries.

7.3.4 Testing and Optimization:

Extensive testing was conducted to ensure the chatbot's accuracy, relevancy, and user experience. Optimization techniques were implemented to enhance the chatbot's performance, including refining retrieval strategies, improving response generation, and handling diverse user inputs.

7.3.5 Deployment and User Experience:

After successful testing and optimization, the application was deployed using Streamlit for a user-friendly interface. The chatbot's usability and responsiveness were key considerations in providing a seamless user experience.

Overall, the methodology involved a comprehensive approach to developing an AI-powered chatbot that can effectively engage with website content and provide valuable information to users in a conversational manner.

7.4 YouTube Video Summarizer Using Chatbot

The primary objective was to develop an application capable of summarizing YouTube videos using AI-driven chatbot technology. Users often face challenges in extracting key information or insights from lengthy video content, making a summarization tool valuable. The requirements involved integrating YouTube video retrieval, transcript extraction, NLP-based summarization, and conversational AI capabilities into a cohesive application. The aim was to create a user-friendly tool that could generate concise summaries from video content. The chosen technologies for this project include Streamlit for the user interface, LangChain for NLP capabilities, Google Generative AI for embeddings, and Chroma for vector stores. These technologies were selected based on their suitability for handling text processing, summarization tasks, and AI-driven conversation generation.

7.4.1 Application Architecture:

The application architecture consists of several key components:

- YouTubeLoader: Retrieves the YouTube video content for processing.

- RecursiveCharacterTextSplitter: Splits the video transcript into manageable chunks.

- GoogleGenerativeAIEmbeddings and Chroma: Create a vector store from the text chunks to facilitate efficient retrieval and analysis.

- ChatOpenAI and ChatPromptTemplate: Utilized for conversational AI capabilities, allowing the chatbot to generate summaries based on user queries and context.

- create history aware retriever and create retrieval chain: Create a retrieval chain that considers conversation history and context, improving the relevance of generated summaries.

7.4.2 Implementation Steps:

- User inputs a YouTube video link.

- The application retrieves the video transcript and processes it using LangChain and other libraries.

- The transcript is indexed and stored in a vector store for efficient retrieval and summarization.

- Users can input queries or messages into the chat interface to interact with the chatbot.

- The chatbot retrieves relevant information from the video transcript and generates a summary based on user queries and context.

- The summarized content is presented to the user in a conversational manner.

7.4.3 Testing and Optimization:

Extensive testing was conducted to ensure the accuracy and coherence of the generated summaries. Optimization techniques were implemented to improve the chatbot's performance in summarizing video content accurately and efficiently.

7.4.4 Deployment and User Experience:

After successful testing and optimization, the application was deployed using Streamlit for a user-friendly interface. The chatbot's usability and responsiveness were key considerations in providing a seamless user experience.

Potential future enhancements include incorporating multi-language support for video transcripts, enhancing the summarization algorithms for better accuracy, and integrating more advanced AI models for improved conversation quality and summarization capabilities.

Overall, this involved a comprehensive approach to developing a YouTube video summarizer using AI-driven chatbot technology, aiming to provide users with a convenient and efficient tool for extracting valuable insights from video content.

7.5 Image-based Question Answering

The aim of this project was to develop a Streamlit web application, called Gemini Pro Vision, that utilizes Google's Generative AI model (Gemini Pro) for image-based question answering, specifically focusing on understanding invoices and receipts.

7.5.1 Setting Up Google API Key:

The initial step involved configuring the Google API key for accessing Google's Generative AI services. This required loading the API key from environment variables using the ‘dotenv‘ library and setting up the GenerativeModel with specific generation configurations and safety settings.

7.5.2 Application Components:

- GenerativeModel Setup: The ‘GenerativeModel‘ from ‘google.generativeai‘ was configured with settings such as temperature, top-k sampling, and safety thresholds for categories like harassment, hate speech, sexually explicit content, and dangerous content.

- Input Image Setup: The application allows users to upload images (invoices or receipts) and select an image for asking questions. Uploaded images are processed and prepared for input to the GenerativeModel.

- Question Prompt: Users can input a question prompt related to the uploaded image, specifying the context for the AI model to generate a response.

7.5.3 Main Functionality:

It takes the input prompt, uploaded files (images), selected image index, and question prompt to generate a response using the Gemini Pro model. The response is then displayed in the Streamlit app.

7.5.4 User Interface:

The Streamlit app's user interface includes:

- A sidebar for uploading images and selecting images for questions.

- A text input field for entering the question prompt.

- A button to generate the response based on the uploaded image and question prompt.

7.5.5 Workflow Execution:

- Users upload one or more images (invoices or receipts) via the sidebar.

- They select an image from the uploaded images to ask questions about.

- Users input a question prompt related to the selected image.

- Upon clicking the ”Generate Response” button, the application processes the image and question prompt to generate a response using the Gemini Pro model.

- The generated response is displayed in the app, providing answers or insights based on the uploaded image and user query.

7.5.6 Error Handling:

The application includes error handling to manage scenarios where users fail to upload images or select an image for questions, ensuring a smooth user experience.

Gemini Pro Vision Streamlit App provides a convenient way to leverage Google’s Generative AI model for image-based question answering. Future enhancements could include improving the UI/UX, integrating additional AI functionalities, and enhancing the model’s capabilities for better accuracy and context understanding.

Chapter 8

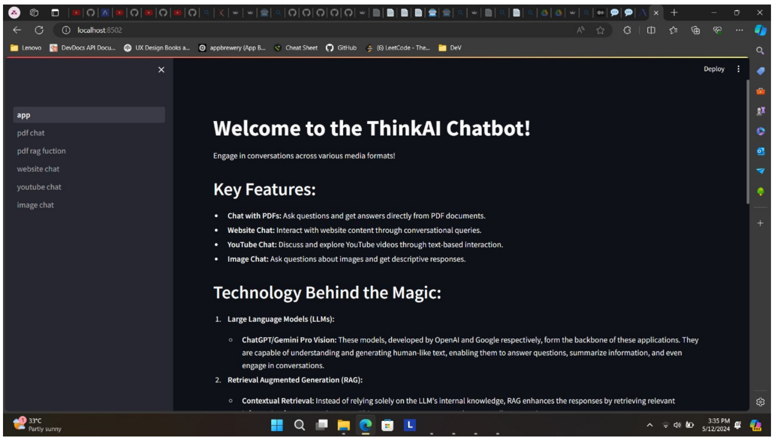
RESULTS

8.1 Developing an AI-Powered Chatbot for PDF Document Processing

- Utilizes Google Gemini Pro and LangChain for AI capabilities.

- Focuses on processing multiple PDF documents efficiently using FAISS vector embeddings.

- Features include text extraction, processing, and embedding, along with chatbot integration for streamlined interaction with PDF content.



Welcome to the ThinkAI Chatbot!

Engage in conversations across various media formats!

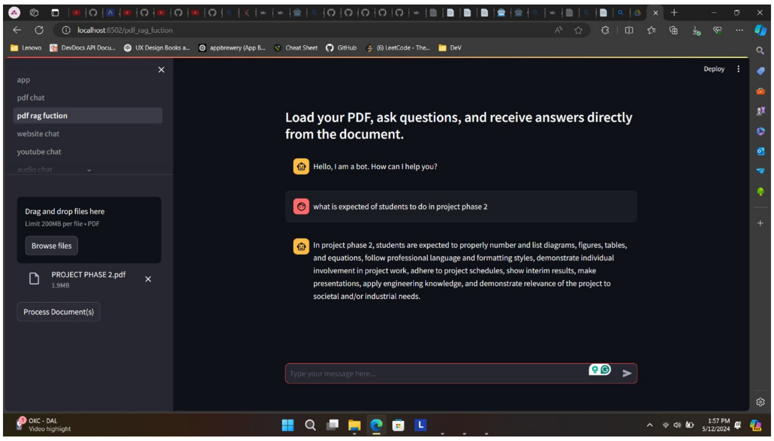
Key Features:

- **Chat with PDFs:** Ask questions and get answers directly from PDF documents.
- **Website Chat:** Interact with website content through conversational queries.
- **YouTube Chat:** Discuss and explore YouTube videos through text-based interaction.
- **Image Chat:** Ask questions about images and get descriptive responses.

Technology Behind the Magic:

1. **Large Language Models (LLMs):**
   - **ChatGPT/Gemini Pro Vision:** These models, developed by OpenAI and Google respectively, form the backbone of these applications. They are capable of understanding and generating human-like text, enabling them to answer questions, summarize information, and even engage in conversations.
2. **Retrieval Augmented Generation (RAG):**
   - **Contextual Retrieval:** Instead of relying solely on the LLM's internal knowledge, RAG enhances the responses by retrieving relevant

Figure 8.1: Home page



Load your PDF, ask questions, and receive answers directly from the document.

Hello, I am a bot. How can I help you?

what is expected of students to do in project phase 2

In project phase 2, students are expected to properly number and list diagrams, figures, tables, and equations, follow professional language and formatting styles, demonstrate individual involvement in project work, adhere to project schedules, show interim results, make presentations, apply engineering knowledge, and demonstrate relevance of the project to societal and/or industrial needs.

PROJECT PHASE 2.pdf 1.9MB

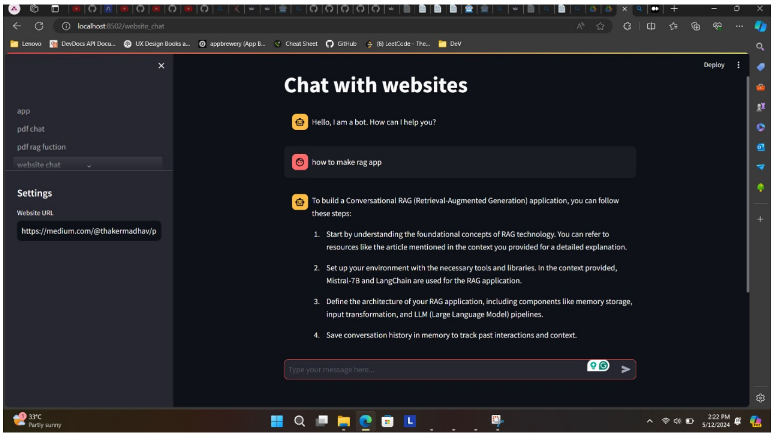
Figure 8.2: Chat with PDF

8.2 Chatting with Websites Using AI

- Incorporates LangChain and ChatOpenAi for AI-powered interactions.

- Aimed at enhancing user experience while engaging with websites through conversational AI.

- Features may include real-time responses, information retrieval, and personalized interactions based on user queries.



Chat with websites

Settings

Website URL: https://medium.com/@thakermadhav/p

Hello, I am a bot. How can I help you?

how to make rag app

To build a Conversational RAG (Retrieval-Augmented Generation) application, you can follow these steps:

1. Start by understanding the foundational concepts of RAG technology. You can refer to resources like the article mentioned in the context you provided for a detailed explanation.
2. Set up your environment with the necessary tools and libraries. In the context provided, Mistral-7B and LangChain are used for the RAG application.
3. Define the architecture of your RAG application, including components like memory storage, input transformation, and LLM (Large Language Model) pipelines.
4. Save conversation history in memory to track past interactions and context.

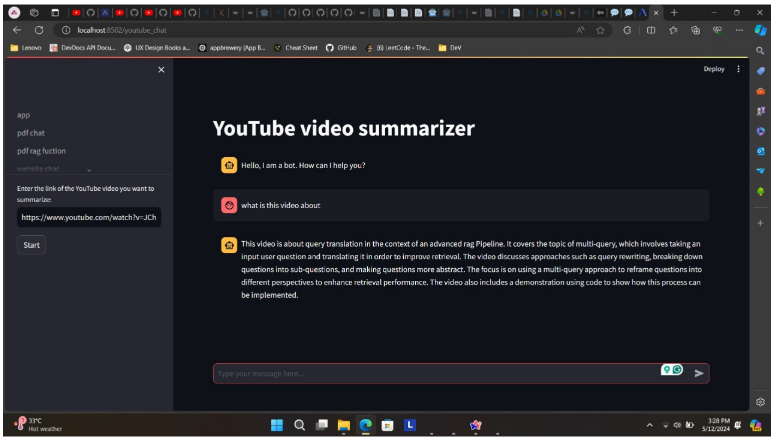
Figure 8.3: Chat with Websites

8.3 YouTube Video Summarizer Using Chatbot

- Leverages LangChain, GoogleGenerativeAIEmbeddings, and YouTubeTranscriptApi for video analysis and summarization.

- Focuses on extracting key insights from YouTube videos using AI-driven techniques.

- Capabilities may include generating summaries, answering questions about video content, and providing contextual understanding.



YouTube video summarizer

Hello, I am a bot. How can I help you?

what is this video about

This video is about query translation in the context of an advanced rag Pipeline. It covers the topic of multi-query, which involves taking an input user question and translating it in order to improve retrieval. The video discusses approaches such as query rewriting, breaking down questions into sub-questions, and making questions more abstract. The focus is on using a multi-query approach to reframe questions into different perspectives to enhance retrieval performance. The video also includes a demonstration using code to show how this process can be implemented.

Enter the link of the YouTube video you want to summarize:
https://www.youtube.com/watch?v=JCh

Type your message here...

Figure 8.4: Youtube Video Summarizer

8.4 Image-based Question Answering

- Utilizes GoogleGenerativeAIEmbeddings and image analysis techniques.

- Designed for answering questions based on image inputs, possibly related to invoices, receipts, or visual data.

- Features may include image processing, content understanding, and generating accurate responses to user queries.

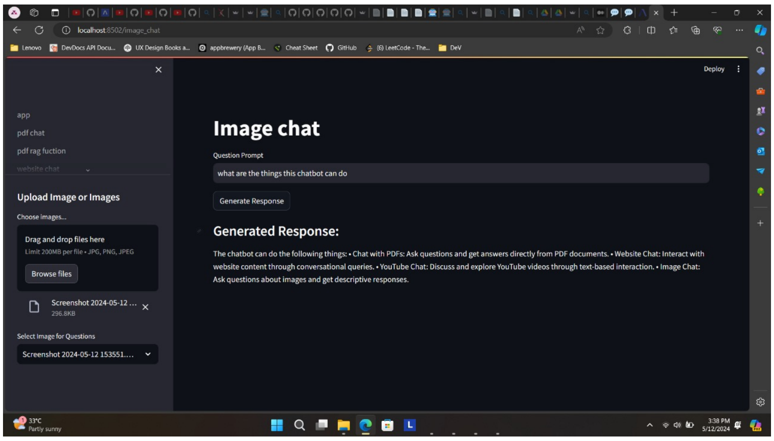


Image chat

Question Prompt
what are the things this chatbot can do

Generated Response:

The chatbot can do the following things: • Chat with PDFs: Ask questions and get answers directly from PDF documents. • Website Chat: Interact with website content through conversational queries. • YouTube Chat: Discuss and explore YouTube videos through text-based interaction. • Image Chat: Ask questions about images and get descriptive responses.

Upload Image or Images
Choose images...
Drag and drop files here
Limit 200MB per file • JPG, PNG, JPEG
Screenshot 2024-05-12 ...
296.8KB

Select Image for Questions
Screenshot 2024-05-12 153551....

Figure 8.5: Chat with Image

Considering these components, the overarching project aims to develop a comprehensive AI ecosystem for information processing, interaction with various media types (PDFs, websites, videos, images), and efficient question-answering capabilities. It integrates advanced AI models, text/image analysis techniques, and chatbot functionalities to create a seamless and intelligent user experience across different information sources.

Chapter 9

CONCLUSION

In conclusion, our project represents a significant advancement in AI-driven technology aimed at enhancing various aspects of digital interaction and information processing. Through the development of an AI-Powered Chatbot for PDF Document Processing, we have streamlined the cumbersome task of managing and extracting insights from complex PDF files, thereby revolutionizing the way researchers, students, and professionals interact with textual data. Additionally, our integration of AI technology for Chatting with Websites Using AI has opened up new avenues for seamless communication and information retrieval from online sources, facilitating efficient knowledge acquisition and interaction. The incorporation of a YouTube Video Summarizer Using Chatbot further extends the utility of our AI capabilities by providing users with concise and informative summaries of video content, saving time and effort in accessing valuable information. Moreover, our Image-based question answering feature demonstrates the versatility of our AI system in understanding and responding to diverse types of queries, catering to a wide range of user needs. Overall, our project underscores the immense potential of AI-driven solutions in enhancing productivity, improving information access, and facilitating intelligent interaction across various digital platforms.

Chapter 10

FUTURE WORKS

In future iterations of our project, we aim to enhance the capabilities of our AI-powered chatbot across multiple fronts. Firstly, we plan to integrate advanced natural language processing techniques to improve the chatbot's understanding and response generation, especially when dealing with complex PDF document processing. Additionally, we envision incorporating more sophisticated conversational models and algorithms to enable seamless interactions with websites using AI, facilitating efficient information retrieval and interaction. Moreover, we aim to expand the chatbot's functionality to include video summarization capabilities, leveraging chatbot technology to generate concise and informative summaries of YouTube videos. Furthermore, we plan to strengthen the chatbot's image-based question-answering capabilities by integrating cutting-edge computer vision algorithms, allowing for more accurate and comprehensive responses to user queries based on visual content. Overall, our future work will focus on advancing the chatbot's intelligence, versatility, and usability across various domains, ensuring a more engaging and valuable user experience.

References

[1] Denny Zhou, Nathanael Sch arli, Le Hou† Jason Wei† Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Google Research, Brain Team,"Least-To-Most Prompting Enables Complex Reasoning in Large Language Models",2023, https://arxiv.org/pdf/2205.10625.pdf

[2] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, Ashish Sabharwal, "Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions", 2022, https://arxiv.org/abs/2212.10509.pdf

[3] Aditya Jain, Divij Bhatia, Manish K Thakur [2017], Extractive Text Summarization using Word Vector Embedding , https://ieeexplore.ieee.org/document/8320258

[4] Addi Ait-mlouk AND Lili Jiang [2020], KBot: A Knowledge Graph Based ChatBot for Natural Language Understanding Over Linked Data , https://ieeexplore.ieee.org

[5] Zhao Yan, Nan Duan, Junwei Bao, Peng Chen, Ming Zhou, Zhou Li, Jianshe Zhou [2016] , DocChat: An Information Retrieval Approach for Chatbot Engines Using Unstructured Documents, https://aclanthology.org/P16-1049.pdf

[6] Arjun Pesaru, Taranveer Singh Gill, Archit Reddy Tangella [2023] , AI ASSISTANT FOR DOCUMENT MANAGEMENT USING LANG CHAIN AND PINECONE , https://www.irjmets.com

[7] Haritha Akkineni, P. V. S. Lakshmi, and Lasya Sarada [2022] , Design and Development of Retrieval-Based Chatbot Using Sentence Similarity, https://link.springer.com

[8] Oguzhan Topsakal1, and Tahir Cetin Akinci [2023] , Creating Large Language Model Applications Utilizing LangChain: A Primer on Developing LLM Apps Fast , https://as-proceeding.com

[9] Andreas Lommatzsch and Jonas Katins [2019] , An Information Retrieval-based Approach for Building Intuitive Chatbots for Large Knowledge Bases , https://ceur-ws.org/Vol-2454/paper 60.pdf

[10] Norbert Braunschweiler and Rama Doddipatla and Simon Keizer and Svetlana Stoyanchev[2023], Evaluating Large Language Models for Document-grounded Response Generation in Information-Seeking Dialogues , https://arxiv.org

[11] Pinky Sitikhu,Kritish Pahi,Pujan Thapa,Subarna Shakya[2019] , A Comparison of Semantic Similarity Methods for Maximum Human Interpretability, https://arxiv.org/abs

[12] Adrian H. Raudaschl, ”Forget RAG, the Future is RAG-Fusion The Next Frontier of Search: Retrieval Augmented Generation meets Reciprocal Rank Fusion and Generated Queries”, 2023, https://towardsdatascience.com/forget-rag-the-future-is-rag-fusion-1147298d8ad1