PARSHWANATH CHARITABLE TRUST'S

A.P. SHAH INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering

Data Science

LUMIS:

LLM BASED UNIFIED MULTIMODEL INTELLIGENT SYSTEM

Group No. 9

Dalbirsingh Matharu	**21107005**
Umesh Pawar	**21107014**
Shreyas Patil	**21107061**
Vedant Parulekar	**21107034**

Project Guide

Ms. Sarla Mary

Ms. Richa Singh

Contents

- Abstract
- Introduction
- Objectives
- Literature Review
- Research Gap
- Problem Definition
- Scope
- Technological Stack
- Proposed System Architecture/Working
- Prototype Design Demonstration
- Implementation Status
- Review Suggestions (Given in Last meeting)

Abstract

Project **LUMIS** is an advanced LLM-based multimodal system designed to integrate diverse capabilities into a unified platform. It offers text generation, summarization, YouTube and website transcription, image-to-text conversion, and file retrieval through APIs like OpenAI and Gemini. Powered by the LangChain framework and CrewAI, LUMIS features intelligent AI agents that learn from user interactions and automate tasks. The system includes a real-time code assistant for code comprehension, error detection, and suggestion generation, seamlessly integrating with code editors. With a focus on user-friendliness. It uses GenAI capabilities for automated SQL generation and data extraction, and leverages OpenCV with Air Canvas technology to perform real-time calculations based on finger-drawn annotations. It employes a dynamic graphical user interface, Lumis enhances efficiency and accuracy, serving as a versatile tool for developers, professionals, and content creators.

Keywords: LUMIS, LLM, API, OpenAI, Gemini, Langchain, CrewAI, Agents, GenAI, OpenCV, Air Canvas.

Abstract

Real time problem or challenges.

- **Real-Time Interactive Code Debugging**: Integrating video and audio inputs to detect and resolve code errors instantly, enabling users to interactively pinpoint issues with a mouse and receive immediate, relevant solutions.
- **Advanced Content Processing**: Combining text generation, RAG, and transcription into a single system for efficient content creation and data extraction.
- **Automated SQL Generation and Extraction**: Implementing RAG functionality directly with MySQL to automate SQL queries and streamline data extraction for efficient schema management.
- **Real-Time Visual Problem-Solving**: Utilizing OpenCV and Air Canvas to interpret and address user-drawn problems in real time.

Introduction

- **Real-Time Observation of the Problem Domain:**
  Developers today struggle with inefficient debugging due to traditional tools lacking real-time feedback and integration. Many solutions focus on separate aspects like code analysis or text generation but do not combine these functions. This results in fragmented workflows and ineffective real-time support, highlighting the need for a unified platform that integrates video, audio, and text.
- **Motivation:**
  Lumis was inspired by the limitations of existing tools that fail to offer seamless integration of video, audio, and real-time code analysis. Developers face slow error detection and inadequate feedback with current solutions. Recognizing this gap, Lumis aims to provide a comprehensive, efficient tool that enhances productivity and user experience through cohesive, advanced AI capabilities.

Objectives

- To develop a multimodal system that detects and resolves code errors in real-time using video and audio input, with interactive error pinpointing via mouse.
- To integrate advanced functionalities such as text generation, retrieval-augmented generation (RAG), and transcription into a unified platform.
- To enable automated SQL generation and data extraction using GenAI, and leverage OpenCV with Air Canvas technology for real-time problem-solving based on user-drawn annotations.
- To ensure seamless integration of technologies and continuous updates for maintaining advanced functionality, usability, and relevance.

Literature Review

Sr.no	Title	Author(s)	Year	Methodology	Drawback
1	Retrival Augmented Genration for Knowledge Intensive NLP task	Patrick Lewis, Ethan Perez,Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal,Heinrich Kuttler,Mike Lewis, Wen tau Rocktaschel, Sebastian Riedel, Douwe Kiela	2020	The methodology combines seq2seq models with Dense Passage Retrieval (DPR) to enhance language generation. It uses RAG-Token for autoregressive beam search and RAG-Sequence with thorough or fast decoding for efficient text generation.	The RAG model relies heavily on external knowledge (Wikipedia), which limits scope and incurs high computational costs for retrieval. It reduces but doesn't eliminate hallucinations, lacks full interpretability, and struggles with real-time updates or multimodal inputs.
2	LLM potentiality and awareness: a position paper from the perspect ive of trustworthy and responsible AI modeling	Iqbal H. Sarker	2024	The methodology includes defining tasks and acquiring data, selecting and fine-tuning models, and evaluating their performance with metrics. Post-evaluation, models are deployed and monitored to ensure reliability	The paper highlights LLMs' susceptibility to generating misinformation, biases, and adversarial attacks, while also being opaque in decision-making and raising privacy and ethical concerns. These issues challenge trust, fairness, and responsible deployment in AI systems.
3	A Complete Survey on LLM-based AI Chatbots	Sumit Kumar Dam, Choong Seon Hong , Yu Qiao, Chaoning Zhang	2024	The paper surveys the evolution, applications, and challenges of LLM-based chatbots, from early models to current systems like ChatGPT. It aims to provide a comprehensive overview and explore future improvements for these technologies	This paper does not adequately address specific challenges faced by LLM-based chatbots, such as ethical concerns and data biases, nor does it discuss technical limitations like scalability and complex conversation handling. It lacks concrete examples or solutions related to the misuse of generated knowledge.

Literature Review

Sr.no	Title	Author(s)	Year	Methodology	Drawback
4	Mini-Gemini: Mining the Potential of Multi-modality Vision Language Models	Yanwei Li1∗ Yuechen Zhang1∗ Chengyao Wang1∗ Zhisheng Zhong1 Yixin Chen1 Ruihang Chu1 Shaoteng Liu1 Jiaya Jia1,	2024	The methodology of Mini-Gemini focuses on improving Vision Language Models (VLMs) by enhancing visual tokens through an additional visual encoder for high-resolution refinement, constructing a high-quality dataset that supports precise image comprehension and reasoning-based generation, and leveraging VLM-guided generation techniques.	Mini-Gemini's emphasis on high-resolution visual tokens, complex architecture, and computational demands makes it overly resource-intensive and ill-suited for audio-centric projects like yours, where simpler and more focused frameworks such as Whisper and Sentence-BERT are more effective.
5	Using an LLM to Help With Code Understanding	Daye Nam, Andrew Macvean, Vincent Hellendoorn, Bogdan Vasilescu, Brad Myers	2024	A user study with 32 participants compared GILT to web search, analyzing task completion and code comprehension. Quantitative and qualitative data were collected.	No significant improvement in task completion time or understanding. Benefits varied between students and professionals.
6	Creating Large Language Model Applications Utilizing LangChain: A Primer on Developing LLM Apps Fast	Oguzhan Topsakal , and Tahir Cetin Akinci	2023	LangChain, a framework designed to build AI applications utilizing large language models (LLMs). It uses modular components like prompts, chains, and memory to develop customizable pipelines that interact with external data sources for tasks like question-answering and	LangChain's reliance on LLMs means it may still face common issues like generating inaccurate outputs. Additionally, handling complex tasks can require multiple API calls, which may lead to slower execution or increased computational

Literature Review

Sr.no	Title	Author(s)	Year	Methodology	Drawback
7	Gemini in Reasoning: Unveiling Commonsense in Multimodal Large Language Models	Yuqing Wang, Yun Zhao	2023	Four LLMs and two MLLMs are tested on 12 commonsense reasoning datasets using zero-shot and few-shot learning to assess accuracy, with focus on visual-textual integration	The study is limited to English datasets, missing cross-cultural nuances, and results may vary with future model updates or new datasets.
8	Vision language models are blind	Pooyan Rahmanzadehgervi, Logan Bolton , Mohammad Reza Taesiri, Anh Totti Nguyen	2024	The BlindTest benchmark evaluates four top Vision Language Models (VLMs) on seven low-level visual tasks, such as counting shapes and identifying intersections, focusing on simple geometric challenges.	VLMs perform poorly on basic visual tasks, with an average accuracy of 58.57%, highlighting limitations in processing simple visual information and the inadequacy of current benchmarks in capturing these shortcomings.

Research Gap (Limitations of existing systems)

Survey Analysis:

- No existing system integrates multimodal LLM capabilities (video/audio input, mouse-based pinpointing) for real-time error detection.
- Lack of projects combining OpenCV for user-drawn problem solving or MySQL with RAG functionality.
- Current AI assistants lack dynamic interaction and intelligent problem-solving in integrated environments.

Problem Status:

- The problem remains unsolved; no system offers the full suite of multimodal interaction, AI assistance, and real-time code correction which LUMIS aims to deliver.

Observations and Improvement Possibilities:

- Multimodal Integration: Strong potential, but integration of video, audio and user interactions can be improved.

Research Gap (Limitations of existing systems)

Technologies Performing Well:

- **OpenCV**: OpenCV and Air Canvas for solving real-time, user-drawn problems on-screen.
- **MySQL RAG**: Powerful for structured database queries and retrieval-augmented generation

Confidence in Solution:

- LUMIS can bridge the gap by integrating these technologies effectively, pushing boundaries in real-time code analysis and multimodal interaction.

Problem Definition

- Lumis addresses the challenge of real-time code error detection and resolution by integrating video, audio, and text inputs from a user's screen. It aims to provide immediate, context-aware feedback and interactive error pinpointing for developers. Additionally, Lumis seeks to unify advanced AI functionalities—such as text generation, retrieval-augmented generation (RAG), and transcription—into a single platform. The system focuses on reducing latency, enhancing precision, and offering a user-friendly interface to streamline real-time interaction and improve overall productivity

Scope

- **Integrated Error Detection**: Develop a system that combines video, audio, and text inputs for instant identification and resolution of coding errors.
- **Interactive Debugging Tools**: Create functionalities that enable users to actively pinpoint and correct errors on-screen with immediate feedback.
- **Unified AI Features**: Merge advanced AI capabilities, including text generation, retrieval-augmented generation (RAG), and transcription, into a single, efficient platform.
- **Advanced User Interface and Automation**: Design a dynamic, user-centric interface and automate SQL generation and data management for seamless interaction and enhanced productivity.

Technology Stack

Models:

- Gemini 1.5 Pro
- Gemini 1.5 Flash
- Gemini text-embedding-004
- OpenAI GPT 3.5 Turbo
- OpenAI text-embedding-3-large

Libraries:

- opencv-python
- Pyautogui
- langchain_google_genai
- google-generativeai
- SpeechRecognition
- load_dotenv
- crewai

Database:

- MySQL
- CassandraDB

Frontend:

- Django
- Django Rest Framework

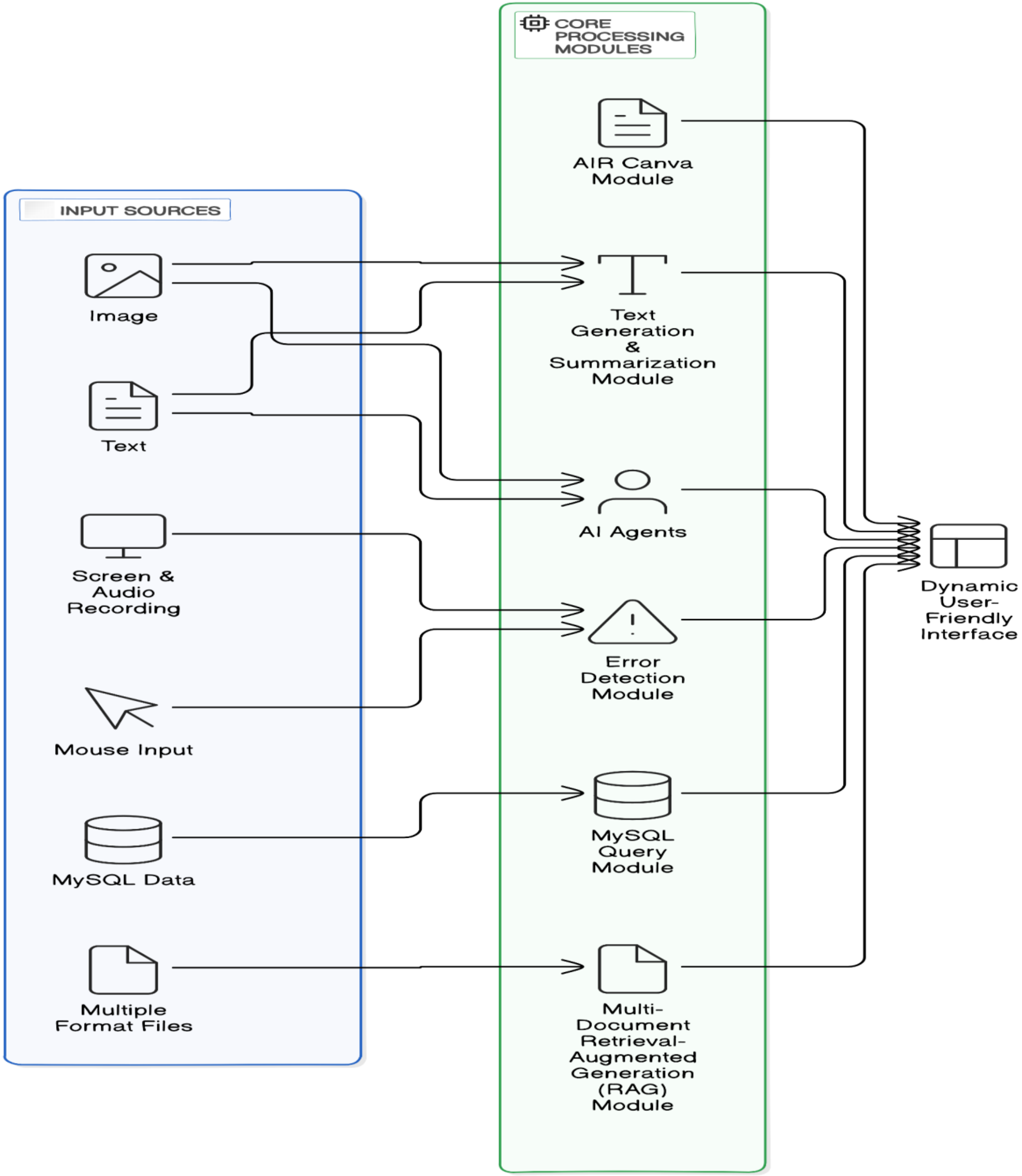
Framework:

- LangChain
- LangSmith
- CrewAI

Technology Stack

- **Models**: Utilizes Gemini and OpenAI models for advanced AI capabilities, text generation, and semantic understanding.
- **Libraries**: Incorporates OpenCV, PyAutoGUI, LangChain, and others for computer vision, automation, generative AI, and audio processing.
- **Database**: Uses MySQL for structured data and CassandraDB for scalable data management.
- **Frontend**: Built on Django and Django Rest Framework for web development and API creation.
- **Frameworks**: Employs LangChain, LangSmith, and CrewAI for managing AI workflows and enhancing application intelligence.

LUMIS Architecture



Proposed system architecture/Working

LUMIS Architecture
The LUMIS architecture integrates diverse input sources into specialized processing modules for effective code analysis and user interaction:

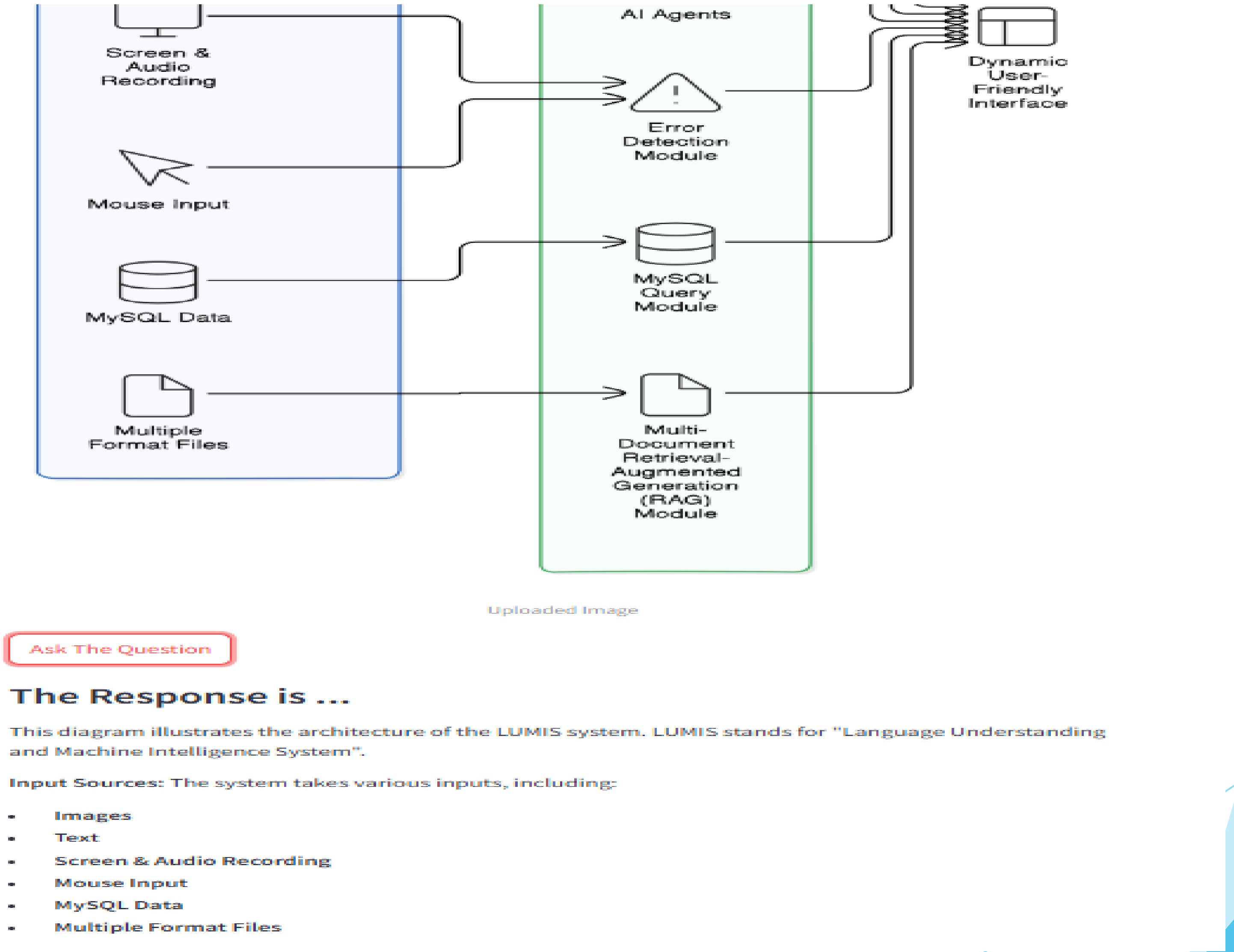
Input Sources:

- **Screen & Audio, Mouse Input**: Connected to the Error Detection Module for real-time coding error identification.
- **Web/YT Video Links**: Linked to the RAG & Transcription Module for information retrieval and audio transcription.
- **Text, Image**: Directed to the Text Generation & Summarization Module and AI Agents for content generation and summarization.
- **MySQL Data**: Connected to the MySQL Query Module for executing database queries.
- **Multiple Format Files**: Processed by the Multi-Document RAG Module for comprehensive information retrieval.

Core Processing Modules: Each module specializes in handling its respective input for optimal results.

Dynamic User-Friendly Interface: Integrates all functionalities into a seamless interface, enhancing user experience and task management.

Prototype Design Demonstration



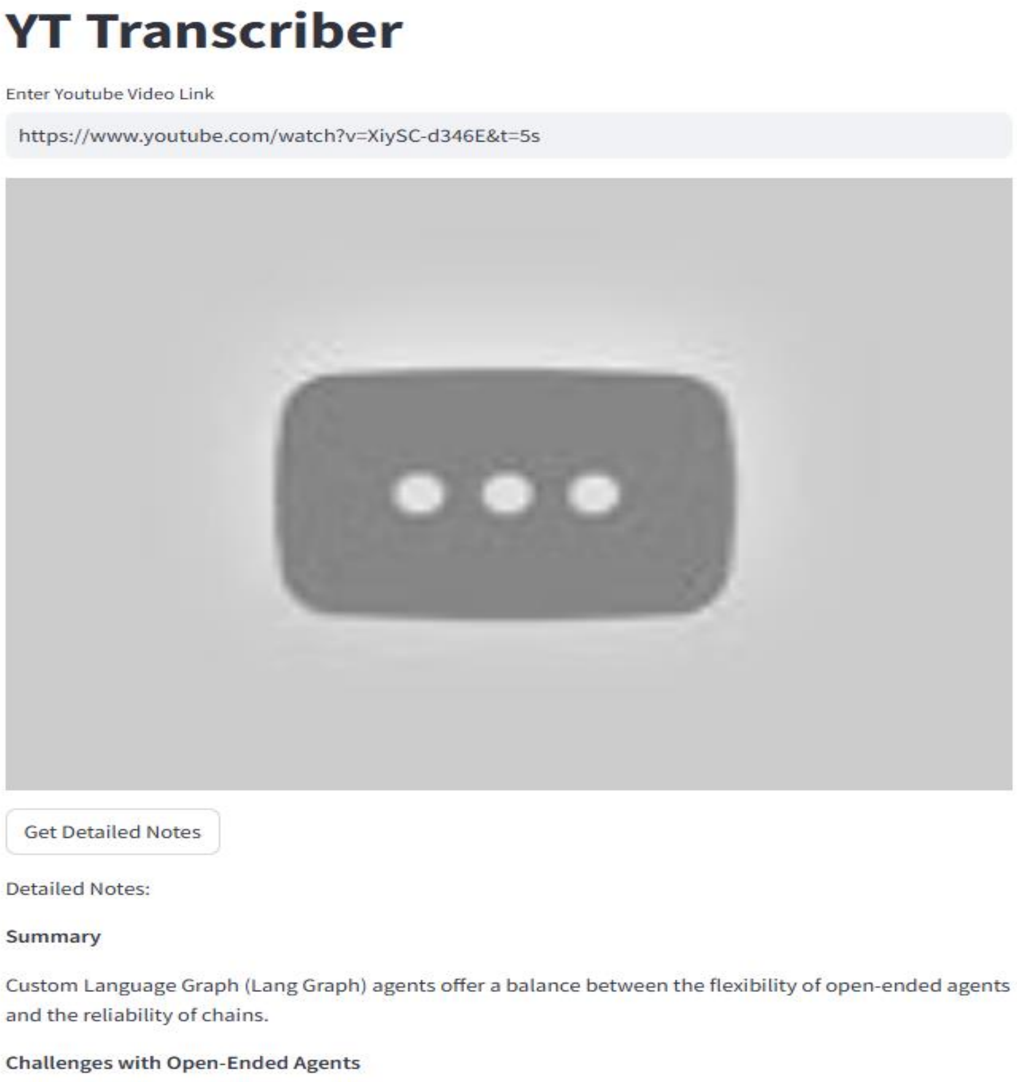
The Response is ...

This diagram illustrates the architecture of the LUMIS system. LUMIS stands for "Language Understanding and Machine Intelligence System".

Input Sources: The system takes various inputs, including:

- **Images**
- **Text**
- **Screen & Audio Recording**
- **Mouse Input**
- **MySQL Data**
- **Multiple Format Files**

Prototype Design Demonstration



YT Transcriber

Enter Youtube Video Link

https://www.youtube.com/watch?v=XiySC-d346E&t=5s

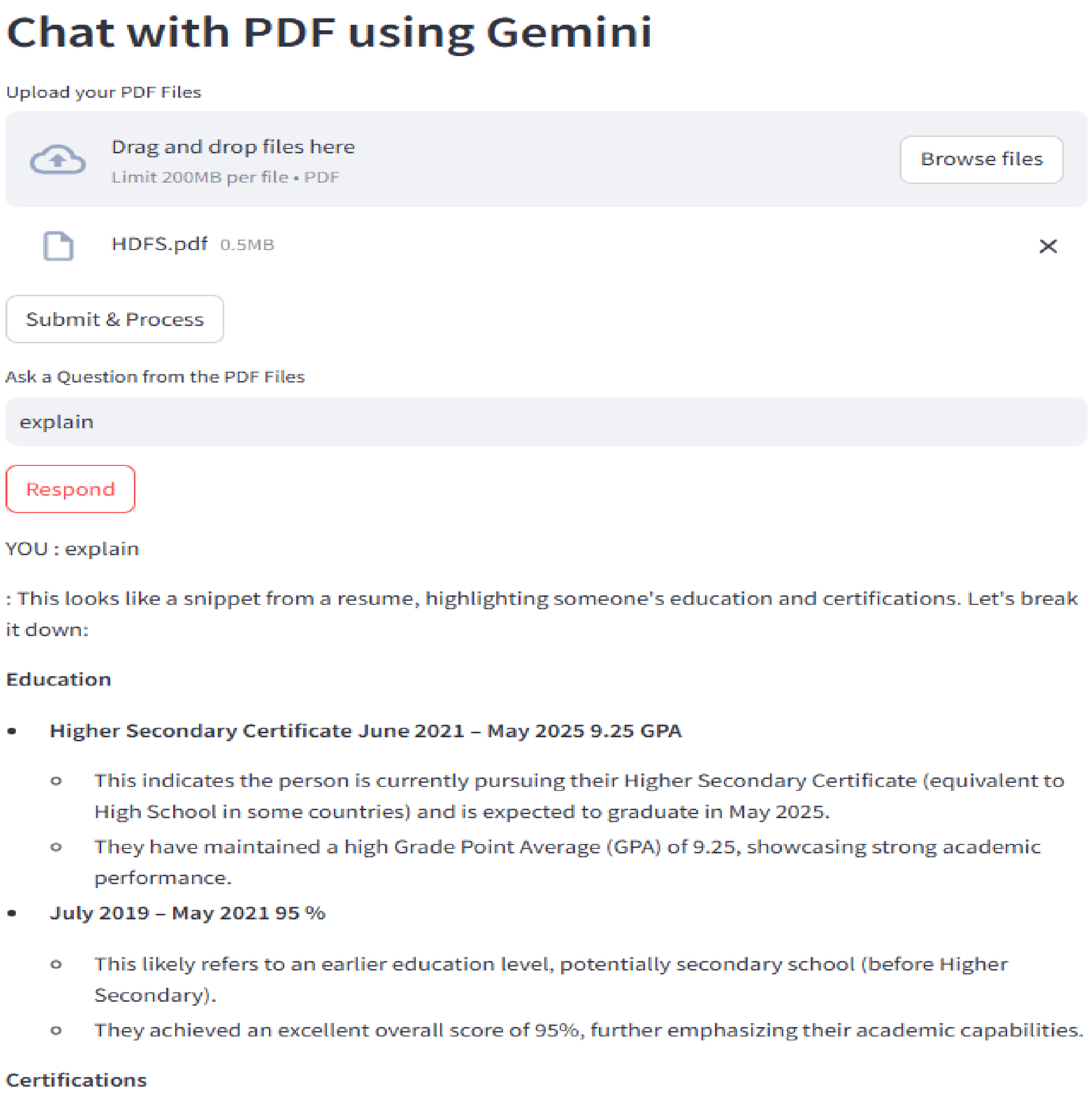
Detailed Notes:

Summary

Custom Language Graph (Lang Graph) agents offer a balance between the flexibility of open-ended agents and the reliability of chains.

Challenges with Open-Ended Agents

Prototype Design Demonstration



Chat with PDF using Gemini

Upload your PDF Files

HDFS.pdf 0.5MB

Ask a Question from the PDF Files

explain

YOU : explain

: This looks like a snippet from a resume, highlighting someone's education and certifications. Let's break it down:

Education

- **Higher Secondary Certificate June 2021 – May 2025 9.25 GPA**
  - This indicates the person is currently pursuing their Higher Secondary Certificate (equivalent to High School in some countries) and is expected to graduate in May 2025.
  - They have maintained a high Grade Point Average (GPA) of 9.25, showcasing strong academic performance.
- **July 2019 – May 2021 95 %**
  - This likely refers to an earlier education level, potentially secondary school (before Higher Secondary).
  - They achieved an excellent overall score of 95%, further emphasizing their academic capabilities.

Certifications

Implementation Status

- **Image to Text**: Completed. Users can upload an image and query information extracted from it
- **YouTube Transcription**: Completed. Users can input a YouTube link and receive detailed video transcription.
- **Multiple PDFs RAG**: Completed. Retrieval-augmented generation is implemented for querying multiple PDF documents.

References

- **LangChain,** https://python.langchain.com/v0.2/docs/introduction/
- **LangSmith,** https://docs.smith.langchain.com/
- **OpenAI,** https://platform.openai.com/docs/models
- **Gemini,** https://ai.google.dev/gemini-api/docs/models/gemini
- **Gemini Vision,** https://ai.google.dev/gemini-api/docs/vision?lang=python
- **Gemini Audio,** https://ai.google.dev/gemini-api/docs/audio?lang=python
- **Crew AI,** https://docs.crewai.com/

References

- Pooyan Rahmanzadehgervi, Logan Bolton, Mohammad Reza Taesiri, Anh Totti Nguyen, **Vision language models are blind**, Auburn University, AL, USA University of Alberta, Canada, 67, **26 July 2024, https://arxiv.org/pdf/2407.06581**
- Sumit Kumar Dam, Choong Seon Hong, Fellow, IEEE, Yu Qiao, Student Member, IEEE, and Chaoning Zhang, **A Complete Survey on LLM-based AI Chatbots,23, 17 July 2024, https://arxiv.org/abs/2406.16937**
- Daye Nam,Andrew Macvean, Vincent Hellendoorn, Bogdan Vasilescu, Brad Myers,**Using an LLM to Help With Code Understanding**,13, **April 2024**,https://dl.acm.org/doi/abs/10.1145/3597503.3639187
- Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, Jiaya Jia, **Mini-Gemini: Mining the Potential of Multi-modality Vision Language Models,** The Chinese University of Hong Kong,19,**27 March 2024**, https://arxiv.org/pdf/2403.18814
- Iqbal H. Sarker,**LLM potentiality and awareness: a position paper from the perspective of trustworthy and responsible AI modeling**,7,1 March 2024

References

- Yuqing Wang, Yun Zhao, **Gemini in Reasoning: Unveiling Commonsense in Multimodal Large Language Models**, Stanford University ywang216@stanford.edu,Meta Platforms, Inc.yunzhao20@meta.com, 13**,29 December 2023 , https://arxiv.org/pdf/2312.17661**
- Oguzhan Topsakal,T. Cetin Akinc, **Creating Large Language Model Applications Utilizing LangChain: A Primer on Developing LLM Apps Fast , 8 , July 2023** https://www.researchgate.net/publication/372669736_Creating_Large_Language_Model_Applications_Utilizing_LangChain_A_Primer_on_Developing_LLM_Apps_Fast
- Patrick Lewis, Ethan Perez,Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal,Heinrich Kuttler,Mike Lewis, Wen tau Rocktaschel, Sebastian Riedel, Douwe Kiela, **Retrival Augmented Genration for Knowledge Intensive NLP task,** 16**, 2020, https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html**

Thank You...!!