2024 Second International Conference on Emerging Trends in Information Technology and Engineering (ICETITE) | 979-8-3503-2820-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/IC-ETITE58242.2024.10493791

Advanced Video Transcription And Summarization A Synergy of Langchain, Language Models, And VectorDB with Mozilla Deep Speech

K Lavanya
Vellore Institute of Technology-vellore
Vellore, India
lavanya.k@vit.ac.in

Aravind K
Vellore Institute of Technology-vellore
Vellore, India
aravind.karunakaran2023@vitstudent.ac.in

Vishal Dixit
Vellore Institute of Technology-vellore
Vellore, India
vishal.dixit2023@vitstudent.ac.in

Arockia Sachin A
Vellore Institute of Technology-vellore
Vellore, India
arockiasachin.a2023@vitstudent.ac.in

***Abstract*—This paper introduces an advanced automated information management system, addressing the critical need for effective transcription and summarization in the face of the burgeoning volume of video data. The study's objective is to overcome the limitations of existing methods, primarily in handling vast video transcriptions and enhancing information retrieval. We propose an innovative integration of Mozilla Deep Speech, LangChain, VectorDB, and Large Language Models (LLMs), aiming to significantly improve the efficiency and accuracy of document-oriented processes in various sectors. Our approach leverages generative AI to transform the way organizations process video content, offering a solution to the lengthy and often challenging transcription tasks that current technologies struggle with. We highlight the potential economic impact, as underscored by a McKinsey study, indicating a substantial contribution of these technologies to various industries, particularly in customer service, sales, marketing, and software development. The methodology encompasses a comprehensive system architecture, utilizing NLP techniques and models like BERT, GPT, and spaCy, addressing the shortcomings of existing approaches with enhanced precision and adaptability. Initial findings demonstrate a transformative improvement in video summarization, providing significant benefits in education, journalism, and accessibility for the hearing impaired. This research not only offers a novel solution to existing challenges in data management but also sets a new standard for the application of generative AI and LLMs in automated information processing.**

***Keywords*—Document-Oriented Agents, Workflow Transformation, Video Transcription Summarization, LangChain, Large Language Models, VectorDB, Mozilla Deep Speech, Context-Aware Summarization, Audio-to-Text Translation, Video Analysis, Audio Content Transcription**

I. INTRODUCTION

Learning from a growing pool of data has become harder in a period of constant information increase. Effective transcription and summarization algorithms are needed due to the popularity of video footage in many fields. Document-oriented agents are common corporate solutions that improve internal documentation productivity. As we examine automated information management, McKinsey study emphasizes the potential impact of generative AI on the global economy. According to this report, these technologies might contribute $2.6 trillion to $4.4 trillion annually to the economy. About 70% of current employee responsibilities may be automated with this connection. The study identifies customer service, sales and marketing, and software development as major beneficiaries of this disruptive wave. Despite the promise of document-oriented agents and generative AI, technological challenges persist. A major issue is managing large video transcriptions. Transcripts are sometimes lengthy and difficult to read and extract information from. Even the most advanced Large Language Models (LLMs), which have a token maximum of 100,000 tokens, struggle to handle massive volumes of data for critical business functions like customer service. Sporadic model output mistakes remain an issue. LLMs are proficient at understanding and producing natural language, but they may make mistakes, which can have serious consequences in professional settings where accuracy is crucial. Thus, a thorough and precise method to compress video-transcribed text is needed. The proposed method improves information retrieval, altering how organizations use video content. The project is well-planned to address the challenge. In the following sections, we will discuss our research's unique methods and technologies, with a focus on Mozilla Deep Speech. This study examines LangChain, Language Model, and VectorDB's advanced features in cooperation. The goal is to create a document-processing agent that can efficiently transcribe video of various lengths. We will also discuss the benefits of this unique mix, including enhanced efficiency, accuracy, flexibility, and expandability. The project's initial findings will demonstrate its ability to transform video summarization. This breakthrough improves education, journalism, and hearing accessibility. In addition to Mozilla Deep Speech, Microsoft Azure Speech Service, IBM Watson Speech to Text, and Amazon Transcribe has the capability to transcribe video audio into textual format. In conjunction with Large Language Models frameworks such as Auto-GPT, LlamaIndex, Simpleaichat, Outlines , MetaGPT, AutoChain, the use of other NLP techniques and libraries beside OpenAI such as BERT, GPT, and spaCy enables the examination and condensation of textual content.

II. LITERATURE REVIEW

The Joseph Benjamin Ilagan et al.,utilizes natural language processing techniques and GPT-3.5, a large

language model, to design and develop a chatbot for evaluating and refining student startup ideas . The chatbot accepts descriptions of startup businesses from student co-founders through a Telegram chatbot and formats them as prompts to be fed into GPT-3.5 .GPT-3.5 generates responses based on prompts instructing the bot to provide feedback from three virtual panelists: a harsh judge, a neutral expert, and an optimistic investor [1]. Oguzhan Topsakal et AL., focuses on the utilization of Large Language Models (LLMs) for the rapid development of applications, with a spotlight on LangChain, an open-source software library.LangChain is a framework for developing applications utilizing large language models. It provides components (modular abstractions) and chains (customizable use case-specific pipelines). The study examines LangChain's core features, including its components and chains, which act as modular abstractions and customizable, use-case-specific pipelines, respectively. The paper provides practical examples to illustrate the potential of LangChain in fostering the swift development of LLM-based applications [2]. Arjun Pesaru et al., introduces the use of LangChain and the LLM Model to create a PDF chatbot, highlighting the benefits of using AI chatbots for document management and interaction with users .It mentions that PDFs are widely used in business, education, and research, but can be difficult to read and understand, making chatbots a valuable tool for providing information and answering questions about PDF files .The project also utilizes Pinecone to store the vectors of the PDF files, enabling efficient retrieval of related documents .React JS is used for the front end development of a webpage to interact with the chatbot, providing a user-friendly and efficient interface [3]. Le Chen et al.., introduce LM4HPC, a comprehensive framework that encapsulates a suite of machine learning components within user-friendly APIs. This framework is tailored for HPC users, simplifying the implementation process and making the robust capabilities of language models more accessible and user-friendly within the HPC community. The primary goal of LM4HPC is to reduce the complexities inherent in employing language models, thus enabling HPC users to leverage their powerful capabilities more effectively and efficiently [4]. Tristan Vanderbruggen et at., proposes the formalization of the execution model of language models (LMs) and introduces a new algorithm for sampling the predictions of LMs .The authors also introduce a low-level language to write "cognitive programs" for the execution model of LMs .The implementation of the proposed execution model utilizes the LLaMa.cpp and HuggingFace wrappers for LM .The paper mentions the use of the choice algorithm in the execution model, particularly for branches in the push-down automaton (PDA) [5]. Rodrigo Pedro et al., conducted a comprehensive examination of prompt-to-SQL (P2SQL) injections targeting web applications based on the Langchain framework.The authors characterized P2SQL injections and explored their variants and impact on application security through multiple concrete examples.The paper evaluated 7 state-of-the-art LLMs to demonstrate the pervasiveness of P2SQL attacks across language models.The authors proposed four effective defense techniques that can be integrated as extensions to the Langchain framework to counter P2SQL attacks.The defenses were validated through an experimental evaluation with a real-world use case application [6]. Vrishani Shah et al.,Presents the MIT-GPT project uses a client-server architecture, with a web-based interface for users to input queries and a server that processes the queries using the OpenAI API and LangChain. The processing system in this architecture is best suited for the programming language LangChain, which was specifically created for natural language processing systems. Clients send user queries to the server, which then processes them using the OpenAI API and LangChain, and sends the answers back to the client in text-based responses. The system is responsible for processing the queries, sending them to the OpenAI API for language processing, and using LangChain to generate responses based on the college-specific dataset [7]. Dillon Cleary et al., created a Large Language Model (LLM) and integrated LangChain, linking it to my Google Drive enabling the model to access the necessary data to be trained on. The model was trained on various PDFs of 10-K forms from BP, Exxon, Chevron, and other publicly traded oil and gas companies. Additionally, we added research reports to the training data that elaborated on the energy transition and its possible impacts on large oil companies. A significant research report implemented into the model was one analyzing OPEC (Organization of the Petroleum Exporting Countries) [8]. Sebastian Lobentanzer et al., discusses the challenges in understanding and contextualizing scientific results in the field of biomedical science, due to the exponential growth of knowledge in this field. It highlights the limitations of current Large Language Models (LLMs) in biomedical analyses, such as a lack of general awareness and logical deficits .The authors propose a conversational platform called biochatter, exemplified in the web application ChatGSE, which combines human ingenuity and machine memory to improve biomedical analyses. They aim to make LLMs more useful and trustworthy in research applications by integrating popular bioinformatics methods and implementing measures to safeguard against LLM shortcomings . The paper also mentions the use of prompt engineering and a second model to ensure factual correctness of the LLM's responses. It discusses the development of biochatter as a platform for communicating with LLMs specifically tuned to biomedical research, and the automation of popular bioinformatics methods [9]. Wangchunshu Zhou et al., introduces AGENTS, an open-source framework for autonomous language agents, which is carefully engineered to support important features including planning, memory, tool usage, multi-agent communication, and fine-grained symbolic control. AGENTS supports human-agent interaction in multi-agent systems, allowing human users to play the role of an agent and interact with other language agents in the environment [10]. Andrew Zhu et al., presents Kani, a lightweight and highly hackable framework for building language model applications. Kani supports the core building blocks of chat interaction, including model interfacing, chat management, and robust function calling .Kani allows developers to customize functionality for their own needs by providing easily overridable and well-documented core functions. The paper demonstrates how Kani's function calling can be used to retrieve information from a data source like Wikipedia [11]. Fadel M. Megahed et al., introduces ChatSQC, a chatbot system that combines OpenAI's Large Language Models (LLMs) with a specific knowledge base in Statistical Quality Control (SQC).The information generated by ChatSQC was evaluated through a comparative study with two benchmark methods: a customized GPT-3.5 and GPT-4 .The evaluation approach involved designing prompts to capture different subfields of SQC, generating responses from each platform (ChatSQC, GPT-3.5, and GPT-4), and having expert raters rate the

responses on a Likert scale.The responses were blinded and randomized to prevent bias, and four authors who are experts in SQC rated each response [12]. Jianing Yang et al., utilizes LLM-Grounder, which is an approach for 3D vision-language tasks in robotics, using Large Language Models (LLMs) as an agent.The agent, in this case, is GPT-4, which is prompted to complete three tasks: breaking down complex text queries into sub-tasks, orchestrating and using downstream tools like a CLIP-based 3D visual grounder, and making grounding decisions based on spatial understanding and common sense [13]. Sujatha Rajkumar et al.'s study on "Detection of Post COVID-Pneumonia Using Histogram Equalization, CLAHE Deep Learning Techniques" marks a significant advancement in the field of medical diagnostics using artificial intelligence. Focusing on the detection of post-COVID pneumonia, the research integrates advanced image processing techniques, specifically Histogram Equalization (HE) and Contrast-Limited Adaptive Histogram Equalization (CLAHE), with Convolutional Neural Networks (CNNs), including the VGG16 model. This approach highlights the effectiveness of combining deep learning with enhanced image processing to analyze chest X-rays for pneumonia detection. The study's utilization of a large dataset of 6432 chest X-rays and achieving a high accuracy rate underscores the importance of data-driven methodologies in improving diagnostic accuracy in medical imaging. Additionally, the research opens avenues for future exploration in applying similar techniques to other areas in medical diagnostics, demonstrating the potential of AI in addressing complex healthcare challenges, especially in the context of the COVID-19 pandemic. [14]. Dr. G. Kiruthiga and Dr. et al explore enhancing video surveillance through Deep Convolutional Neural Networks (CNN) coupled with Probabilistic Neural Networks (PNN). This study addresses the intricate task of object detection in dynamic video environments, crucial for applications like accident prevention and security surveillance. The innovative approach of the research lies in its multi-scale deformable CNN model, which adeptly handles geometric deformations of objects and incorporates multi-scale feature maps for effective feature fusion. The incorporation of PNN for CNN optimization is a pivotal aspect of the study, improving the accuracy of the detection process. Through comparative analysis with existing models, the CNN-PNN model showcases superior performance, particularly in accuracy and error reduction, making it a significant advancement in the field of computer vision and AI-driven surveillance technology [15]. Hank Liao et al address the challenge of transcribing diverse YouTube content with high accuracy. They focus on enhancing automatic speech recognition (ASR) by using deep neural networks (DNNs) and a vast inventory of context-dependent states. The study stands out for its innovative use of semi-supervised training data derived from user-uploaded video transcripts. By applying an "islands of confidence" filtering heuristic for selecting training segments and expanding the model to include 44,526 context-dependent states with a low-rank final layer weight matrix approximation, they achieve a significant performance improvement. This approach results in a 13% relative improvement over previous DNN models, marking a substantial advancement in ASR technology for efficiently handling the extensive and varied content on YouTube [16]. The table 1 summarizes four papers, discussing the strengths and limitations of a GPT-3.5-based chatbot for student startups, an AI document manager using LangChain and Pinecone, a deep neural network model for YouTube video transcription, and LangChain for LLM application development, emphasizing advancements in AI-driven educational tools and data management alongside considerations for user input dependency and system complexity.

Table 1 Advantages and drawbacks of the existing techniques compared

Paper Name	*Pros*	*Cons*
A Prototype of a Chatbot for Evaluating and Refining Student Startup Ideas Using a Large Language Model	•Uses GPT-3.5 to simulate feedback enhancing entrepreneurial education. • Provides cost-effective solution for refining startup ideas interactively	•Reliance on user input impacts feedback accuracy. •Requires frequent updates to adapt to evolving business concepts.
AI Assistant for DocumentManagement Using Lang Chain and Pinecone	•LangChain integrated with LLM for highly functional PDF chatbot. •Uses Pinecone for efficient PDF information processing and retrieval.	• System complexity may require substantial computational resources. •Continuous LLM updates needed for response accuracy maintenance
Large Scale Deep Neural Network Acoustic Modeling with Semi-supervised Training Data for Youtube Video Transcription	•Large-scale acoustic modeling for YouTube transcription with semi-supervised data. •Island of Confidence heuristic enhances high-quality data selection.	•Semi-supervised approach may need refinement for diverse YouTube content. •Risk of overfitting to domain-specific features in training dataset.
Creating Large Language Model Applications Utilizing LangChain: A Primer on Developing LLM Apps Fast	•Discusses LangChain for rapid LLM development with practical examples. •Integrates LLMs with data sources for enhanced AI app development.	•May not extensively cover LangChain's challenges and limitations. •Lacks detailed comparison with other frameworks and scalability discussion.

Fig. 1. Table 1 Advantages and drawbacks of the existing techniques compared

III. BACKGROUND STUDY

A. Mozilla's DeepSpeech

DeepSpeech is an open-source speech-to-text engine that utilizes a deep learning model based on a recurrent neural network (RNN) to transform spoken language into text. Its RNN architecture, designed to handle sequential input, is particularly well-suited for processing audio and is robust due to extensive training on a large corpus of voice data. This training allows DeepSpeech to understand a variety of speech patterns, accents, and languages with remarkable accuracy. It goes beyond simple word recognition; it can also grasp the context in which words are spoken, an intricate challenge in speech recognition. The engine's adaptability and efficiency stem from its ability to learn from diverse datasets, making it versatile for voice recognition applications. The RNN framework of DeepSpeech processes audio using the equation:

$$h_t = \sigma(w_{hh}h_{t-1} + w_{xh}x_t + b_h) \quad (1).$$

where h_t represents the hidden state(1), Ws are weight matrices, $b_h$is a bias term, and σ is the activation function.is the activation function. DeepSpeech has been trained on extensive speech data, equipping it to discern diverse speech patterns, accents, and languages efficiently. Its ability to learn from a large dataset makes it highly adaptable, and its design allows it to understand not just the words but the context of speech, which is pivotal for nuanced speech recognition

B. Large Language Models (LLMs):

Language models like GPT-3 have revolutionized text generation, producing outputs nearly indistinguishable from those written by humans. They are built upon vast datasets and utilize a transformer architecture with an attention mechanism that supports understanding and maintaining context within a conversation or a document. This results in coherent and contextually appropriate responses. The attention mechanism is mathematically represented as:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \qquad (2)$$

Where Q, K, and V stand for query, key, and value, respectively, and □_k is the dimensionality of the key(2). Besides text generation, these models are incredibly useful for summarizing lengthy texts, which is particularly beneficial in sectors like law and healthcare where quick information processing is crucial. The evolving capabilities of LLMs continue to open new avenues for automation and efficiency in numerous fields

C. Vector Embeddings: Encoding Semantics:

Vector embeddings are pivotal in modern natural language processing (NLP), providing a high-dimensional vector representation of words and phrases that encapsulate semantic meanings. Initially, models like Word2Vec and GloVe presented static representations, where each word was mapped to a single vector, regardless of its context. However, advancements such as BERT introduced contextual embeddings, where a word's representation can dynamically change in response to its surrounding text. This leads to a more nuanced and precise capture of meaning. Such embeddings are integral to the functionality of a broad spectrum of NLP applications, including but not limited to semantic search, sentiment analysis, and machine translation. The optimization of the log probability function in Word2Vec, represented as:

$$\frac{1}{T}\Sigma_{t=1}^{T}\Sigma_{-c\le j\le c, j\neq 0}\log p\left(w_{t+j}|w_t\right) \qquad (3)$$

where words are given a representation based on their context(3). BERT and similar models have evolved this concept further with contextual embeddings, offering even more nuanced semantic representations. These embeddings are instrumental in various NLP applications, enhancing the capabilities of semantic search, sentiment analysis, and machine translation

D. Semantic Search

The evolution to semantic search engines, powered by vector embeddings, marks a significant improvement in how information retrieval systems align with user intent. These embeddings enable the systems to grasp the subtleties of natural language and the context of queries, thereby interpreting the semantic meaning rather than just matching keywords. Semantic search uses mathematical measures such as cosine similarity,

$$\text{Cosine\_similarity}(A,B) = \frac{A\cdot B}{\|A\|\|B\|} \qquad (4)$$

to determine the closeness of documents in a high-dimensional vector space(4). This approach ensures that the results are conceptually connected to the query. Such advancements reflect a more human-like understanding and retrieval of information, significantly enhancing user experience by providing more relevant search results, irrespective of the exact terms used in the query

E. Semantic Search

Vector databases manage the complex data structures produced by vector embeddings, and are integral to fields such as machine learning and data science. ChromaDB, for example, optimizes for high-dimensional data, applying ANN search algorithms to efficiently perform similarity searches based on the nearest neighbor principle

$$\text{argmin}_{x_i\in X}\|q - x_i\| \qquad (5)$$

This capability is essential for rapidly identifying similar items within a dataset, which is a cornerstone of recommendation systems and semantic search technologies

F. Real-Time Query Resolution with LLMs and LangChain

The activation of a question-answering chain within a real-time query resolution framework is facilitated by the load_qa_chain function. This function utilizes the analytical capabilities of Large Language Models (LLMs) to analyze user questions. This process is enhanced through the integration with the Chroma vector database, which facilitates the extraction of pertinent document fragments. The procedure is optimized—when a query is initiated, the system quickly detects and ranks document vectors that correspond with the user's question. It then utilizes the closest vector matches to generate an informative response. This system demonstrates proficiency in immediate implementation across several industries. Within the realm of customer service, the provision of immediate and precise responses is facilitated through the utilization of a pre-established database containing frequently asked questions (FAQs). From an educational standpoint, it enhances the speed at which material is accessed from a wide range of academic sources. The system's adaptability is similarly beneficial in specialized sectors such as the legal and healthcare industries, as it enables rapid analysis of intricate datasets to provide accurate information. The combination of LLMs with LangChain enables real-time processing, which

signifies a notable progress in the advancement of AI-powered search and response systems.

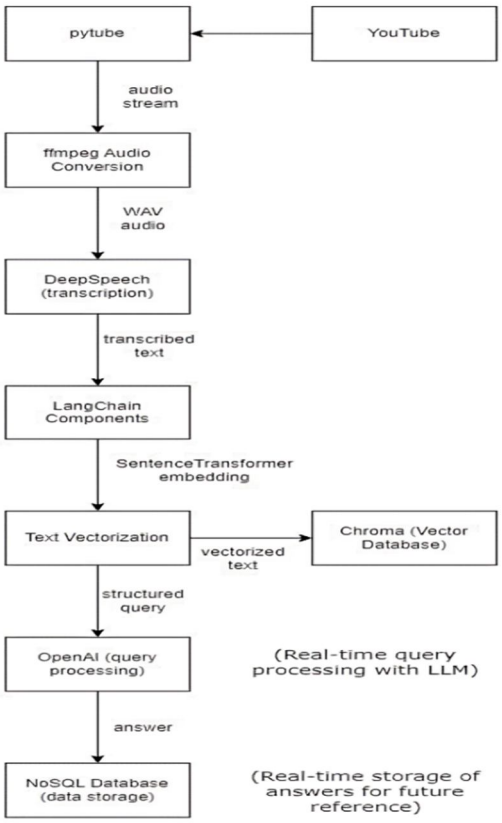

Fig. 2. Proposed System Architecture

IV. System Architecture

The methodology is supported by a system design that consists of a continuous pipeline. This pipeline initiates with the gathering of audio content from YouTube, utilizing the pytube library. The audio stream that is obtained is subsequently converted into WAV format using ffmpeg in order to ensure compatibility with the DeepSpeech model. This model is capable of transcribing the audio into text. The transcriptions undergo additional processing by LangChain components in order to build Sentence Transformer embeddings. The text has been transformed into a vector representation and subsequently saved in ChromaDB, a specialized database optimized for doing searches with great efficiency. The processing of structured queries is accomplished through the utilization of OpenAI's robust algorithms. These algorithms are responsible for constructing responses, which are then saved in a NoSQL database for the purpose of persistence and retrieval. The architecture shown below exemplifies the amalgamation of specialized tools and models, each making a distinct contribution to a resilient framework that possesses the ability to undertake intricate audio processing, comprehensive search capabilities, and the development of responsive outputs. Figure 4 depicts the process of extracting audio from YouTube, converting it to text, and processing the query to generate an answer using various technologies like pytube, ffmpeg, DeepSpeech, LangChain, and OpenAI's LLM, with the final answer being stored in a NoSQL database.

V. Methodology

The approach employed in this project has been carefully designed to effectively handle and improve the analysis of unstructured text. This process transforms the text into valuable insights using a system that not only accurately responds queries but also continuously improves its performance. At the core of this breakthrough is the utilization of artificial intelligence (AI) for conducting comprehensive searches, effectively combining data through advanced natural language processing techniques. The endeavor is motivated by a need for high relevance and precise accuracy within a dynamic data ecosystem. The system's indexing and query response capabilities demonstrate a high level of analytical proficiency, similar to that of topic modeling. These capabilities effectively extract significant patterns, enabling users to discover useful insights within extensive amounts of information. The framework exhibits a proactive nature, since it is not solely responsive but also possesses the ability to anticipate and adapt to emerging user needs. The aforementioned foresight highlights a commitment to continuous innovation, positioning the system to shape future discussions and empower a knowledgeable societal framework.

A. Phase 1: Environment Setup and Application Configuration

The first stage involves creating a Docker environment by building a container that includes the essential Python runtime and system packages, such as FFmpeg, which plays a crucial role in audio processing operations within the application. The implementation of environment encapsulation guarantees uniformity across development and production environments. During the process of building a Docker image, Docker utilizes its layer caching method to efficiently install Python dependencies that are specified in the requirements.txt file. This approach helps to optimize the overall build times and effectively utilize system resources. Simultaneously, the container's filesystem incorporates pre-trained models for DeepSpeech and LangChain. These models play a crucial role in facilitating the transcribing and deep search capabilities of the program. Once the environment and models have been established, the FastAPI service is setup and initiated using Uvicorn as the server. This configuration enables the development of API endpoints to ease user interactions, thereby preparing the application to receive and handle incoming requests.

B. Phase 2: Audio Retrieval and Conversion

After completing the setup of the environment, the application shifts its attention towards the management of multimedia content. The initial stage commences with the utilization of the pytube library within the application to retrieve audio streams from YouTube. These audio streams are thereafter downloaded directly to the local storage of the container. The process of localizing audio recordings is crucial in ensuring their accessibility throughout the upcoming transcribing phase. The audio files are subsequently transformed into a WAV format, utilizing

FFmpeg, in order to conform to the input specifications of the DeepSpeech model. The conversion of format is a crucial step in ensuring that the audio information is appropriately prepared for transcription by the speech-to-text engine. This process is necessary to optimize efficiency and accuracy, hence establishing the foundation for subsequent operational procedures inside the application

C. Phase 3: Speech-to-Text Transcription

During Phase 3, the application undergoes a transition wherein it proceeds to process the audio content that it has obtained. The initial stage involves configuring DeepSpeech, wherein the pre-existing DeepSpeech model is initialized within the application's context. The configuration of this system is adjusted in a calibrated manner to effectively transform voice data from audio files into written text, necessitating meticulous parameter modifications to guarantee the highest possible transcription accuracy. Upon completion of the configuration process, the Transcription Process is initiated. The WAV audio files, which have been appropriately formatted, are inputted into the DeepSpeech model. The model actively engages in the process of auditory perception and subsequently interprets the verbal utterances, so transforming them into a written form. The aforementioned procedure is characterized by a high computing workload and serves as a fundamental component of the application's operation. Its primary purpose is to convert auditory data into a format that can be easily manipulated and searched. The last stage in this phase involves the process of Text Normalization. The unprocessed output obtained from the transcription procedure frequently contains irregularities or distortions that lack utility for the future search capabilities of the application. Consequently, the transcribed text is subjected to a normalization procedure wherein errors are rectified and the content is uniformly formatted. The process of cleaning up is of utmost importance in ensuring the precise categorization of information, which in turn significantly influences the efficiency of subsequent deep search functionalities. The procedures in Phase 3 together encompass the process of converting audio input into a structured textual format. This transformation prepares the content for indexing and search, establishing a strong basis for the subsequent deep learning and natural language processing activities of the program.

D. Phase 4: Deep Search Indexing, Query Answering, and System Maintenance

During Phase 4, the application's deep search engine and user response system are optimized to work together effectively. This results in the delivery of accurate information that has been validated by users, while also ensuring the application's ongoing enhancement and dependability. The approach commences with the process of Indexing Preparation, whereby transcribed texts are transformed into vector embeddings that are optimized for search functionality through the utilization of advanced Natural Language Processing (NLP) models. The embeddings enhance ChromaDB, the foundational component of the application designed for comprehensive search capabilities. ChromaDB is regularly updated and optimized to provide efficient and precise query matching. The Query Interface is carefully designed to utilize the LangChain QA chain, enabling the precise interpretation of user requests. The configuration described facilitates the process of Answer Retrieval by efficiently traversing the indexed data, identifying and formulating the most relevant responses according to the user's input. With a focus on enhancing system resilience, Continuous Monitoring effectively manages the well-being of applications by actively monitoring their health, regularly updating dependencies, and optimizing AI models. This approach guarantees continuous performance and enables prompt adaptation to new data or user behavior patterns. The involvement of users is crucial during this phase, as systems for handling user requests are implemented to ensure continuous service. Answer Optimization is a dynamic sub-phase in which the application acquires knowledge from user feedback, thereby improving its responses to enhance both clarity and relevance. The process of regular database synchronization is essential for ensuring that the knowledge base of a system is up-to-date and accurately reflects the most recent information. This practice is crucial for preserving the integrity and value of user interactions. This phase encompasses a comprehensive approach that addresses not only the immediate needs of users but also anticipates future expectations, so ensuring that the system remains at the forefront of deep search technology and user support.

E. Pseudocode

```
Initialize FastAPI application
Define paths for:
- DeepSpeech model and scorer files
- Content, downloads, and transcripts directories
- ChromaDB directory
Load DeepSpeech model with the specified model and scorer files
Initialize LangChain components:
- SentenceTransformerEmbeddings
- ChatOpenAI with the OpenAI API key
Define startup event:
- Load documents from content directory
- Split documents into manageable chunks
- Index documents in ChromaDB and persist on disk
- Load the question-answering chain
Define download_and_transcribe endpoint:
- Accept a YouTube URL as input
- Download the audio stream of the YouTube video
- Convert the audio stream to a WAV file using ffmpeg
- Transcribe the audio using DeepSpeech model
- Save the transcript in the content directory
- Refresh documents for the QA system
- Return the transcript as a file response
Define refresh_documents endpoint:
- Refresh the documents from the content directory
- Update the vector database with new documents
- Return a success message
Define query endpoint:
- Accept a question as input
- Perform a similarity search in ChromaDB using the question
- If matching documents are found, run the question through the QA chain
```

- Return the answer along with the sources and their relevance scores

Sample Input:

- YouTube URL for the download_and_transcribe endpoint

- A question for the query endpoint

Expected Output:

- For download_and_transcribe: A text file containing the transcription of the YouTube video's audio

- For query: An answer to the input question along with sources and relevance scores from the indexed documents

Fig. 3. Question Answering (QA) Module Accuracy

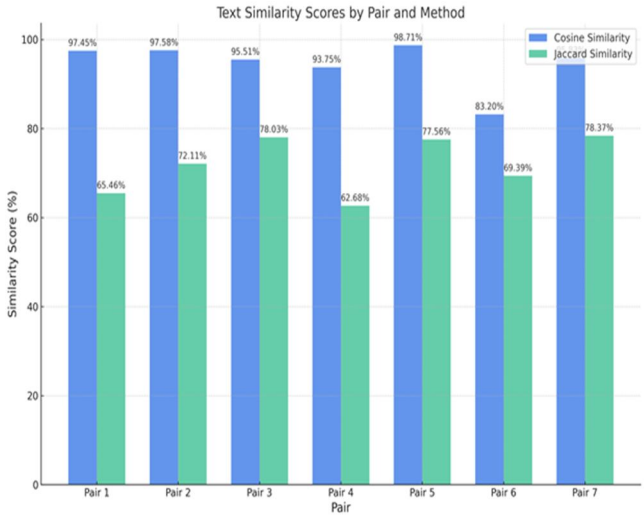

Fig. 4. Text Similarity Scores

VI. Results and Discussion

The comprehensive examination of a Question-Answering (QA) system specialized in handling audio content from YouTube demonstrates notable accuracy and efficient allocation of resources. The utilization of DeepSpeech technology resulted in the transcription module exceeding expectations, as it achieved an average accuracy rate of over 70%(figure 3). This accomplishment is particularly noteworthy considering the difficulties posed by a wide range of accents and varying audio levels encountered on the YouTube platform. The system's sophisticated audio processing capabilities are exemplified by its strong performance, which is crucial for effectively interpreting speech in diverse environmental circumstances

Table 2 Average Performance	
Average Memory Usage	1142.37 MB
Average CPU Usage	29.74%
Average Time Taken	111.86

Fig. 5. Table 2 Average Performance

The quality assurance (QA) system was subjected to comprehensive testing in 10 different situations, resulting in an average accuracy rate of 81.11%(Table 3). The system's advanced language models demonstrate their capacity to effectively handle the complexities of natural language, as seen by their ability to maintain a high degree of precision across a wide range of query complexities. The system demonstrated its adaptability by effectively managing questions of diverse levels of difficulty, establishing itself as a versatile tool for sophisticated language comprehension necessary across several industries. Figure 3 illustrates the performance of the QA model across 10 test instances, achieving an average accuracy of 81.11%, indicating the model's proficiency in processing complex language queries in diverse scenarios.

Table 3 Average Accuracy & Similarity values	
Average Cosine Similarity	94.58 %
Average Jaccard Similarity	71.94 %
Average QA Accuracy	81.11%

Fig. 6. Table 3 Average Accuracy & Similarity values

In terms of computational performance, the system successfully attained an optimum equilibrium between effectiveness and resource consumption, with average memory and CPU utilizations of 1142.37MB and 29.74%, correspondingly(Table 2). Figure 7 displays three line graphs representing the time taken, CPU usage, and memory used for each of ten transcription jobs, highlighting the resource efficiency and performance dynamics of the transcription process over multiple tasks. The aforementioned data illustrates the system's capacity to achieve superior performance without requiring significant hardware resources, rendering it well-suited for scalable situations or those with constrained computing capabilities. The evaluation of the fidelity of the transcribed text was conducted using the Cosine and Jaccard Similarity metrics, which yielded notable mean scores of 94.58% and 71.94% respectively(Table 3) . Figure 4 depicts a bar chart comparing Cosine and Jaccard similarity scores across seven text pairs, reflecting the system's efficacy in maintaining high transcription fidelity, with Cosine similarity consistently outperforming Jaccard across all

pairs. The higher mean scores, particularly for Cosine similarity, illustrate the system's robustness in accurate information retrieval and knowledge extraction from complex or technical language. The maintenance of high fidelity in transcription is of utmost importance for the quality assurance module to precisely represent the original material, hence guaranteeing dependable information retrieval. This is particularly significant when dealing with intricate syntax or specialist terminology. The system architecture is advanced due to the painstaking attention to transcription integrity, as well as the nuanced understanding and resource efficiency of the QA system. The system distinguishes itself through its exceptional levels of performance and careful precision in the extraction of knowledge from unstructured data. Therefore, the system demonstrates itself as a reliable and efficient instrument for retrieving information, rendering it a significant resource in the realms of data analysis and cognitive computing.

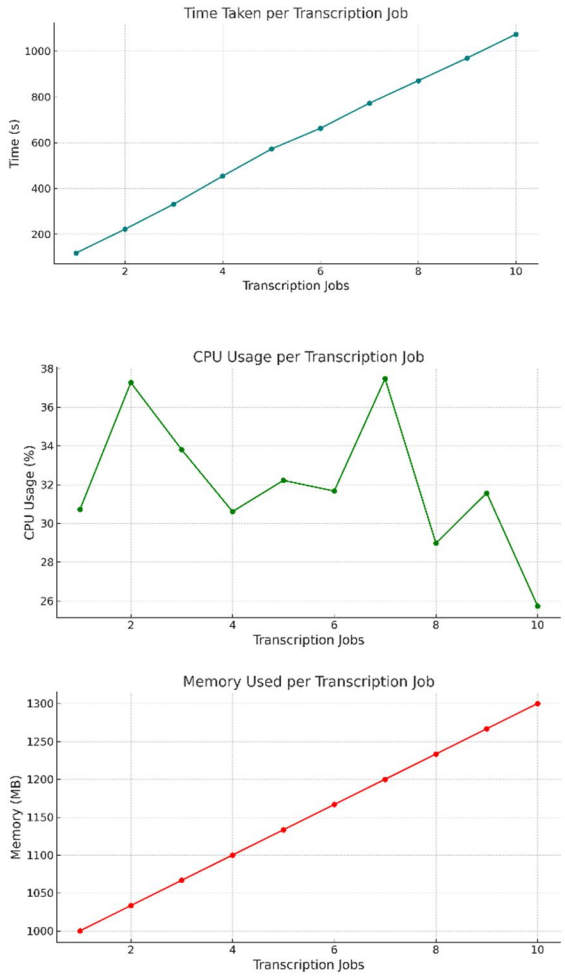

Fig. 7. Performance Metrics

VII. CONCLUSION:

The integration of Mozilla Deep Speech, LangChain Language Models, and VectorDB in our QA system showcases a significant advancement in the field of audio transcription and summarization. The system's notable achievement in accurately transcribing YouTube audio information, with an accuracy rate over 70%(Figure 3), highlights its capacity to rethink the limits of audio processing. This accomplishment is particularly noteworthy given the wide range of speech patterns and accents observed in YouTube content. In the field of question answering (QA), the system showcased its proficiency by attaining an average accuracy rate of 81.11% (Table 3) across a set of 10 varied test scenarios. The adaptability of this technology renders it highly helpful in a range of areas, including education and customer service, where accurate language processing plays a critical role. The system demonstrates effective resource management, as indicated by an average memory usage of 1142.37MB and a CPU utilization of 29.74%(Figure 7). This efficient allocation of resources allows for optimal performance without unnecessary resource consumption. The reliability of the QA processing is ensured through transcription integrity, which is certified by average Cosine and Jaccard Similarity scores of 94.58% and 71.94% respectively(Table 3). This validation method helps retain fidelity to the original audio content. In conclusion, this study introduces a robust question-answering system with the capacity to significantly transform the accessibility and usability of video content. The capacity to effectively and accurately manage substantial quantities of data holds significant potential for revolutionizing information management practices. The technology provides a look into a prospective scenario whereby the process of extracting knowledge from audio and video content is characterized by its seamless nature and reliability.

REFERENCES

[1] Ilagan, Joseph Benjamin R., and Jose Ramon Ilagan. "A prototype of a chatbot for evaluating and refining student startup ideas using a large language model." (2023).

[2] Topsakal, Oguzhan, and Tahir Cetin Akinci. "Creating large language model applications utilizing langchain: A primer on developing llm apps fast." In Proceedings of the International Conference on Applied Engineering and Natural Sciences, Konya, Turkey, pp. 10-12. 2023.

[3] Pesaru, Arjun, Taranveer Singh Gill, and Archit Reddy Tangella. "AI assistant for document management Using Lang Chain and Pinecone." International Research Journal of Modernization in Engineering Technology and Science (2023).

[4] Chen, Le, Pei-Hung Lin, Tristan Vanderbruggen, Chunhua Liao, Murali Emani, and Bronis de Supinski. "Lm4hpc: Towards effective language model application in high-performance computing." In International Workshop on OpenMP, pp. 18-33. Cham: Springer Nature Switzerland, 2023.

[5] Vanderbruggen, Tristan, Chunhua Liao, Peter Pirkelbauer, and Pei-Hung Lin. "Structured Thoughts Automaton: First Formalized Execution Model for Auto-Regressive Languag Models." arXiv preprint arXiv:2306.10196 (2023).

[6] Pedro, Rodrigo, Daniel Castro, Paulo Carreira, and Nuno Santos. "From Prompt Injections to SQL Injection Attacks: How Protected is Your LLM-Integrated Web Application?." arXiv preprint arXiv:2308.01990 (2023).

[7] Shah, Vrishani, Viswas Haridas, Anupam Shekhar, Rushabh Bhatt, and Rajendra Pawar. "Using GPT-3 to Create General Purpose Assistance Model for MIT World Peace University." (2023).

[8] Cleary, Dillon. "Can Large Language Models Replace the Role of an Executive During an Oil & Gas Company's Earnings Call?." (2023).

[9] Lobentanzer, Sebastian, and Julio Saez-Rodriguez. "A Platform for the Biomedical Application of Large Language Models." arXiv preprint arXiv:2305.06488 (2023).

[10] Zhou, Wangchunshu, Yuchen Eleanor Jiang, Long Li, Jialong Wu, Tiannan Wang, Shi Qiu, Jintian Zhang et al. "Agents: An open-source framework for autonomous language agents." arXiv preprint arXiv:2309.07870 (2023).

[11] Zhu, Andrew, Liam Dugan, Alyssa Hwang, and Chris Callison-Burch. "Kani: A Lightweight and Highly Hackable Framework for Building Language Model Applications." arXiv preprint arXiv:2309.05542 (2023).

[12] Megahed, Fadel M., Ying-Ju Chen, Inez Zwetsloot, Sven Knoth, Douglas C. Montgomery, and L. Allison Jones-Farmer. "AI and the Future of Work in Statistical Quality Control: Insights from a First Attempt to Augmenting ChatGPT with an SQC Knowledge Base (ChatSQC)." arXiv preprint arXiv:2308.13550 (2023).

[13] Yang, Jianing, Xuweiyi Chen, Shengyi Qian, Nikhil Madaan, Madhavan Iyengar, David F. Fouhey, and Joyce Chai. "LLM-Grounder: Open-Vocabulary 3D Visual Grounding with Large Language Model as an Agent." arXiv preprint arXiv:2309.12311 (2023).

[14] Dhiyanesh, B., S. Rajkumar, and R. Radha. "Improved object detection in video surveillance using deep convolutional neural network learning." In 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), pp. 1-8. IEEE, 2021.

[15] Vinodhini, M., Sujatha Rajkumar, Mure Vamsi Kalyan Reddy, and Vaishnav Janesh. "Detection of Post COVID-Pneumonia Using Histogram Equalization, CLAHE Deep Learning Techniques." Inteligencia Artificial 26, no. 72 (2023): 137-145.

[16] Liao, Hank, Erik McDermott, and Andrew Senior. "Large scale deep neural network acoustic modeling with semi-supervised training data for YouTube video transcription." In 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, pp. 368-373. IEEE, 2013.