International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 05 | May - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Content Summarization and Question Answering System Using LLM

Manikanta S
School of CIT
REVA University
Bengaluru, India
manikanta.srinivasan@gmail.com

Mythri Varadaraju
School of CIT
REVA University
Bengaluru, India
mythrivaradaraju2412@gmail.com

Rithika M Choudhary
School of CIT
REVA University
Bengaluru, India
rithikaz2652003@gmail.com

Ravuri Sri Venkata Naga Sai Mani Pavan
School of CIT
REVA University
Bengaluru, India
pavanravuri88@gmail.com

Sarvamangala D R
School of CIT
REVA University
Bengaluru, India
sarvamangala.dr@reva.edu.in

Abstract—This paper presents a novel approach to content summarization, a pivotal task in information processing and knowledge extraction. Our methodology integrates cutting-edge technologies, including large language models (LLMs) and ad-vanced retrieval techniques, to efficiently distill extensive textual data into concise and informative summaries. By leveraging Anyscale LLMs for language comprehension and employing prompt engineering for model guidance, we propose a frame-work that prioritizes efficient document indexing and rapid retrieval using DuckDB vector stores. Additionally, we introduce the RetrievalQAWithSourcesChain framework, which combines LLMs with retrievers for proficient question answering. Em-pirical validation demonstrates the efficacy of our approach in generating high-quality content summaries with minimal manual intervention, thereby enhancing information retrieval and knowledge extraction processes.

Index Terms—LLMs, Anyscale, prompt engineering, document indexing, DuckDB, vector stores, RetrievalQAWithSourcesChain framework, question answering, empirical validation

I. Introduction

Content summarization, a crucial aspect of information processing, enables the extraction of key insights from vo-luminous text data efficiently. In recent years, the conver-gence of advanced technologies, including large language models (LLMs) and sophisticated retrieval techniques, has revolutionized content summarization methodologies. This re-search paper delves into the utilization of these cutting-edge technologies to elevate the process of content summariza-tion, focusing on the intricate interplay between document comprehension and question answering. The primary aim of this paper is to investigate how the integration of LLMs, particularly Anyscale, with advanced retrieval techniques can enhance content summarization capabilities. By leveraging state-of-the-art methodologies and tools, this study seeks to address the inherent challenges associated with summarizing extensive textual content while preserving its essence and critical information. The paper commences by providing a comprehensive overview of content summarization, empha-sizing its significance in information retrieval and knowl-edge extraction contexts. Subsequently, it delves into the methodological framework adopted, which encompasses the utilization of Anyscale LLMs for nuanced language compre-hension, prompt engineering techniques to guide model behav-ior, and the implementation of efficient document indexing and retrieval mechanisms using vector stores. A substantial portion of the paper is dedicated to elucidating the exper-imental setup and workflow. This encompasses the process of loading PDF documents, segmenting text into manageable chunks, and generating sentence embeddings using advanced embedding techniques. The adoption of DuckDB as a vector store facilitates rapid and scalable document retrieval, thereby enabling the seamless extraction of pertinent information from textual data. Furthermore, the paper elucidates the de-velopment of a novel framework termed RetrievalQAWith-SourcesChain, which amalgamates LLMs with retrievers to execute question-answering tasks proficiently. By retrieving pertinent documents from the vector store and harnessing the language model's capabilities, the framework adeptly furnishes accurate and contextually enriched summaries of the content. In summary, this research paper underscores the remarkable strides made in content summarization through the integration of LLMs and retrieval techniques. Through empirical valida-tion and meticulous analysis, it demonstrates the efficacy of these technologies in enhancing the efficiency and precision of content summarization processes, thereby propelling the domain of information retrieval and knowledge extraction into new realms of possibility.

II. Literature Review

The effectiveness of aggregated prompting strategies in improving the performance of LLMs across various tasks without requiring additional training is explored in [1]. The paper introduces the Ask Me Anything (AMA)

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 05 | May - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

approach, which tackles the limitations of single prompts by utilizing multiple imperfect prompts and combining their outputs. AMA transforms inputs into question-answering formats iteratively, generating diverse interpretations that are merged through weak supervision. Experimental results across different LLMs and tasks demonstrate significant performance gains with AMA, particularly in scenarios with existing input knowledge. By effectively aggregating prompt responses, AMA addresses challenges related to prompt variability and dependency, empowering even smaller LLMs to achieve competitive performance compared to larger models on diverse benchmarks. This study underscores the potential of aggregated prompting strategies for facilitating efficient and robust LLM utilization.

HuggingGPT, a framework leveraging Large Language Models (LLMs) like ChatGPT to manage diverse AI models across domains is introduced in [2]. HuggingGPT operates through four stages: task planning, model selection, task execution, and response generation. In task planning, user requests are interpreted to identify solvable tasks, followed by model selection based on descriptions from the Hugging Face repository. Tasks are then executed using selected models, and outputs integrated for response generation. HuggingGPT extends ChatGPT's capabilities by handling complex AI challenges through coordination with expert AI models. Experimental results demonstrate its effectiveness in automating diverse AI tasks, highlighting its potential for advancing artificial general intelligence by combining LLM language understanding with specialized AI model knowledge.

Emergent abilities in LLMs, unique to larger scales and unpredictable from smaller models is explored in [3]. It examines whether these abilities can be amplified with additional scaling. Scaling LLMs generally improves performance and efficiency across various NLP tasks, but certain abilities emerge unexpectedly as models grow larger, indicating a phase transition in performance. These emergent abilities, absent in smaller models, include improvements in tasks like few-shot prompting. The paper illustrates this with examples from arithmetic, language understanding, and truthfulness tasks, where performance escalates beyond a certain model size. Additionally, it discusses implications for AI research, suggesting that understanding the emergence of these abilities could lead to more efficient AI development, potentially requiring fewer computational resources. This challenges existing scaling laws in AI and prompts a reconsideration of their limitations. Overall, the paper contributes to understanding how scaling impacts LLM capabilities and suggests that further increases in model size could reveal unforeseen abilities, advancing AI frontiers.

the significance of text summarization in managing vast digital data is examined in [4]. It covers various approaches, including abstractive and extractive methods, alongside query-based summarization, emphasizing their application in fields like medicine and news. The utilization of machine learning, neural networks, and deep learning techniques in text summarization is examined, with models evaluated on datasets like the CNN corpus and DUC2000. Researchers have explored diverse techniques, from machine learning to query-based approaches, evaluating their performance using metrics like ROUGE scores. Each method has its strengths, such as neural networks combined with Rhetorical Structure Theory or query-based summarization using TF-IDF and fuzzy logic. The paper concludes by highlighting ongoing research efforts aimed at refining text summarization models for more accurate and concise summaries.

The concept of extractive text summarization, which emphasizes the selection of pivotal sentences or paragraphs for generating concise summaries, is investigated in [5]. It categorizes text summarization methods into abstractive and extractive approaches, with a specific emphasis on the latter. The study delineates various features utilized in extractive summarization, encompassing content words, title words, cue phrases, biased words, uppercase words, as well as sentence-level attributes like location, length, and cohesion. Extractive methods are further categorized into unsupervised and supervised approaches, including graph-based, fuzzy logic-based, and concept-oriented models within unsupervised methods, and machine learning and neural networks within supervised methods. Evaluation metrics such as human assessment and ROUGE scores are discussed, along with challenges in automatic evaluation due to subjectivity, paving the way for future research avenues. Overall, the paper underscores the significance of extractive text summarization in efficiently condensing information.

[6] delves into the Chain-of-Questions Training with Latent Answers framework, crafted to bolster multistep question answering (QA) through the utilization of question decomposition techniques. Utilizing human-annotated question decomposition meaning representation (QDMR), the framework trains a model to sequentially generate sub-questions and their corresponding sub-answers. Through a combination of Hard-EM and reinforcement learning, specifically dynamic mixture of Hard-EM and memory augmented policy optimization (MAPO), the model optimizes latent sub-answers effectively. Experimental results demonstrate the framework's superior performance over existing neuro-symbolic and prompting methods on various multistep reasoning benchmarks such as DROP and HOTPOTQA, including their contrast and adversarial sets. Task-specific modifications, like incorporating a regular expression module for DROP and introducing supporting fact prediction and span prediction tasks for HOTPOTQA, further enhance the framework's efficacy. In summary, the Chain-of-Questions framework offers a robust solution for multistep QA, showcasing significant improvements through question decomposition strategies and hybrid learning techniques.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 05 | May - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

III. Methodology

A. Overall Architecture

The Content summarizing and question-answering system we're proposing leverages cutting-edge natural language pro-cessing methods, incorporating a blend of advanced tech-niques. This includes tapping into pre-trained language mod-els, document retrieval mechanisms, and embeddings. Our system's architecture is meticulously crafted to handle user queries swiftly and effectively extract pertinent information from a vast repository of documents.

B. Components

1) Prompt Template: A prompt template serves as a structured framework for presenting user input questions in a format that the language model can comprehend effectively. It essentially acts as a blueprint for how questions are framed and how responses are expected. The template typically comprises designated slots or placeholders for both the question itself and the expected answer. This format facilitates seamless integration with the question-answering pipeline.

Let's break down the components of a detailed prompt template:

1. Question Placeholder: This placeholder denotes where the user's question will be inserted within the template. It provides guidance to the language model about the nature and context of the inquiry. For example:

Question: What is the capital of France?

2. Answer Placeholder: Similarly, this placeholder indicates where the expected answer should be placed within the template. It defines the target output the model should generate in response to the given question. For instance:

Answer: The capital of France is Paris.

3. Contextual Information: In some cases, additional contextual information may be included in the prompt template to provide background or constraints for the question. This could involve specifying a particular domain, providing relevant facts, or setting limitations on the scope of the answer. For example:

Context: In the context of geography,

4. Formatting and Structure: The template should adhere to a consistent formatting and structure to ensure clarity and ease of processing. This may involve using specific keywords or cues to signal the start and end of different sections within the prompt. For instance:

Question: What city serves as the capital of [Country]?

By employing a well-defined prompt template, the question-answering system can efficiently interpret user queries, generate accurate responses, and seamlessly integrate with the underlying language model and processing pipeline.

2) Anyscale Model: We chose Mixtral 8x7B model for the task and the following gives a introduction and capabilities of the model. Mixtral 8x7B epitomizes the pinnacle of natural language processing technology, boasting a high-quality sparse mixture of experts (SMoE) model with open weights. Under the Apache 2.0 license, Mixtral sets itself apart by not only outperforming benchmarks but also offering a 6x faster inference speed compared to its counterparts, notably surpassing Llama 2 70B. What distinguishes Mixtral is not just its remarkable performance but also its accessibility and licensing. Positioned as the strongest open-weight model with a permissive license, Mixtral strikes an optimal balance between cost and performance, making it an unrivaled choice in the field. This model's capabilities are extensive and diverse. It adeptly handles a context of 32k tokens, enabling nuanced understanding of lengthy and intricate language constructs. Furthermore, its multilingual prowess encompasses English, French, Italian, German, and Spanish, ensuring versatility across linguistic domains. A standout feature of Mixtral lies in its exceptional performance in code generation tasks, demonstrating a keen ability to translate human-readable instructions into executable code with precision and efficiency. Moreover, its adaptability shines through in its capacity to be fine-tuned into an instruction-following model, achieving an impressive score of 8.3 on the MT-Bench evaluation metric. In essence, Mixtral represents a significant advancement in natural language processing, rivaling and often surpassing established models like GPT3.5 across various standard benchmarks. Its blend of performance, flexibility, and accessibility positions it as a frontrunner in propelling language-based AI applications forward.

3) Data Preprocessing: In our document processing sys-tem, PDF files are first imported using the PyPDFLoader module. This module plays a crucial role in extracting text, metadata, and other pertinent details from the PDFs. Loading PDF documents involves parsing the contents of the file and making them accessible for further processing. PyPDFLoader enables this by providing functions to read text and retrieve other relevant information from PDFs. This process ensures that the system can effectively work with the content within the documents.

a) Chunking Strategy: In our project, the RecursiveChar-acterTextSplitter played a pivotal role in handling document processing tasks, particularly when it came to managing PDF files efficiently. By employing this tool, we were able to break down documents into smaller, more manageable chunks based on individual characters.

Here's how we utilized this approach in our project:

Using a recursive approach, the RecursiveCharacterTextSplit-ter ensured that each segment met specific criteria, such as maximum character count or designated delimiters. The number of chunks generated varied depending on factors like the document's size and complexity, as well as the chunking parameters we set. This approach proved invaluable for improving scalability. By enabling parallel processing of document segments, we were able to distribute the workload across multiple processing units, significantly reducing overall processing time.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 05 | May - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Furthermore, the flexibility of the RecursiveCharacter-TextSplitter allowed it to seamlessly adapt to documents of varying sizes and complexities. This adaptability contributed to more efficient and streamlined document processing work-flows within our project.

4) Vectorization and Storage: We use the SentenceTrans-former model, particularly the ”all-MiniLM-L6-v2” version, known for its ability to grasp the subtle nuances of text. This model goes beyond basic word embeddings, capturing the full context of sentences for richer representation. Through semantic encoding, each text snippet is transformed into a multidimensional vector space, preserving intricate word re-lationships. This vectorization enables advanced similarity as-sessments and enhances our understanding of textual content. To ensure efficient storage and retrieval, we utilize DuckDB, a lightweight yet robust database designed for analytical tasks. Here, these vectorized text chunks are readily accessible, supporting our methodology’s effectiveness and scalability in processing inquiries with semantic depth

5) Document Retrieval: In this process, DuckDB acts as the engine facilitating swift and effective searches grounded in vector similarity. By comparing the vectorized representation of user queries with those stored in the database, relevant documents are swiftly identified. This retrieval mechanism is finely tuned to furnish the top-k most pertinent documents, de-termined by their similarity scores, ensuring that the retrieved results align closely with the user’s inquiry.

6) Question Answering Chain: In our project, we’ve imple-mented a RetrievalQAWithSourcesChain to create a sophis-ticated question-answering system that seamlessly combines language understanding with document retrieval. This chain architecture integrates the Anyscale model, renowned for its advanced language understanding capabilities, with a docu-ment retriever, allowing us to generate precise and contextually relevant answers to user queries. By harnessing the strengths of both components, our system ensures seamless integration of multiple functionalities, resulting in an end-to-end solution for question answering. This approach not only enhances our system’s ability to interpret and respond to user questions accurately but also ensures that the retrieved documents are highly relevant and informative. Overall, the utilization of the RetrievalQAWithSourcesChain in our project signifies a significant advancement in developing intelligent question-answering systems tailored to meet diverse user needs with utmost precision and efficiency.

IV. Result

Content summarizing involves distilling extensive textual material, like articles, documents, or books, into more concise versions that still capture the core ideas and crucial

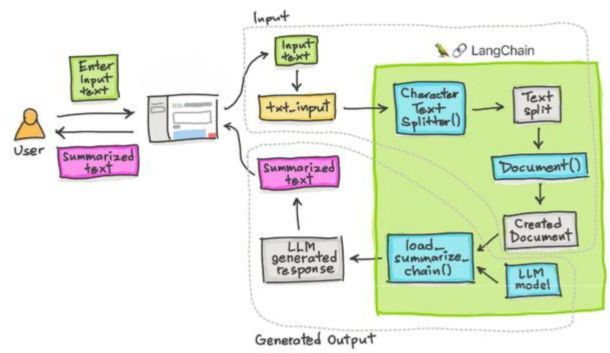

Fig. 1. Methodology Figure

points. Its aim is to furnish readers with a compact yet comprehensive snapshot of the original text, facilitating a swift understanding of its primary arguments, conclusions, or noteworthy details without necessitating a full read-through of the entire document.

The following is the result of content summarizing and a question answering system we worked on when we provided a pdf to the system.

1. Enter your question (or ’quit’ to exit): what is machine learning ?

Answer: ’question’: ’what is machine learning ?’, ’answer’: ’ Machine learning is the study of computer algorithms that improve automatically through experience.’, ’sources’: ’1-pl, 5-pl, 24-pl, 28-pl, path of the pdf’

In the above we can see it even gives a source for the answer its given and it even understand what we are looking for even when a word is misspelt.

2. For instance what if we ask it summarize a specific topic from the book

Enter your question (or ’quit’ to exit): summarize about support vector machines Answer: ’question’: ’summarize about support vector machines’, ’answer’: ’ A support vector machine (SVM) is a supervised machine learning algorithm which analyzes data and recognizes patterns, used for classification and regression analysis. The basic idea of SVMs is to find a hyperplane in high dimensional space that distinctly classifies the data points.’, ’sources’: ’thebook.pdf (p. 155), thebook.pdf (p. 174), thebook.pdf (p. 189), thebook.pdf (p. 194)’

As we can see it can even can summarize based on the topic we need.

3. It can even give out answers in one of these languages English, French, Italian, German and Spanish.

Example: Enter your question (or ’quit’ to exit): summarize about support vector machines and give the answer in French Answer: ’question’: ’summarize about support vector machines’, ’answer’: ”Une machine a` vecteurs de support (SVM) est un algorithme d’apprentissage automatique supervise´ qui analyse les donnees´ et reconnaˆıt les motifs, utilise´ pour la classification et l’analyse de regression´. L’idee´ de base des SVM est de trouver un hyperplan dans un espace

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 05 | May - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

de grande dimension qui classifie distinctement les points de donnees´.', 'sources': 'thebook.pdf (p. 155), thebook.pdf (p. 174), thebook.pdf (p. 189), thebook.pdf (p. 194)'

1) Model performance : • It gracefully handles a context of 32k tokens.
• It shows strong performance in code generation.
• t handles English, French, Italian, German and Spanish.
• IIt can be finetuned into an instruction-following model that achieves a score of 8.3 on MT-Bench.
When Mixtral is compared to the Llama 2 family and the GPT3.5 base model. Mixtral matches or outperforms Llama 2 70B, as well as GPT3.5, on most benchmarks.

	LLaMA 2 70B	GPT - 3.5	Mixtral 8x7B
MMLU (MCQ in 57 subjects)	69.9%	70.0%	**70.6%**
HellaSwag (10-shot)	87.1%	85.5%	86.7%
ARC Challenge (25-shot)	85.1%	85.2%	**85.8%**
WinoGrande (5-shot)	**83.2%**	81.6%	81.2%
MBPP (pass@1)	49.8%	52.2%	**60.7%**
GSM-8K (5-shot)	53.6%	57.1%	**58.4%**
MT Bench (for Instruct Models)	6.86	**8.32**	8.30

Fig. 2. Performance Comparison

2) Graph: On the following figure, we measure the quality versus inference budget tradeoff. Mistral 7B and Mixtral 8x7B belong to a family of highly efficient models compared to Llama 2 models.

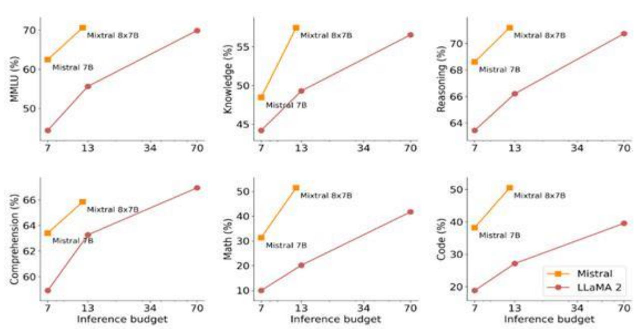

Fig. 3. Graph

V. CONCLUSION

In conclusion, our project represents a significant advancement in the field of natural language processing and document management. By integrating state-of-the-art techniques such as language understanding, document loading and splitting, content summarization, and question-answering, we've developed a comprehensive solution for efficiently processing and extracting insights from textual data. Our system's ability to seamlessly handle PDF documents, analyze their content, and generate accurate summaries or answers to user queries underscores its versatility and effectiveness. Whether it's summarizing lengthy reports, answering user questions in real-time, or extracting key information from documents, our project offers a powerful toolset for enhancing productivity and decision-making across various domains. Moving forward, further refinements and enhancements to our system promise to deliver even greater capabilities, cementing its role as a valuable asset in the realm of natural language processing and document management.

REFERENCES

[1] Simran Arora, Avanika Narayan, Mayee F. Chen, Laurel Orr, Neel Guha, Kush Bhatia, Ines Chami, Frederic Sala, Christopher Re´ - Ask Me Anything: A simple strategy for prompting language models.
[2] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, Yueting Zhuang - HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face.
[3] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, William Fedus - Emergent Abilities of Large Language Models.
[4] Rahul, Surabhi Adhikari, Monika-NLP based Machine Learning approaches for text summarization.
[5] N.Moratanch- Research Scholar, S.Chitrakala-A Survey on extractive Text Summarization.
[6] Wang Zhu, Jesse Thomason, Robin Jia - Chain-of-Questions Training with Latent Answers for Robust Multistep Question Answering .
[7] https://mistral.ai/news/mixtral-of-experts/ .
[8] https://www.rungalileo.io/blog/mastering-rag-how-to-architect-an-enterprise-rag-system .