International Journal on Science and Technology (IJSAT)
E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

Locally Deployed NLP System for Secure Document Summarization and Context-Aware Question Answering Using LLMs and Vector Embeddings.

Rama Krishna Pradhan[1], Durga Pavan Seerapu[2], Gayatri Routhu[3], Sarath Kumar Manda[4], Giri Rajasekhar Jami[5]

[1]M. Tech, Information Technology, Aditya Institute of Technology and Management, Tekkali
[2,3,4]Information Technology, Aditya Institute of Technology and Management, Tekkali, Andhra Pradesh, India

Abstract
This paper presents a locally deployed Natural Language Processing (NLP) system designed to perform secure document summarization and context-aware question answering. The proposed framework addresses critical concerns related to data privacy by eliminating dependence on external APIs or cloud-based services. By leveraging advanced Large Language Models (LLMs), vector databases, and modern NLP tools such as LangChain, Hugging Face, Qdrant, and Streamlit, the system enables users to interact intelligently with unstructured documents in a fully offline environment. The architecture facilitates efficient document parsing, semantic embedding, and retrieval-augmented generation (RAG), empowering users to generate concise summaries and retrieve precise information through natural language queries. This solution is particularly well-suited for privacy-sensitive domains such as healthcare, law, finance, and research, where local data control and confidentiality are essential. The framework not only enhances document comprehension but also promotes efficient information retrieval and workflow automation.

Keywords: Natural Language Processing, Text Summarization, Local Deployment, Semantic Search, Retrieval-Augmented Generation (RAG), Large Language Models, Qdrant, LangChain, HuggingFace, Streamlit.

1. Introduction

In the digital age, the proliferation of information has posed challenges in efficiently extracting meaningful and relevant data. With the exponential growth of online content, users often struggle to locate specific information or grasp the core ideas of lengthy documents. This has spurred the development of advanced Natural Language Processing (NLP) technologies to bridge this gap. Large Language Models (LLMs) and NLP frameworks such as LangChain and HuggingFace have revolutionized how we process and comprehend textual data. These technologies enable powerful tools for tasks like summarization, contextual understanding, and query-based information retrieval. By integrating such cutting-edge

advancements, it becomes possible to address challenges in navigating complex documents and providing precise, actionable insights [1] [2].The ability to efficiently summarize and interact with documents has transformative potential across domains such as education, business, healthcare, and research. Tools capable of summarizing URLs, PDFs, and articles not only save time but also enhance decision-making by distilling critical information. Additionally, functionalities like "Chat with PDF" empower users to engage interactively with documents, enabling deeper exploration and clarity on specific queries. By leveraging LLMs, NLP, and vector databases, these technologies enhance the accessibility and usability of vast data repositories. They promote informed decision-making, ensure clarity in communication, and democratize access to knowledge, paving the way for more effective workflows and enriched user experiences [3] [4].Key applications of these tools span across various sectors, providing significant value in diverse fields. In education, they assist students and professionals in quickly digesting extensive study materials, making learning more efficient. In corporate workflows, these tools streamline the review process of lengthy reports, contracts, and legal documents, saving time and improving decision-making. In healthcare, they simplify complex medical research papers, enabling practitioners and patients to access key information more easily. Media and journalism benefit by extracting concise summaries from news articles, interviews, and other content, allowing journalists to stay updated with minimal effort. Additionally, these tools can enhance productivity in sectors like law, finance, and research, where time-sensitive document review is crucial. By enabling faster comprehension of detailed content, they contribute to improved workflow and decision-making. They can also be applied to personal use, aiding individuals in quickly understanding long-form content[5] [6].

2. The integration of advanced NLP technologies and LLM frameworks like LangChain and HuggingFace offers numerous advantages. These tools enable efficient summarization of lengthy documents, saving time and reducing cognitive load for users. They enhance contextual understanding, ensuring that extracted insights are accurate and relevant. Query-based information retrieval capabilities allow users to interact with content conversationally, improving accessibility and engagement. Furthermore, these technologies support seamless navigation of complex datasets, delivering precise and actionable insights that enhance productivity and decision-making across various domains. By automating information extraction, they address the challenges of information overload and streamline workflows effectively[7].Current NLP systems and large language models (LLMs) face several challenges despite their advancements. They often struggle with maintaining semantic coherence and contextual understanding, especially in domain-specific or nuanced texts. Handling very long documents is problematic due to input size constraints, leading to potential loss of critical information. Additionally, the absence of standardized evaluation metrics complicates the assessment of summarization quality. These systems also reflect biases from training data, lack true reasoning capabilities, and require significant computational resources, limiting accessibility. Privacy concerns further arise when processing sensitive documents, highlighting the need for more secure and efficient solutions[8].

3. The authors proposed an algorithm for automatic text summarization that combines extractive and abstractive methods to improve accuracy and reduce redundancy, supporting multilingual summarization in English, Hindi, and Marathi. The approach uses NLP tools, including WordNet, to identify action words and employs algorithms involving Jaccard similarity, mean functions, and shingling for clustering and selecting relevant text segments. The method efficiently generates summaries with good coverage, avoiding redundancy, and supports multiple languages. The study concludes that this system enhances summarization accuracy, leveraging NLP and WordNet to create concise and meaningful summaries with

reduced delays [9], [10].The authors developed an extractive summarization model for newspaper articles in Kannada and English, using ontology and word scoring techniques. They implemented AutoSum, a tool that calculates keyword and sentence scores based on lexicons, word frequency, and sentence position. The approach combines statistical methods like term frequency with a customized XML-based Kannada lexical database for effective summarization. Machine-generated summaries aligned with human-generated ones by up to 80% in Kannada and 70% in English, with better results for longer summaries. AutoSum proved efficient for summarizing text in both languages. Future work includes expanding lexical databases and supporting more languages[11].The authors proposed HIBERT, a hierarchical bidirectional transformer model pre-trained for extractive document summarization, achieving state-of-the-art performance on the CNN/DailyMail and New York Times datasets. HIBERT uses hierarchical transformers to encode sentence- and document-level representations, pre-trained with a masked sentence prediction task on unlabeled documents and fine-tuned on summarization datasets to predict important sentences. The model outperformed previous methods, showing a 1.25 ROUGE improvement on CNN/DailyMail and 2.0 on NYT50, with fewer parameters compared to BERT. This hierarchical approach effectively captured relationships across sentences and documents, enhancing summarization quality and offering potential for broader hierarchical text-processing applications like document QA.

4. The authors developed a system to generate extractive summaries of legal documents using NLP and ML techniques, simplifying lengthy legal texts for quick comprehension. The process involved preprocessing steps like tokenization, text cleaning, and stop-word removal, followed by semantic vectorization of sentences using the Law2Vec embedding model. A similarity matrix was computed using cosine similarity, and sentences were ranked using the PageRank algorithm. The study demonstrated that the system effectively captures legal terminology, generating concise summaries that save time while preserving essential information. The authors concluded that the proposed system enhances legal document understanding and suggested future work on extending it to abstractive summarization for improved conciseness[12].

5. Huan Yee Koh et al. conducted an empirical survey on long document summarization, analyzing baseline and state-of-the-art systems using benchmarks like arXiv. They reviewed various methods, categorizing systems into supervised, unsupervised, neural, and hybrid approaches, with innovations such as discourse-aware and transformer-based architectures. The study found that models like Pegasus and Longformer achieve high performance but struggle with very long documents and maintaining semantic coherence. The authors concluded that advancements are required in handling discourse structures and developing hybrid approaches to overcome the limitations of current summarization systems[13].The authors proposed HIBERT, a hierarchical bidirectional transformer model pre-trained for extractive document summarization, achieving state-of-the-art performance on the CNN/DailyMail and New York Times datasets. HIBERT uses hierarchical transformers to encode sentence- and document-level representations, pre-trained with a masked sentence prediction task on unlabeled documents and fine-tuned on summarization datasets to predict important sentences. The model outperformed previous methods, showing a 1.25 ROUGE improvement on CNN/DailyMail and 2.0 on NYT50, with fewer parameters compared to BERT. This hierarchical approach effectively captured relationships across sentences and documents, enhancing summarization quality and offering potential for broader hierarchical text-processing applications like document QA[14].

2. Methodology

Natural Language Processing (NLP) is a field of artificial intelligence focused on enabling machines to understand, interpret, and generate human language. It plays a crucial role in applications like text summarization, translation, sentiment analysis, and question answering. Leveraging NLP, web applications can process large volumes of text data to extract meaningful insights. Modern frameworks such as Lang Chain facilitate building NLP pipelines that connect Large Language Models (LLMs) with external data sources, while Hugging Face provides access to pre-trained models for various language tasks. Tools like Qdrant offer vector storage for efficient semantic search, and Ollama aids in managing LLMs locally. Streamlit integrates these components into interactive web applications, enabling real-time summarization of PDFs and other text content, showcasing the power of NLP in enhancing information processing. The framework of the system is to develop a web application that summarizes the pdf using the Streamlit, Langchain, and NLPs in which the architecture of the model can be seen in figure 1.

2.1 Architecture:

This diagram represents the architecture of a retrieval-augmented generation (RAG) system, which combines a retriever and a large language model (LLM) to generate informed responses by leveraging external knowledge sources such as vector databases.

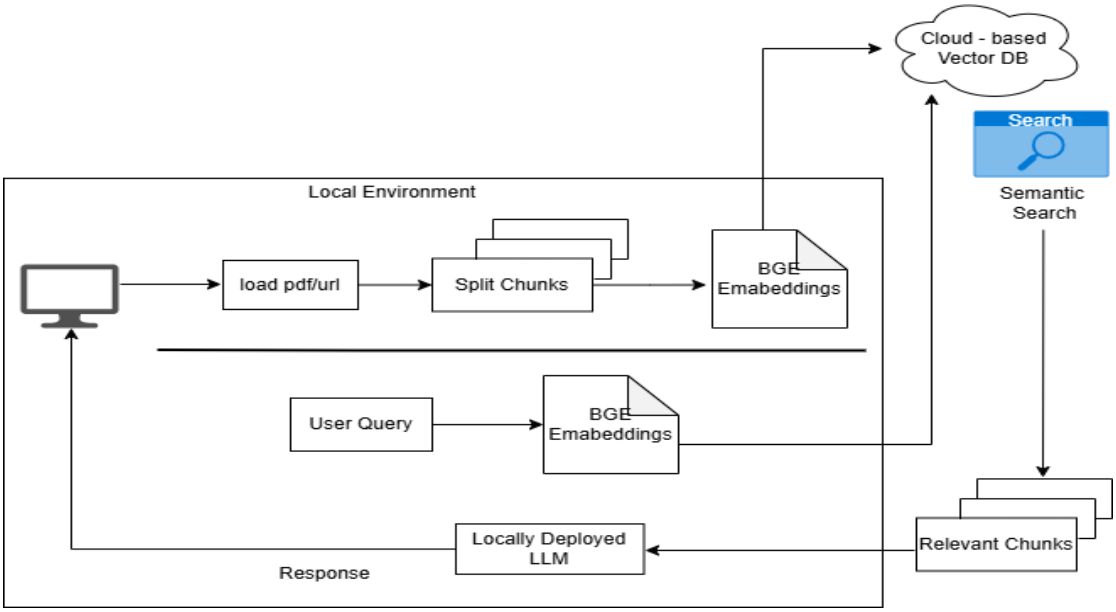

Figure 1: Architecture of the Model

The process outlined in the handwritten notes represents an advanced pipeline for handling unstructured PDF documents, converting them into vector embeddings, storing them efficiently, and using them for intelligent semantic search with a Large Language Model (LLM). Below is a step-by-step process:

I. **PDF Upload**:

- The process starts with uploading a PDF file. This step is performed through a web-based interface. This step ensures that the file is accessible for further processing.

II. **PDF Parsing:**
- Once the PDF is uploaded, it undergoes parsing using UnstructuredPDFLoader. This tool extracts text from the PDF, breaking it down into a structured format. At this stage, text is also chunked into smaller pieces to facilitate embedding and efficient retrieval.

III. **Text Embedding**:
- Once the text is extracted and chunked, it is converted into numerical representations called embeddings. Embeddings are high-dimensional vectors that capture the semantic meaning of words, sentences, or entire documents.
- Embedding Models: The process typically uses a model is Hugging Face transformers to generate these embeddings. These models ensure that semantically similar sentences are mapped closer in vector space. Each chunk of text is passed through the embedding model. The model outputs a fixed-size vector representation of the chunk. These vectors capture meaning rather than just word similarity.

IV. **Storage in a Vector Database**:
- The generated embeddings are stored in a vector database (Vector DB). A vector database is optimized for storing and retrieving high-dimensional vectors, making it suitable for semantic search.

V. **Query Processing and Vector Retrieval:**
- When a user submits a query, it is also converted into an embedding (a vector representation) and compared against the stored vectors in the vector database. The database retrieves the most relevant text chunks based on their similarity to the query.

VI. **Semantic Search and Query Expansion:**
- The retrieved chunks, which contain semantically relevant information, are combined with the query. This step ensures that related content is retrieved and passed to the next stage, improving the response quality.

VII. **LLM Processing and Response Generation:**
- The query, along with the related chunks, is fed into a large language model (LLM). The LLM processes the information, understands the context, and generates a relevant output or response for the user.

This process creates an efficient AI-powered document search and question-answering system. By leveraging PDF parsing, embeddings, vector databases, semantic search, and LLMs, it provides decent accurate, context-aware responses to user queries. This workflow can be applied in legal document search, research paper retrieval, customer support automation, and enterprise knowledge management.

2.2 Proposed Block Diagram:

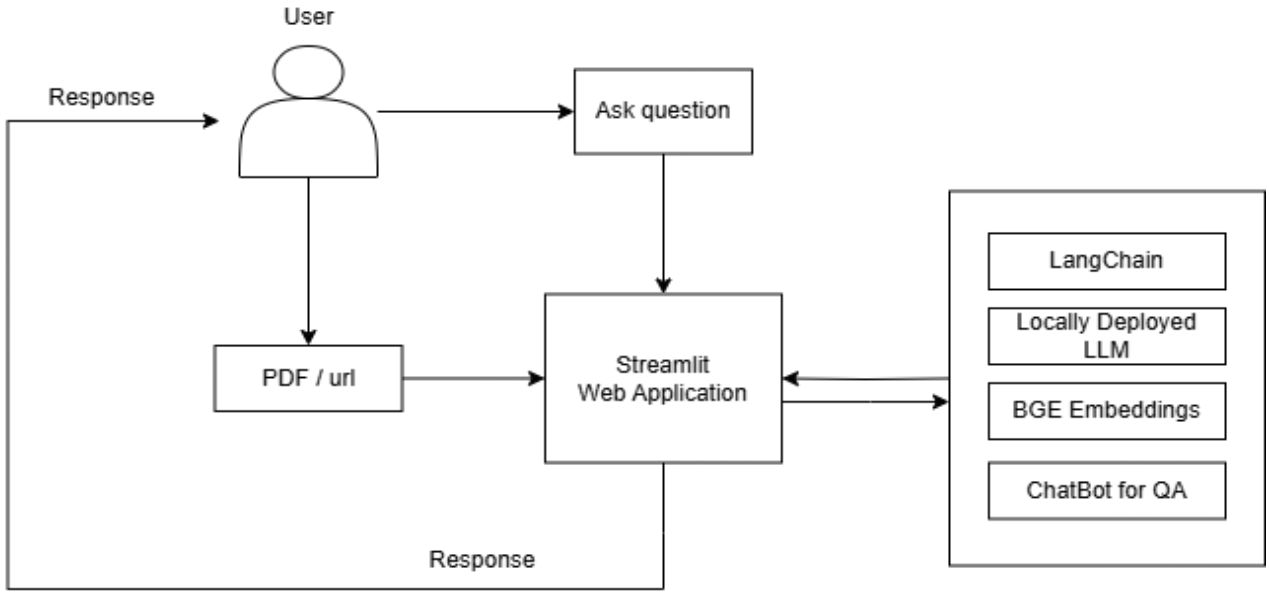

Figure 2: Proposed Block Diagram

The proposed system architecture is designed to facilitate secure, efficient, and interactive document analysis by leveraging locally deployed NLP components and large language models (LLMs). The flow begins at the User Interface (UI), where users can upload documents, submit queries, or interact with the system via a chatbot. This interface can be developed using lightweight frameworks like Streamlit for web applications or integrated into more complex front-end platforms for desktop or mobile access.

Once input is received, it is routed through a Preprocessing and Embedding Layer, where queries or document content are tokenized and transformed into vector representations using models such as BERT, SBERT, or Hugging Face's transformer models. These embeddings preserve the semantic meaning of the text and are essential for similarity-based retrieval.

The embeddings are stored in a Vector Database, such as Qdrant, which is optimized for high-dimensional data storage and fast similarity search. When a user asks a question, the system generates an embedding for the query and searches the vector database to find the most contextually relevant document segments. The retrieved segments are then passed to a Large Language Model (LLM)—such as LLaMA, GPT, or Falcon—which performs Retrieval-Augmented Generation (RAG) to synthesize accurate, context-aware responses. This ensures that the model produces outputs grounded in the uploaded documents, improving relevance and precision.

This modular architecture allows for secure, offline deployment while offering a scalable and intelligent solution for document summarization and interactive question answering, particularly suited to domains with strict data confidentiality requirements.

3. Results and Discussions

The Application Interface is a user-friendly platform designed for managing and interacting with documents efficiently. It is built using an open-source stack that includes Llama 3.2, BGE Embeddings, and Qdrant running within a Docker container. The interface allows users to easily upload PDF documents,

generate concise summaries, and interact with the document content using an intelligent chatbot. With a clean and organized layout, it offers seamless navigation and enhances the document management experience by combining document processing with AI-driven insights.

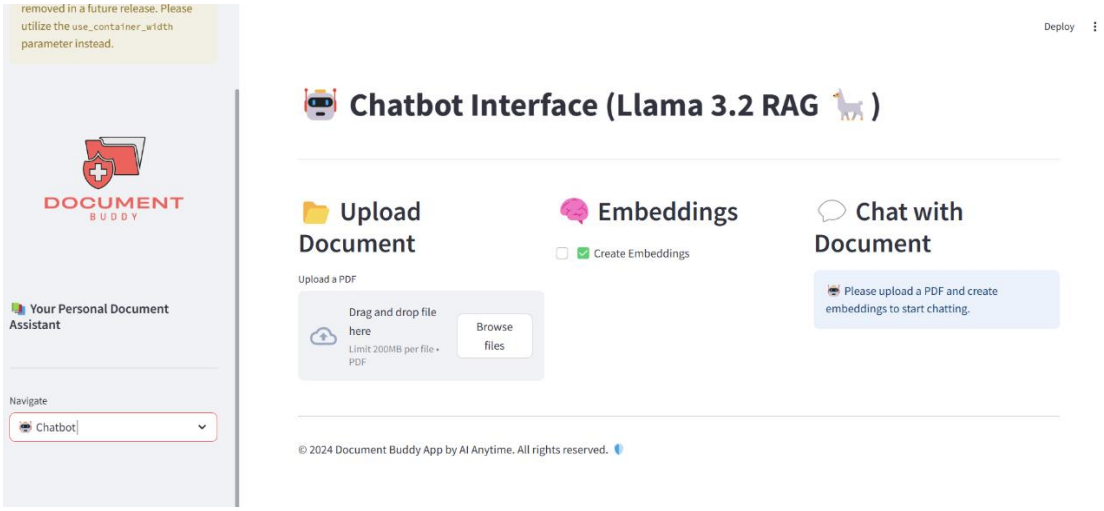

Figure 3: Interface of the Application

The Application interface is designed to provide a seamless and interactive document analysis experience by integrating Llama 3.2 RAG for efficient document querying. The interface consists of three primary sections: Upload Document, where users can drag and drop or browse PDF files (up to 200MB); Create Embeddings, which generates embeddings from the document content to facilitate semantic search; and Chat with Document, which allows users to interact with the document through a conversational interface.

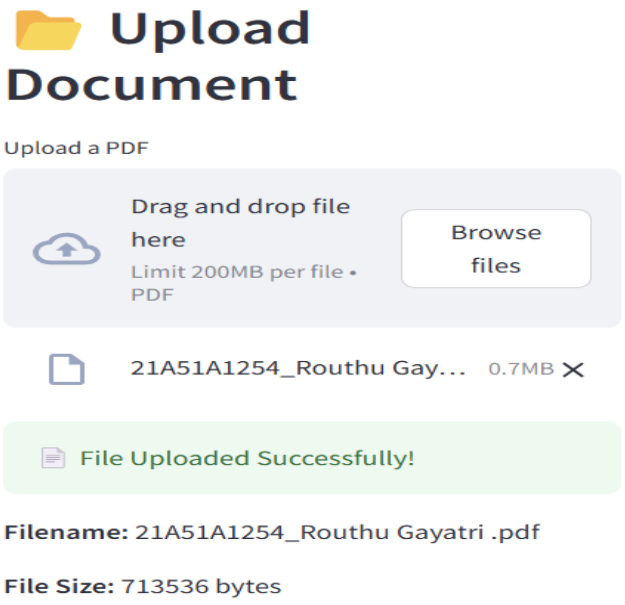

Figure 4: Uploading the pdf

The Document Upload section of the application interface allows users to easily upload PDF files by either dragging and dropping the file or using the Browse files button to select a document manually. The interface supports PDF files with a size limit of 200MB per file. Once the file is successfully uploaded, a confirmation message stating "File Uploaded Successfully!" is displayed, along with details such as the filename and the file size in bytes. This intuitive and user-friendly interface ensures a smooth document upload process, paving the way for subsequent embedding creation and interactive document querying.

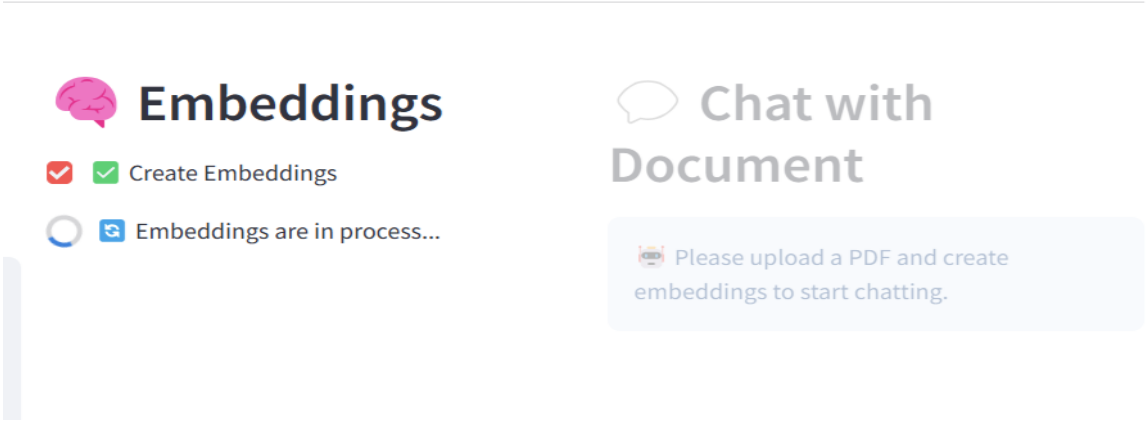

Figure 5: Creating the Embeddings

The Qdrant Collections Interface provides a user-friendly platform to manage and monitor vector-based databases efficiently. It is widely used in applications such as **semantic search, recommendation systems, and document retrieval** by storing and querying high-dimensional vector embeddings. The interface allows users to view the status and configuration of collections, ensuring smooth handling of vector data. It displays important details such as the collection name, status, number of points, segments, and shards. Additionally, the vector configuration specifies the distance metric used for similarity calculations, which in this case is **Cosine similarity** with a vector size of **384 dimensions.** Users can also perform actions like uploading snapshots, managing collections, and monitoring performance, making Qdrant an ideal choice for scalable vector database management.

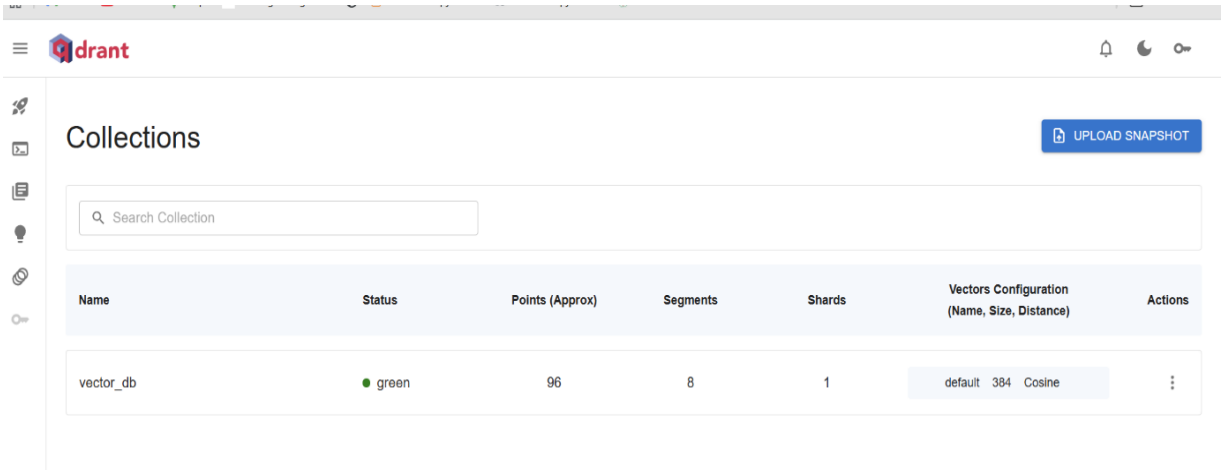

Figure 6: Qdrant Collections Interface

The embedding generation process transforms textual content from uploaded PDF documents into high-dimensional vector representations, enabling efficient retrieval of relevant information. By capturing semantic relationships within the text, embeddings allow the system to perform similarity-based searches, ensuring that user queries retrieve the most contextually relevant document segments. This process is a crucial component of the Retrieval-Augmented Generation (RAG) framework, as it bridges the gap between static document storage and dynamic, AI-driven interaction. Once embeddings are created, they serve as the foundation for intelligent querying, allowing Llama 3.2 to generate coherent and contextually appropriate responses based on retrieved document fragments.

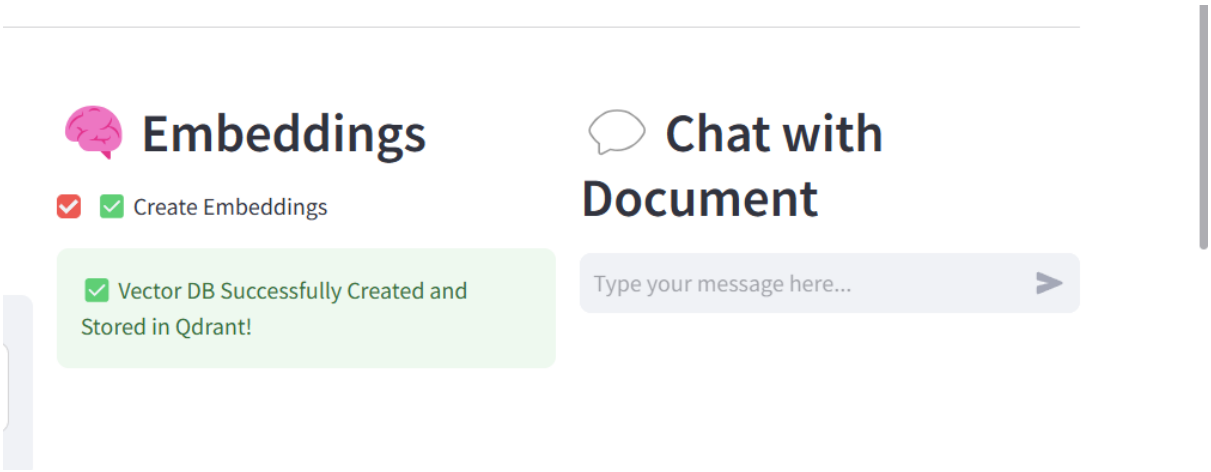

Figure 7: Q/A Chatbot for Question-Answering

The embedding generation process in application stores vector representations in Qdrant, a high-performance vector database designed for efficient similarity search. Once embeddings are created from the uploaded document, they are indexed in Qdrant, enabling fast and accurate retrieval of relevant text segments based on user queries. This database ensures scalable and optimized storage, allowing the chatbot to efficiently process large documents and provide precise, context-aware responses in real-time. Once the PDF is uploaded, the question-answering chatbot is initialized and a question from the user about the PDF is asked.

The prompt given by user is:

PROMPT 1: From the given pdf file “NEWS ARTICLE.pdf” summarize all the information that you know in about 200 words. (shown in figure 8)

3. Result

The study investigated independent strategies for accurate text summarization, exploring both extractive and abstractive methods. It highlighted the importance of natural language processing (NLP) techniques and modern approaches such as Google's PEGASUS model in efficiently managing vast amounts of information.

The PEGASUS model was analysed in detail, showcasing its architecture and sequence-to-sequence learning approach. The results demonstrated the model's ability to generate accurate and human-like summaries across various dimensions. The study concluded that these summarization methods are crucial for preserving the original meaning while condensing it into concise yet comprehensive summaries.

Extractive text summarization involves selecting specific phrases or sentences from the original text, whereas abstractive summarization generates new text based on the input. The PEGASUS model falls under the category of abstractive summarization, utilizing NLP techniques to understand the context and generate human-like summaries.

The study's findings suggest that these independent strategies for accurate text summarization can be effective in managing vast amounts of information online.

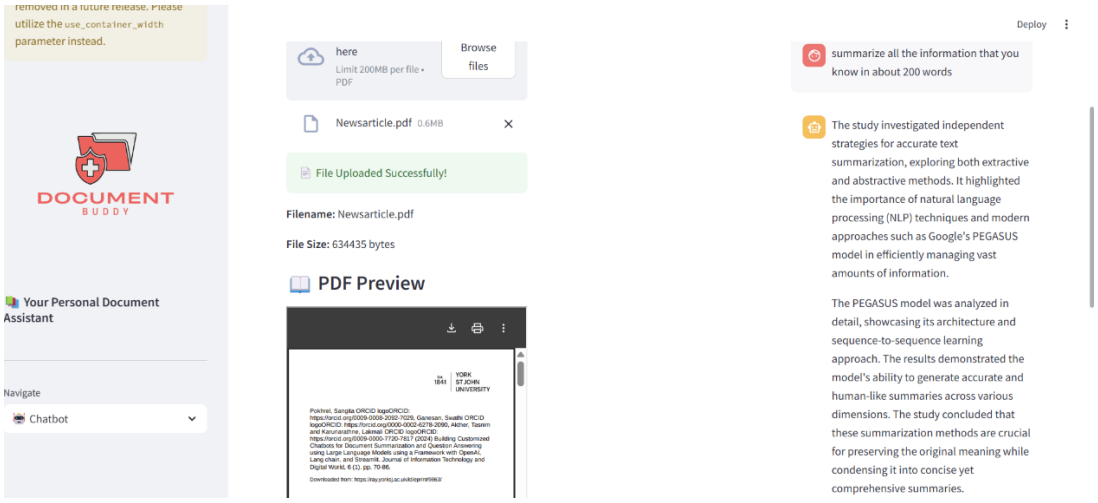

Figure 8: Results for prompt1

4. Conclusions

The proposed locally deployed NLP system successfully addresses the growing demand for secure, intelligent document processing in privacy-sensitive environments. By eliminating reliance on external APIs and cloud-based services, the framework ensures complete data confidentiality while delivering advanced capabilities for document summarization and question answering. The integration of large language models, vector databases, and semantic search techniques enables users to interact with unstructured documents in a meaningful and efficient manner. This system demonstrates that it is possible to achieve high-performance natural language understanding without compromising data security. Its modular design and use of open-source tools like LangChain, Hugging Face, Qdrant, and Streamlit make it flexible, scalable, and suitable for deployment across a variety of domains including law, healthcare, education, and enterprise knowledge management. The approach not only enhances document comprehension but also streamlines workflows by automating information retrieval and reducing manual effort. Future work may include expanding language support, improving summarization accuracy for domain-specific content, and incorporating reinforcement learning to enhance response quality over time. Overall, the system sets a strong foundation for privacy-preserving AI applications in document intelligence.

References

1. M. S. Daley, D. Gever, H. F. Posada-Quintero, Y. Kong, K. Chon, and J. B. Bolkhovsky, "Machine Learning Models for the Classification of Sleep Deprivation Induced Performance Impairment During a Psychomotor Vigilance Task Using Indices of Eye and Face Tracking," *Front. Artif. Intell.*, vol. 3, p. 17, Apr. 2020, doi: 10.3389/frai.2020.00017.
2. P. Martínez-Gómez, K. Mineshima, Y. Miyao, and D. Bekki, "On-demand Injection of Lexical Knowledge for Recognising Textual Entailment," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, Valencia, Spain: Association for Computational Linguistics, 2017, pp. 710–720. doi: 10.18653/v1/E17-1067.
3. M. Yang *et al.*, "Sentence-Level Agreement for Neural Machine Translation," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy: Association for Computational Linguistics, 2019, pp. 3076–3082. doi: 10.18653/v1/P19-1296.
4. Z. Jing, Y. Su, and Y. Han, "When Large Language Models Meet Vector Databases: A Survey," Nov. 01, 2024, *arXiv*: arXiv:2402.01763. doi: 10.48550/arXiv.2402.01763.
5. R. Figari Jordan, S. Sandrone, and A. M. Southerland, "Opportunities and Challenges for Incorporating Artificial Intelligence and Natural Language Processing in Neurology Education," *Neurology Education*, vol. 3, no. 1, p. e200116, Mar. 2024, doi: 10.1212/NE9.0000000000200116.
6. A. Faccia and P. Petratos, "NLP And IR Applications For Financial Reporting And Non-Financial Disclosure. Framework Implementation And Roadmap For Feasible Integration With The Accounting Process," in *Proceedings of the 2022 6th International Conference on Natural Language Processing and Information Retrieval*, Bangkok Thailand: ACM, Dec. 2022, pp. 117–124. doi: 10.1145/3582768.3582796.
7. I. Awasthi, K. Gupta, P. S. Bhogal, S. S. Anand, and P. K. Soni, "Natural Language Processing (NLP) based Text Summarization - A Survey," in *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, Coimbatore, India: IEEE, Jan. 2021, pp. 1310–1317. doi: 10.1109/ICICT50816.2021.9358703.
8. A. Afzal, J. Vladika, D. Braun, and F. Matthes, "Challenges in Domain-Specific Abstractive Summarization and How to Overcome Them:," in *Proceedings of the 15th International Conference on Agents and Artificial Intelligence*, Lisbon, Portugal: SCITEPRESS - Science and Technology Publications, 2023, pp. 682–689. doi: 10.5220/0011744500003393.
9. VAISHALI P. KADAM, SAMAH ALI ALAZANI, and C. NAMRATA MAHENDER, "A TEXT SUMMARIZATION SYSTEM FOR MARATHI LANGUAGE," Sep. 2022, doi: 10.5281/ZENODO.7073510.
10. M. Sharma, G. Goyal, A. Gupta, R. Rani, A. Sharma, and A. Dev, "Evaluating Multilingual Abstractive Dialogue Summarization in Indian Languages using mT5-small & IndicBART," in *2024 IEEE 9th International Conference for Convergence in Technology (I2CT)*, Pune, India: IEEE, Apr. 2024, pp. 1–6. doi: 10.1109/I2CT61223.2024.10543588.
11. J. S. Kallimani, K. G. Srinivasa, and B. Eswara Reddy, "Summarizing News Paper Articles: Experiments with Ontology- Based, Customized, Extractive Text Summary and Word Scoring," *Cybernetics and Information Technologies*, vol. 12, no. 2, pp. 34–50, Jun. 2012, doi: 10.2478/cait-2012-0011.

12. R. C. Kore, P. Ray, P. Lade, and A. Nerurkar, “Legal Document Summarization Using Nlp and Ml Techniques,” *int. jour. eng. com. sci*, vol. 9, no. 05, pp. 25039–25046, May 2020, doi: 10.18535/ijecs/v9i05.4488.
13. H. Y. Koh, J. Ju, M. Liu, and S. Pan, “An Empirical Survey on Long Document Summarization: Datasets, Models, and Metrics,” *ACM Comput. Surv.*, vol. 55, no. 8, pp. 1–35, Aug. 2023, doi: 10.1145/3545176.
14. X. Zhang, F. Wei, and M. Zhou, “HIBERT: Document Level Pre-training of Hierarchical Bidirectional Transformers for Document Summarization,” 2019, *arXiv*. doi: 10.48550/ARXIV.1905.06566.
15. Sun, Lanfang, Lu Jiang, Menghui Li, and Dacheng He. "Statistical analysis of gene regulatory networks reconstructed from gene expression data of lung cancer." Physica A: Statistical Mechanics and its Applications 370, no. 2 (2006): 663–671.
16. Woenckhaus, M., Klein-Hitpass, L., Grepmeier, U., Merk, J., Pfeifer, M., Wild, P., Bettstetter, M., Wuensch, P., Blaszyk, H., Hartmann, A., Hofstaedter, F., & Dietmaier, W. (2008). Global gene expression analysis reveals specific patterns of cell junctions in non-small cell lung cancer subtypes. Lung Cancer, 61(2), 180–190.
17. Zhang, Y., Li, X., Liu, B., & Ge, J. (2021). Potential predictive value of plasma heat shock protein 90α in lung cancer. Journal of International Medical Research, 49(12), 030006052110670.
18. Kumar, A., & Singh, A. (2024). Screening and identification of gene expression in large cohorts of lung adenocarcinoma patients: A meta-analysis approach. Gene Reports, 38, 101234.
19. Marshall, E. A., Ng, K. W., Anderson, C., Hubaux, R., Thu, K. L., Lam, W. L., & Martinez, V. D. (2015). Gene expression analysis of microtubule affinity-regulating kinase 2 in non-small cell lung cancer. Genomics Data, 6, 145–148.
20. Huang, X., Zhang, Y., Li, Y., Wang, Y., & Liu, J. (2023). Cell-type-specific alternative polyadenylation promotes oncogenic gene expression in non-small cell lung cancer. Molecular Therapy – Nucleic Acids, 33, 1–15.