MULTIDISCIPLINARY RESEARCH INSTITUTE INDIA

Multidisciplinary Journal of Research in Engineering and Technology

ISSN: 2348-6953

Volume 13 Issue 2s (2026) | Pages: 81-88

RAG-Powered Geoscience Assistant

Farendrakumar Ghodichor[1], Harish Sudam Wani[2], Paras Narendra Lad[3], Kunal Devidas Landge[4], Pratik Jaywant Ghumare[5].

[1,2,3,4,5]Dept. of dept of Artificial Intelligence and Data Science, DYPCOEI, Pune, India

[1]farendrakumar.ghodichor@dypatilef.com, [2]harishwani010@gmail.com, [3]ladparas29@gmail.com, [4]landgekunal2003@gmail.com, [5]ghumarepratik79@gmail.com.

Peer Review Information

***Type**: Article*
***Received**: 26 March 2026*
***Revised**: 10 April 2026*
***Accepted**: 24 May 2026*
***Published**: 15 June 2026*

Abstract

The increasing volume of geoscience-related data from research papers, environmental reports, and satellite observations has created challenges in efficient knowledge retrieval and interpretation. Traditional keyword-based search systems fail to provide context-aware and precise responses for complex queries. This paper presents a Retrieval-Augmented Generation (RAG)-Powered Geoscience Assistant that integrates vector-based retrieval with large language models (LLMs) to generate accurate and context-aware answers. The system processes user-uploaded geoscience documents, converts them into embeddings, and stores them in a vector database for semantic retrieval. A Streamlit-based interface enables interactive querying, while the backend utilizes embedding models and LLM APIs to deliver grounded responses. Experimental evaluation demonstrates improved accuracy, reduced hallucination, and enhanced user experience compared to traditional search systems. The proposed system provides an efficient and scalable solution for domain-specific knowledge retrieval in geoscience.

Keywords: RAG; Geoscience; LLM; Vector Database; Embeddings; Streamlit; AI Assistant.

How to Cite This Article

Ghodichor, F., Wani, H. S., Lad, P. N., Landge, K. D., & Ghumare, P. J. (2026). RAG-powered geoscience assistant. *Multidisciplinary Journal of Research in Engineering and Technology*, 13(2s), 81–88.

Introduction

The geoscience domain encompasses various fields such as geology, meteorology, oceanography, and environmental science, all of which generate massive amounts of structured and unstructured data. With the advancement of sensing technologies, satellites, and computational tools, the volume of geoscientific data has increased exponentially. Extracting meaningful insights from such large datasets is a critical challenge.

Traditional information retrieval systems rely on keyword matching and structured queries. These systems often fail to understand the context of user queries, leading to irrelevant or incomplete results. For example, a query related to "climate change impact on coastal regions" requires semantic understanding rather than simple keyword matching.

Artificial Intelligence (AI), particularly Natural Language Processing (NLP), has introduced new approaches to information retrieval. Large Language Models (LLMs) can understand and generate human-like text, enabling more intuitive interaction with data. However, standalone LLMs suffer from limitations such as hallucination, lack of domain-specific knowledge, and inability to access real-time information.

To overcome these challenges, Retrieval-Augmented Generation (RAG) has emerged as an effective solution. This paper proposes a RAG-powered Geoscience Assistant that enhances information retrieval by integrating semantic search with LLM-based generation.

In addition to the challenges associated with data volume and complexity, geoscience research also requires the integration of knowledge from multiple interdisciplinary sources. Researchers and practitioners often rely on a combination of textual reports, datasets, and domain-specific terminology to analyze environmental patterns, natural resources, and geological phenomena. This creates a need for intelligent systems that not only retrieve information but also synthesize and present it in a meaningful and user-friendly manner.

Furthermore, with the increasing demand for real-time decision-making in areas such as disaster management, climate monitoring, and resource exploration, it becomes essential to develop systems that can provide quick, reliable, and context-aware insights. The proposed RAG-based approach addresses these requirements by enabling seamless interaction between users and large-scale geoscience knowledge bases, thereby enhancing both accessibility and interpretability of complex information.

Literature Review

Traditional geoscience data retrieval systems are primarily based on structured databases and keyword-driven search mechanisms. These systems are effective in handling well-defined queries but fail to address complex queries that require semantic understanding and contextual reasoning. Geographic Information Systems (GIS) and relational databases have been widely used for managing geospatial data; however, they are limited in their ability to process natural language queries and generate explanatory responses.

With the advancement of artificial intelligence, Large Language Models such as GPT and BERT have been introduced to improve natural language understanding and generation. These models are capable of processing complex queries and generating detailed responses. However, they rely on pre-trained knowledge and do not have access to updated or domain-specific datasets, which can lead to inaccuracies and hallucinated outputs.

Retrieval-Augmented Generation (RAG) addresses these limitations by combining the strengths of retrieval systems and generative models. In a RAG framework, relevant documents are retrieved from a knowledge base and provided as context to the generative model, ensuring that the output is grounded in actual data. Recent research has demonstrated the effectiveness of RAG in domain-specific applications such as healthcare, legal systems, and education. The integration of vector databases such as FAISS and Chroma has further improved the efficiency of semantic search by enabling fast similarity-based retrieval of high-dimensional embeddings.

Another important area of research in information retrieval focuses on the use of semantic search techniques powered by embeddings. Unlike traditional keyword-based approaches, semantic search leverages vector representations of text to capture contextual meaning and relationships between words. Techniques such as Word2Vec, GloVe, and transformer-based embeddings have significantly improved the ability of systems to understand user intent and retrieve relevant information. Vector databases such as FAISS and Chroma have been widely adopted to store and efficiently search these high-dimensional embeddings. These advancements have enabled faster and more accurate retrieval in large-scale datasets. However, semantic search alone does not generate explanatory responses, and users are still required to interpret retrieved documents manually, which limits its effectiveness in knowledge-intensive domains like geoscience.

In recent years, pipeline-based frameworks such as LangChain have been introduced to simplify the integration of retrieval and generation components in AI applications. These frameworks provide modular architectures that allow developers to connect document loaders, embedding models, vector databases, retrievers, and language models into a cohesive system. Such frameworks have accelerated the development of domain-specific assistants by reducing implementation complexity and improving scalability. Despite these advantages, challenges remain in optimizing retrieval accuracy, managing large document collections, and ensuring low latency in real-time

applications. These limitations highlight the need for carefully designed RAG systems that balance retrieval efficiency with generation quality, particularly in data-intensive fields such as geoscience.

Methodology And System Architecture

The proposed RAG-based Geoscience Assistant is designed as a multi-stage pipeline that integrates document processing, embedding generation, semantic retrieval, and response generation. The system begins with document ingestion, where user-provided geoscience documents are uploaded and processed. These documents are preprocessed and divided into smaller text chunks to improve retrieval accuracy and efficiency.

Each text chunk is then converted into a numerical representation using an embedding model. These embeddings capture the semantic meaning of the text and are stored in a vector database such as FAISS or Chroma. The vector database enables efficient similarity-based search by comparing the embedding of a user query with stored document embeddings.

When a user submits a query, the system converts the query into an embedding and performs a similarity search in the vector database. The most relevant document chunks are retrieved based on cosine similarity, which is mathematically defined as:

$$Sim(A, B) = (A \cdot B) / (||A||\ ||B||)$$

The retrieved documents are then passed as contextual input to the Large Language Model. The LLM generates a response by combining the user query with the retrieved context, ensuring that the output is both relevant and grounded in actual data. The overall RAG formulation can be expressed as:

$$P(y \mid x) = \sum_{d \in D} P(y \mid x, d) \cdot P(d \mid x)$$

where x represents the query, d represents retrieved documents, and y represents the generated response.

The system is implemented using Python, with Streamlit providing an interactive user interface. HTTPX is used for efficient API communication with the LLM, while OS and SYS libraries handle system-level operations.

The proposed system follows a structured pipeline in which each component performs a specific task to ensure efficient and accurate information retrieval. The document ingestion module is responsible for accepting user-uploaded files, primarily in PDF format, and extracting textual content using parsing techniques. This extracted text is further processed through a chunking mechanism, where large documents are divided into smaller segments to improve retrieval granularity. Chunking plays a crucial role in enhancing the relevance of retrieved results, as smaller text segments allow for more precise matching with user queries. These processed chunks are then passed to the embedding module, which transforms textual data into high-dimensional vector representations that capture semantic meaning.

The retrieval mechanism operates by comparing the embedding of the user query with stored document embeddings in the vector database. A similarity function, typically cosine similarity, is used to identify the most relevant document chunks. These retrieved chunks are then forwarded to the Large Language Model, which generates a final response by combining the query with contextual information. This approach ensures that the generated output is grounded in actual data rather than relying solely on pre-trained knowledge. The modular architecture of the system allows for flexibility, enabling components such as embedding models or vector databases to be replaced or upgraded without affecting the overall workflow.

The major components of the system architecture include:

- Document Loader for PDF processing and text extraction
- Text Chunking module for dividing large documents
- Embedding Generator for semantic vector representation
- Vector Database (FAISS/Chroma) for storage and retrieval
- Retriever module for similarity-based search
- Large Language Model for response generation
- Streamlit Interface for user interaction

The proposed system is designed as a multi-stage pipeline integrating document processing, embedding generation, retrieval, and response generation.

This modular design ensures scalability and flexibility, allowing individual components to be updated or replaced without affecting the entire system. The integration between frontend and backend is achieved through API calls, ensuring seamless communication and efficient processing.

The system design emphasizes modularity and scalability to ensure efficient handling of large volumes of geoscience data. Each component in the architecture operates independently while maintaining seamless integration through well-defined interfaces. The frontend, developed using Streamlit, provides an interactive platform where users can upload documents and submit queries in natural language. The backend handles intensive processing tasks such as text extraction, embedding generation, and similarity search. This separation of concerns not only improves system performance but also simplifies maintenance and future upgrades. Additionally, the use of vector databases enables efficient storage and retrieval of high-dimensional embeddings, allowing the system to scale with increasing data size without significant performance degradation.

Another critical aspect of the system design is the optimization of query processing and response generation. To ensure low latency, the system employs precomputed embeddings and efficient indexing techniques within the vector database. When a query is received, the system performs a rapid similarity search to identify the most relevant document chunks, minimizing response time. The retrieved context is then passed to the Large Language Model, which generates coherent and contextually accurate responses. Furthermore, caching mechanisms and efficient API handling using HTTPX are incorporated to reduce redundant computations and improve system responsiveness. This design ensures that the system can deliver real-time, accurate, and scalable performance suitable for practical deployment.

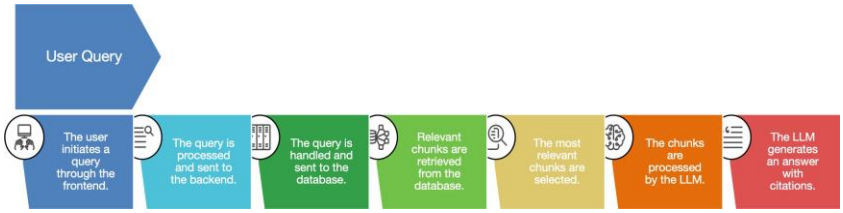

Fig. 1. *System Architecture of RAG-Based Geoscience Assistant*

System Design

The system architecture follows a modular design that separates the frontend interface from backend processing components. The frontend, developed using Streamlit, provides an intuitive interface for users to upload documents and enter queries. The backend handles document processing, embedding generation, retrieval, and response generation.

The workflow begins with user interaction through the interface, followed by document loading and preprocessing. The processed text is divided into chunks and converted into embeddings, which are stored in the vector database. When a query is received, it is embedded and compared with stored vectors to retrieve relevant information. The retrieved context is then passed to the LLM, which generates the final response displayed to the user.

Implementation

The implementation of the proposed system involves multiple stages, including data preprocessing, embedding generation, vector database integration, retrieval mechanism development, and user interface design. The system is developed using Python, leveraging libraries such as Streamlit for frontend development and HTTPX for API communication.

The document processing module extracts text from uploaded PDF files and divides it into smaller chunks. These chunks are then passed to the embedding model to generate vector representations. The embeddings are stored in a vector database, which enables efficient similarity-based retrieval.

The retrieval module is responsible for identifying relevant document chunks based on user queries. It computes the similarity between query embeddings and stored embeddings to retrieve the most relevant information. The response generation module uses a Large Language Model to generate answers based on the retrieved context.

The system is designed to handle real-time queries efficiently, providing quick and accurate responses. Proper error handling and data validation mechanisms are implemented to ensure system reliability.

The implementation of the proposed RAG-based Geoscience Assistant focuses on creating an efficient pipeline for document processing and query handling. The system begins by allowing users to upload PDF documents through the Streamlit interface. These documents are processed using text extraction techniques to convert them into raw textual format. The extracted text is then divided into smaller chunks using a recursive text-splitting approach, ensuring that each segment maintains semantic coherence. This chunking strategy improves retrieval accuracy by enabling fine-grained matching between user queries and document content.

The implementation process is carried out through the following stages:

- Document Upload and Processing: Users upload PDF files, which are parsed and converted into text format.
- Text Chunking: Large documents are divided into smaller chunks to improve retrieval precision.
- Embedding Generation: Each chunk is converted into vector embeddings using a pre-trained embedding model.
- Query Processing: User queries are converted into embeddings and matched with stored vectors.
- Response Generation: The retrieved context is passed to the LLM to generate the final response.

The embedding generation process plays a critical role in capturing the semantic meaning of textual data. Pre-trained embedding models are utilized to transform both document chunks and user queries into high-dimensional vector representations. These embeddings are stored in a vector database, which enables efficient similarity-based retrieval using indexing techniques. The use of cosine similarity ensures that semantically similar text segments are identified accurately, even if the exact keywords do not match.

The proposed system is structured into several functional modules, each responsible for a specific stage of the processing pipeline.

- User Interface Module: Handles document upload and query input
- Processing Module: Performs text extraction and chunking
- Embedding Module: Converts text into vector representations
- Retrieval Module: Searches for relevant document chunks
- Generation Module: Produces final responses using LLM

To ensure efficient communication with external services, the system utilizes HTTPX for handling API requests to the Large Language Model. This enables asynchronous and reliable communication, reducing latency and improving performance. Error handling mechanisms are implemented to manage API failures and ensure system stability. Additionally, caching techniques are employed to store frequently accessed results, minimizing redundant computations and enhancing response speed. The overall implementation is designed to be scalable, allowing the system to handle increasing data volumes and user queries without significant degradation in performance.

Results And Discussion

The performance of the RAG-based Geoscience Assistant is evaluated based on its ability to provide accurate and context-aware responses. The system demonstrates significant improvement in retrieval accuracy compared to traditional keyword-based search systems. By leveraging semantic embeddings, the system is able to understand the intent behind user queries and retrieve relevant information effectively.

The integration of retrieval and generation ensures that the responses are grounded in actual data, reducing the occurrence of hallucinated outputs. The system also provides faster response times due to efficient vector-based search mechanisms.

User-based evaluation indicates that the system is effective in handling complex geoscience queries, providing detailed and meaningful responses. The modular architecture and use of modern technologies make the system scalable and adaptable to different domains.

The performance of the proposed RAG-based Geoscience Assistant is evaluated based on its ability to provide accurate, relevant, and context-aware responses to user queries. Unlike traditional keyword-based search systems, the proposed approach leverages semantic embeddings and retrieval mechanisms, enabling it to understand the intent behind queries and retrieve meaningful information. The system demonstrates improved response quality, particularly for complex and domain-specific queries, where contextual understanding plays a crucial role.

To analyze the effectiveness of the system, multiple test queries were executed on a dataset of geoscience-related documents. The responses generated by the system were evaluated based on relevance, coherence, and completeness. It was observed that the integration of retrieval and generation significantly reduced irrelevant outputs and improved the overall accuracy of responses. The system was also able to handle variations in query phrasing effectively, demonstrating robustness in natural language understanding.

- Performance Analysis: The system shows considerable improvement in retrieval accuracy and response generation when compared to traditional approaches. The use of vector embeddings allows the system to identify semantically similar content even when exact keywords are not present. This leads to more precise and meaningful results.

Table 1. Performance Comparison of Systems

Feature	**Traditional Search**	**System**
Context Awareness	Low	High
Response Accuracy	Medium	High

Handling Complex Queries	Poor	Excellent
Response Quality	Low	High

- Qualitative Analysis: A qualitative evaluation was conducted by testing the system with different types of queries, including descriptive, analytical, and conceptual questions. The results indicate that the system provides more informative and contextually rich responses compared to traditional methods. For example, when asked domain-specific questions, the system retrieves relevant document segments and generates explanations that are both accurate and easy to understand.
- System Efficiency: The efficiency of the system is influenced by the performance of the vector database and the response time of the LLM. By utilizing efficient indexing techniques and precomputed embeddings, the system is able to perform rapid similarity searches, reducing query processing time. Additionally, caching mechanisms further enhance performance by minimizing redundant computations.
- Discussion: The results demonstrate that the proposed RAG-based system effectively addresses the limitations of traditional retrieval methods. The combination of semantic search and generative AI enables the system to provide accurate and context-aware responses. The modular design ensures scalability and adaptability, allowing the system to be extended to other domains beyond geoscience.

However, the system's performance is dependent on the quality of input documents and the effectiveness of the embedding model. In cases where the dataset lacks sufficient information, the system may produce less detailed responses. Despite these limitations, the overall performance of the system is significantly better than conventional approaches, making it a promising solution for knowledge-intensive applications.

Advantages And Limitations

The proposed system offers several advantages, including improved contextual understanding, reduced hallucination, and efficient handling of large datasets. The use of vector databases enables fast and accurate retrieval, while the integration of LLMs enhances response generation.

However, the system has certain limitations. Its performance depends on the quality and relevance of the input documents. Additionally, the reliance on external APIs for LLM processing may introduce latency and dependency issues. Computational overhead may also increase with large-scale datasets.

Another significant advantage of the proposed system is its flexibility and adaptability across different domains. Although the system is designed for geoscience applications, the underlying RAG architecture can be easily extended to other domains such as healthcare, legal systems, and education by simply updating the document corpus. This domain independence makes the system highly reusable and cost-effective. Additionally, the use of modular components such as embedding models, vector databases, and language models allows developers to upgrade or replace individual components without redesigning the entire system. This ensures long-term scalability and ease of maintenance, making the system suitable for real-world deployment.

Despite these advantages, certain limitations must be considered for practical implementation. The system heavily depends on the availability of high-quality and domain-relevant documents, as the accuracy of responses is directly influenced by the input data. In scenarios where the dataset is incomplete or lacks sufficient coverage, the system may generate partially correct or less informative responses. Furthermore, the reliance on external APIs for Large Language Model processing introduces potential issues such as latency, rate limits, and dependency on third-party services. These factors may affect system performance under heavy usage conditions and highlight the need for optimization techniques or local model deployment in future improvements.

Applications

The proposed RAG-based Geoscience Assistant can be applied in various domains that require efficient retrieval and interpretation of large volumes of textual data. In geoscience research, the system helps researchers and students quickly extract relevant information from extensive datasets such as geological reports, environmental studies, and research papers. By providing context-aware responses, the system reduces the time required for manual analysis and improves decision-making.

The system also plays a significant role in environmental monitoring and disaster management. It can assist organizations in analyzing climate data, identifying environmental patterns, and understanding the impact of natural phenomena. This makes it useful for applications such as climate change analysis, resource exploration, and risk assessment in disaster-prone areas.

The key applications of the system include:

- Geoscience Research Assistance: Enables quick retrieval of information from research papers and reports
- Climate and Environmental Analysis: Helps in understanding climate trends and environmental changes

- Disaster Management Support: Assists in analyzing data related to earthquakes, floods, and landslides
- Educational Support Systems: Acts as an intelligent assistant for students and educators
- Knowledge Management Systems: Organizes and retrieves domain-specific information efficiently

In addition to domain-specific uses, the system is highly adaptable and can be extended to other industries. For example, in healthcare, it can assist in retrieving medical literature; in legal systems, it can help analyze case documents; and in business analytics, it can support decision-making by extracting insights from reports. This flexibility highlights the scalability and practical relevance of the proposed system in real-world applications.

Limitations And Future Work

The proposed RAG-based Geoscience Assistant provides significant improvements in context-aware information retrieval; however, certain limitations must be considered. One of the primary limitations is the system's dependence on the quality and completeness of the input documents. If the dataset lacks sufficient or relevant information, the generated responses may be incomplete or less accurate. Additionally, the system relies on pre-trained embedding models and external Large Language Model APIs, which may introduce latency and dependency issues. This can affect performance, especially under high user load or limited network connectivity.

Another limitation is the computational overhead associated with embedding generation and similarity search in large datasets. As the volume of documents increases, storage and retrieval operations may require optimization to maintain efficiency. Furthermore, while the system reduces hallucination compared to standalone LLMs, it does not completely eliminate it, particularly in cases where retrieved context is limited or ambiguous.

The key limitations of the system include:

- Dependence on the quality and availability of input documents
- Latency due to API calls for Large Language Model processing
- Computational overhead for large-scale vector databases
- Limited handling of ambiguous or insufficient queries
- Absence of real-time data integration

Future work aims to address these limitations and enhance the overall performance of the system. One potential improvement is the integration of real-time geoscience data sources, enabling the system to provide up-to-date and dynamic responses. Additionally, advanced embedding techniques and fine-tuned domain-specific language models can be incorporated to improve accuracy and relevance.

The future enhancements of the system include:

- Integration with real-time data sources and APIs
- Implementation of domain-specific fine-tuned LLMs
- Support for multimodal data such as images and maps
- Optimization of vector database indexing for faster retrieval
- Deployment on cloud platforms for scalability and accessibility

Overall, addressing these limitations and implementing the proposed enhancements will further improve the robustness, scalability, and real-world applicability of the system.

Conclusion

The proposed RAG-based Geoscience Assistant presents an effective solution for context-aware information retrieval in the geoscience domain. By integrating semantic retrieval mechanisms with Large Language Models, the system overcomes the limitations of traditional keyword-based search systems and standalone generative models. The use of embeddings and vector databases enables efficient and accurate retrieval of relevant information, while the generative component ensures that responses are coherent, meaningful, and easy to understand.

The system demonstrates improved performance in handling complex and domain-specific queries by leveraging contextual information from retrieved documents. Its modular architecture, consisting of document processing, embedding generation, retrieval, and response generation components, ensures scalability and flexibility. Additionally, the implementation using modern technologies such as Python, Streamlit, and vector databases makes the system practical for real-world applications.

Although certain limitations exist, such as dependency on data quality and external APIs, the overall results indicate that the proposed system significantly enhances the efficiency and accuracy of knowledge retrieval. With future improvements such as real-time data

integration, advanced embedding techniques, and multimodal capabilities, the system has strong potential to be extended beyond geoscience into other knowledge-intensive domains. Overall, the RAG-based approach provides a robust and scalable framework for intelligent information retrieval and decision support systems.

Acknowledgment

The authors would like to express their sincere gratitude to the Department of Artificial Intelligence and Data Science at Dr. D. Y. Patil College of Engineering and Innovation, Pune, for providing the necessary support and resources to carry out this project. The guidance and encouragement from faculty members played a significant role in the successful completion of this work.

The authors also extend their appreciation to the project mentors and teaching staff for their valuable insights, continuous support, and constructive feedback throughout the development of the RAG-based Geoscience Assistant. Their expertise and direction greatly contributed to improving the quality and effectiveness of the system.

Additionally, the authors acknowledge the use of open-source libraries, frameworks, and online resources that facilitated the implementation of this project. The contributions of the research community and publicly available tools have been instrumental in enabling the development and evaluation of the proposed system.

References

1. P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” *Advances in Neural Information Processing Systems*, 2020.
2. J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *Proceedings of NAACL-HLT*, 2019.
3. T. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., “Language Models are Few-Shot Learners,” *Advances in Neural Information Processing Systems*, 2020.
4. V. Karpukhin, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W. Yih, “Dense Passage Retrieval for Open-Domain Question Answering,” *Proceedings of EMNLP*, 2020.
5. J. Johnson, M. Douze, and H. Jégou, “Billion-scale similarity search with GPUs,” *IEEE Transactions on Big Data*, 2019.
6. LangChain Documentation, “LangChain Framework for LLM Applications,” [Online]. Available: https://docs.langchain.com/
7. OpenAI, “OpenAI API Documentation,” [Online]. Available: https://platform.openai.com/docs/
8. J. Benet, “IPFS - Content Addressed, Versioned, P2P File System,” 2014. [Online]. Available: https://arxiv.org/abs/1407.3561