Enhancing Research Paper Knowledge Retrieval Using RAGs and LLMs

Ajay Rao
University of Illinois, Urbana Champaign
aarao3@illinois.edu

Jinang Gandhi
University of Illinois, Urbana Champaign
jinangg2@illinois.edu

Nikunj Agarwal
University of Illinois, Urbana Champaign
nikunja2@illinois.edu

Riya Tendulkar
University of Illinois, Urbana Champaign
rtend@illinois.edu

ACM Reference Format:
Ajay Rao, Jinang Gandhi, Nikunj Agarwal, and Riya Tendulkar. 2026. Enhancing Research Paper Knowledge Retrieval Using RAGs and LLMs. In . ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

This report presents the final outcomes of our project aimed at developing an advanced knowledge retrieval system for research papers. The motivation behind this work comes from the growing volume of scientific literature, which makes it increasingly challenging for researchers to efficiently locate relevant information. Our goal was to design and implement a system that enables semantic search and question answering over collections of research papers, leveraging natural language processing (NLP) and information retrieval (IR) techniques.

The main objectives of this project are as follows:

- **Develop a pipeline** for extracting and preprocessing textual data from research papers in PDF format.
- **Implement semantic chunking** to split documents into meaningful text units that preserve contextual integrity.
- **Generate high-quality vector embeddings** for text chunks using transformer-based models to capture semantic meaning.
- **Create and manage a scalable vector database** using Pinecone to store embeddings and enable efficient similarity search.
- **Design a retrieval mechanism** that accurately fetches relevant document chunks in response to user queries.
- **Integrate a retrieval-augmented generation (RAG) pipeline** that leverages large language models to generate concise, context-aware answers.
- **Develop a basic web-based user interface** using Flask to facilitate easy user interaction with the system.
- **Evaluate the system's performance** through both manual and automated metrics to ensure retrieval accuracy and answer quality.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

We utilized open-source frameworks such as LangChain for document processing, Hugging Face for embedding generation, Pinecone as a scalable vector database, and Flask for the user interface. Through iterative development and evaluation, we have built a pipeline that not only retrieves relevant document passages but also provides concise, context-aware answers to user queries.

This report details the complete methodology, presents evaluation and discusses insights gained, limitations encountered, and future scope.

2 DESCRIPTION

2.1 Methodology

The implementation of our research paper knowledge retrieval system is organized into two main components: **Uploading** and **Querying**. Each component consists of a sequence of steps to ensure that research papers are efficiently processed and that user queries are answered accurately and contextually.

2.1.1 Uploading. The uploading phase is responsible for adding new research papers to the system and preparing their content for semantic search and retrieval. The process consists of the following detailed steps:

- **Document Extraction:**
  Research papers in PDF format, are collected and processed and the text is extracted from each page of the document, and metadata such as document title and total page count are also captured. This ensures that each piece of information can be traced back to its source.
- **Text Cleaning:**
  The raw extracted text is pre-processed to remove noise, such as special characters, page numbers, headers, footers from the original document formatting. These are removed using pattern matching and regular expressions. The result is a clean, normalized version of the document text.
- **Section Segmentation:**
  The cleaned text is analyzed to identify and separate major sections of the research paper, such as Abstract, Introduction, Methodology, Results, Conclusion, and References. This is achieved by detecting common section headers, allowing the system to preserve the logical flow and context of the original document.
- **Chunking:**
  Each identified section is further divided into smaller, overlapping text chunks. This is done to ensure that each chunk

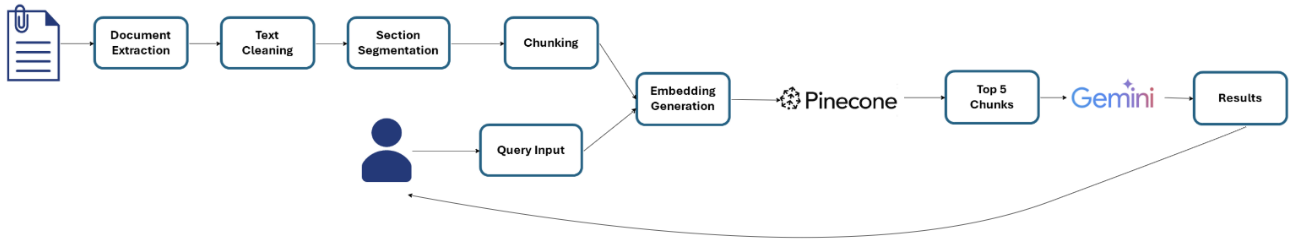

Figure 1: Methodology for Implementing the Retrieval System.

is of a manageable size for semantic processing and that important context is preserved across chunk boundaries. Overlapping chunks help maintain continuity of information, especially for content that spans multiple sections.

- **Embedding Generation:**
  Each text chunk is transformed into a high-dimensional semantic vector (embedding) that captures the meaning and context of the chunk. This enables the system to later compare and retrieve similar content based on meaning rather than just keywords.
- **Indexing:**
  The generated embeddings, along with their associated metadata (such as section, title, and source), are stored in a vector database. This database is optimized for fast similarity search, to ensure efficient retrieval of relevant content in response to user queries.

2.1.2 Querying. The querying phase allows users to interact with the system by submitting questions and receiving concise, contextually relevant answers. The steps involved are as follows:

- **Query Input:**
  Users enter their questions in natural language through the system interface. The system is designed to handle a wide range of research-related queries.
- **Query Embedding:**
  The submitted question is converted into a semantic vector using the same embedding approach as used for the document chunks. This ensures that the query and document content are represented in the same semantic space.
- **Similarity Search:**
  The system performs a similarity search in the vector database to find the document chunks whose embeddings are most similar to the query embedding. Typically, the top few (e.g., five) most relevant chunks are retrieved. This allows the system to identify the most contextually appropriate passages from the entire corpus.
- **Contextual Answer Generation:**
  The retrieved chunks are assembled and provided as supporting context to a language model. The language model synthesizes a concise answer to the user's question, drawing exclusively from the retrieved information. If the answer cannot be found in the context, the system is designed to acknowledge this and avoid speculation.
- **Response Delivery:**
  The final answer, along with references or excerpts from the supporting document chunks, is presented to the user. This ensures transparency, allowing users to trace the answer back to its source material and verify its accuracy.

2.2 Implementation Details

- **Vector Database:**
  The system utilizes **Pinecone** as the vector database for storing and searching semantic embeddings of research paper content. The index is configured for 384-dimensional vectors, matching the size of the embeddings generated. The database is deployed in a serverless configuration on AWS, ensuring scalability and low-latency retrieval. Document chunks are stored along with metadata such as section, title, and source file for traceability.
- **Similarity Measure:**
  Cosine similarity is used as the primary metric for comparing vector embeddings within Pinecone. When a user submits a query, its embedding is compared against all stored embeddings, and the top 5 most similar chunks (those with the highest cosine similarity) are retrieved. This approach ensures that retrieval is based on semantic meaning rather than mere keyword overlap.
- **Embedding Model:**
  Both document chunks and user queries are embedded using the transformer-based model (*sentence-transformers/all-MiniLM-L6-v2*) from Hugging Face. This model produces 384-dimensional embeddings that capture the semantic context of scientific and technical text. Embeddings are generated during both the uploading (for document chunks) and querying (for user queries) phases, ensuring consistency.
- **Chunking and Preprocessing:**
  Uploaded research papers are first extracted as plain text from PDF files using the `PyPDFLoader` module. The text is cleaned by removing special characters, headers, footers, page numbers, and stop words. The cleaned text is then segmented into logical sections (e.g., Introduction, Methodology, Results) by detecting common section headers through regular expression (regex) matching. Each section is subsequently split into overlapping chunks (default: 1000 characters per chunk with 40-character overlap) to preserve context across

boundaries. Each chunk is associated with metadata for traceability in retrieval and answer generation.

- **Retrieval Method:**
  The system uses a retriever built on top of the Pinecone vector store, configured to perform similarity search. For each query, the retriever returns the top 5 most similar document chunks, which are then used as context for answer generation.
- **Large Language Model (LLM):**
  The answer generation component uses **Gemini 1.5 Pro** (Google's state-of-the-art LLM). The model is accessed via the `ChatGoogleGenerativeAI` interface in LangChain, with the latest version (`gemini-1.5-pro-latest`) and a low temperature (0.3) for concise, factual responses. The LLM is prompted to answer based strictly on the retrieved context, and to admit when the answer is not present in the documents.
- **Retrieval-Augmented Generation (RAG) Pipeline:**
  The system follows a RAG approach: the retriever fetches relevant document chunks based on the query, and these chunks are passed, along with the user's question, to the LLM using a prompt template. The prompt instructs the LLM to use only the provided context and to keep the answer concise (maximum three sentences). The final answer and supporting document excerpts are returned to the user.
- **User Interface:**
  A basic **Flask** web application is provided for user interaction. Users can submit queries and upload new research papers through the UI. Upon querying, the system displays both the generated answer and the supporting document chunks, ensuring transparency and allowing users to verify the source of information.

2.3 Results

The results section demonstrates the functionality and user experience of the Research Paper Q&A Assistant. It highlights the intuitive interface for uploading research PDFs, querying content, and receiving well-referenced answers. The main interface consists of two primary sections: one for uploading research PDFs and another for asking questions. Figure 2 displays the main interface showcasing these elements.

2.3.1 Uploading a Research Paper. Users can upload any research paper in PDF format using the upload form as shown in Figure 3. Upon submission, the system indexes the document to enable efficient querying. This process is quick and takes less than 5 seconds to index a new research paper.

2.3.2 Querying and Answer Retrieval. Users can then proceed to query any research paper in the database, including the one recently uploaded. Below are the results of a few test cases done to check the retrieval of the system.

- **Summarizing a paper**
  The Research Paper Q&A Assistant showed it can understand the question well and provide a clear summary of the whole paper as shown in Figure 4. When asked about the

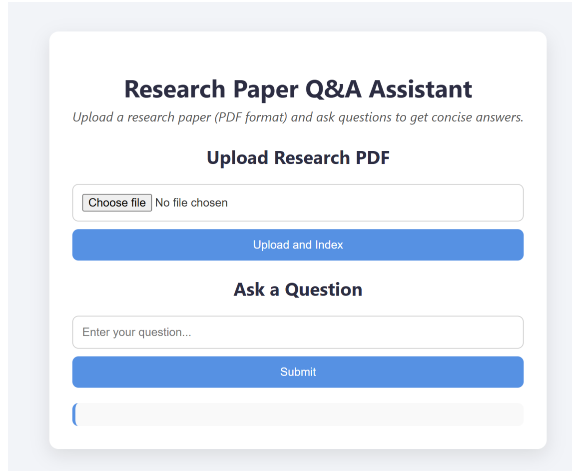

Figure 2: Main interface of the Research Paper Q&A Assistant

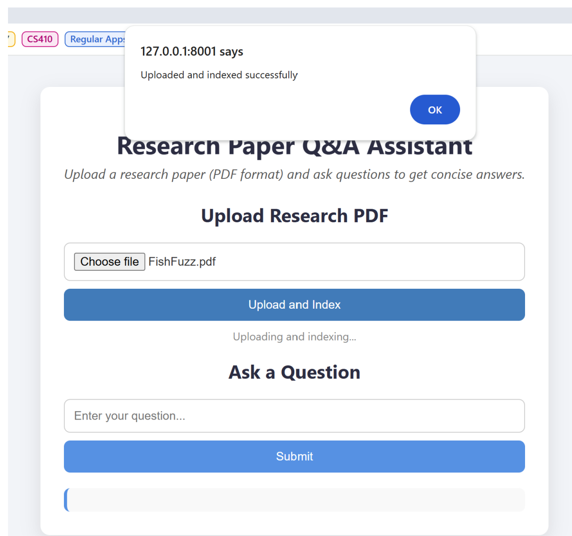

Figure 3: Uploading a PDF on the portal.

Needfinding Machine, it correctly picked out the main idea-highlighting its role in helping designers understand user needs through embedded, interactive systems. The response clearly described key features such as remote observation, real-time interaction, and the use of technologies like machine learning and adaptive interfaces. The answer was easy to follow and covered the important points, showing that the assistant can grasp the main message of a research paper and give a complete, concise response to the user's question.

- **Asking a specific question: Part 1**
  The Research Paper Q&A Assistant accurately interpreted the technical question about the Wizard-of-Oz method and provided a clear, focused explanation as shown in Figure 5. It correctly described the method as a way to simulate complex

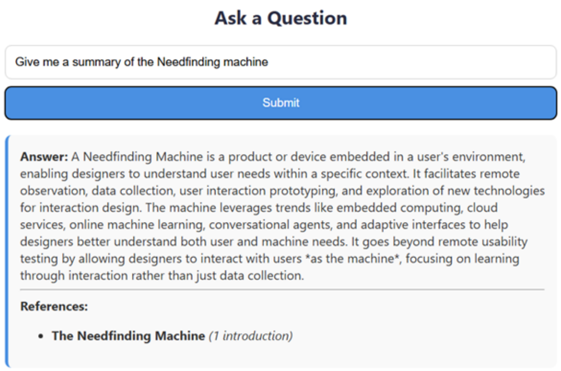

Figure 4: Summarizing a paper.

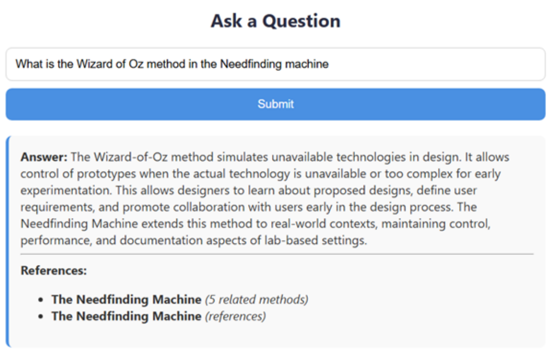

Figure 5: Asking the assistant a specific question from the research paper.

or unavailable technologies, emphasizing its role in early design testing, defining user needs, and supporting real-time user-designer interaction. The answer was well-structured and relevant, showing the assistant's ability to extract and communicate specific methodological details from the paper. Such interpretive capability is valuable for users seeking to quickly grasp nuanced aspects of advanced research works.

- **Asking a specific question: Part 2**
  The Research Paper Q&A Assistant effectively addressed a nuanced question about the Needfinding Machine's limitations and future directions, as shown in Figure 6. It identified key constraints, including the need for network access, power, and sensor infrastructure, which may limit deployment in low-resource environments. The assistant also pointed out the risk of overengineering—adding unnecessary features that could distract from core user needs—and stressed the importance of focusing on user experience rather than just functionality. This response reflects a strong grasp of both practical challenges and design philosophy, offering a well-rounded, insightful answer that helps users think critically about system improvement.

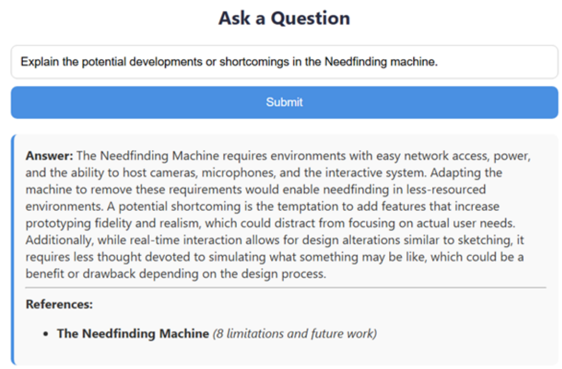

Figure 6: Asking the assistant a specific question from the research paper.

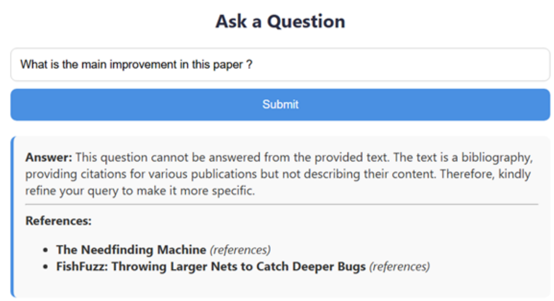

Figure 7: Asking the assistant a slightly ambiguous question.

- **Answering an ambigious question**
  The question shown in Figure 7—"What is the main improvement in this paper?"—is ambiguous due to the absence of clear context or a specified source. Unlike more detailed queries, this one does not indicate which paper is being referred to or what type of improvement—technical, methodological, or experiential—is of interest. Faced with these limitations, the Research Paper Q&A Assistant appropriately acknowledged that it could not answer the question based on the available text.
  This response highlights the assistant's capacity for responsible handling of uncertainty. Rather than generating a speculative or misleading answer, it accurately recognized the lack of actionable information and provided a clear explanation of why the question could not be addressed. In doing so, the assistant demonstrated sound judgment, reinforcing its reliability in academic and technical contexts. While this scenario underscores the importance of well-formed and contextualized user input, it also illustrates the assistant's ability to manage ambiguity by setting appropriate boundaries for its responses. This is a critical feature for any AI system intended to support rigorous, information-sensitive tasks such as academic research or design analysis.

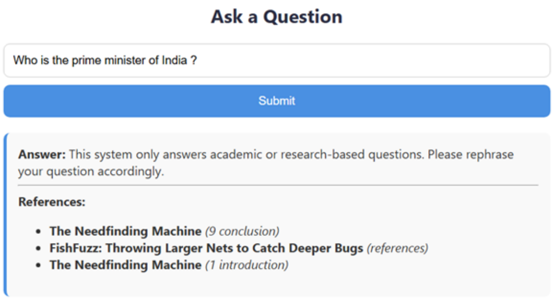

Figure 8: Asking the assistant an question which is not mentioned/related to the document.

- **Asking an unrelated question.**
  The Research Paper Q&A Assistant demonstrated its ability to recognize when a question falls outside the intended scope of the system or the provided material as shown in Firgure 8. When asked, "Who is the prime minister of India?"—a general knowledge question unrelated to the content of the Needfinding Machine paper—the assistant appropriately responded that it only answers academic or research-based questions. Rather than attempting to provide a factually correct but contextually irrelevant answer, the assistant set a clear boundary aligned with its intended use case.
  This response highlights an important capability: discerning and rejecting questions that are not only unsupported by the source text but also beyond the system's defined domain. By doing so, the assistant avoids misleading the user or straying from its academic focus. This behavior reinforces trust and ensures that the system remains a reliable tool for research-related inquiry, particularly in environments where the integrity and relevance of information are critical.

3 EVALUATION

The effectiveness of the Research Paper Q&A Assistant was evaluated through a combination of functionality testing, qualitative assessment, and user-oriented test cases.

3.1 Top-K Retrieval Evaluation

One of the key parameters affecting the performance of our retrieval-augmented system is the value of Top-K *(k)*, which determines how many documents to retreive. The results are summarizes in *Table 1.*

Case Study: ***"What is the purpose of Needfinding?"*** (*The Needfinding Machine*)

- With $k = 3$:
  The top three chunks included key definitions of needfinding, its purpose in the design process, and a brief mention of the Needfinding Machine framework. The retrieved content was highly focused and semantically tight. However, the context was not sufficiently broad, and some nuanced details (e.g., methods used, probes vs. prototypes) were missing.
- With $k = 5$:
  The system retrieved five well-aligned chunks covering the concept and its observational methods and the relationship between needfinding, probes, and prototypes. It provided a comprehensive context that covered both conceptual and practical aspects.
- With $k = 7$:
  Two additional chunks expanded into related areas like design ethnography and observation-action frameworks. These additions introduced contextual drift, slightly moving away from the central topic of needfinding. The response clarity and conciseness declined, and latency increased due to the larger context.

3.2 Cross-Domain Generalization Test

To evaluate the adaptability and robustness of our retrieval-augmented system across different academic disciplines, we conducted a Cross-Domain Generalization Test using research papers from three distinct fields: Environmental Science, Psychology, and Physics. For each paper, we submitted a domain-specific query and assessed whether the system could correctly retrieve and synthesize an accurate response from the document chunks.

3.2.1 Environmental Science: (*Global Warming Has Increased Global Economic Inequality*)

Query: *What years were studied to show high global inequality due to global warming?*
Retrieved Answer:

> The study analyzed the impact of global warming on economic inequality during two periods: 1961–2010 and 1991–2010. The longer period (1961–2010) showed a larger cumulative impact of warming on inequality.

Analysis:
The system accurately identified the specific time periods studied, indicating effective temporal reasoning within the scientific text. The inclusion of context around inequality ratios enhanced the depth of the response.

3.2.2 Psychology: (*The Role of Sensation Seeking in Political Violence*)

Query: *What is the main motive for sensation seeking?*
Retrieved Answer:

> Sensation seeking is driven by a search for novel and intense experiences. This can lead individuals to engage in risky behaviors and adhere to extreme ideologies. The search for meaning in life can also be a motivator.

Analysis:
The response captured the core psychological drivers behind sensation seeking, reflecting the system's ability to understand abstract concepts like motivation and meaning. The mention of extreme ideologies also aligned with the paper's context. Psychological constructs often involve nuanced and layered explanations. While the answer was accurate, it lacked references to theoretical frameworks

Table 1: Comparison of Top-K Retrieval Settings

Top-K (k)	Retrieved Concepts	Coverage Quality	Notes
$k = 3$	Definition of needfinding, early-stage design role	Moderate	Lacks depth and supporting nuance
$k = 5$	+ Needfinding Machine, probes vs. prototypes, user interaction	High	Best balance for RAG; production default
$k = 7$	+ Ethnography, observation loops, academic discussions	Very High	Context drift and redundancy

(e.g., Significance Quest Theory), which are crucial in psychological discourse. Incorporating more theory-aware retrieval could improve completeness.

3.2.3 Physics: (*Lattice-based Monte Carlo simulation of the effects of nutrient concentration and magnetic field exposure on yeast colony growth*)

Query: *What is the primary result of Lattice-based Monte Carlo simulation?*

Retrieved Answer:

> Magnetic fields decrease colony growth, solidity, and roundness, while increasing elongation and boundary fluctuations. Effects depend on nutrient concentration and ploidy. The simulation reproduced experimental results.

Analysis:

The answer successfully synthesized quantitative and qualitative impacts from the simulation. It showed the system's ability to handle complex causal relationships and experimental validation in scientific texts. But, physics and simulation-heavy papers often contain equations, graphs, and data tables, which are not fully represented in text chunks. As a result, numerical precision and modeling assumptions may be underrepresented in the retrieval and response.

3.3 Response Evaluation

The Gemini 1.5 Pro LLM demonstrated strong performance in synthesizing answers from the retrieved context. It provided clear, concise, and accurate responses that aligned with the content of the original document. Notably, when asked about nuanced technical distinctions—such as between Needfinding approaches presented a balanced, informative explanation without favoring one tool over the other. Furthermore, the assistant was able to synthesize abstract summaries (e.g., summarizing an entire paper) and respond to highly specific queries, showcasing both breadth and depth in contextual understanding.

3.4 User Experience

The web-based user interface, implemented using Flask, proved to be intuitive and responsive. Uploading a new research paper took less than 5 seconds, and queries returned answers with minimal latency. The display of both the generated answer and the supporting source chunks enhanced transparency, enabling users to verify responses. The system's design choices—such as chunk overlap and metadata tagging—contributed to a smooth user experience and high trust in answer provenance.

4 DISCUSSION

This project presented a comprehensive framework for building a research paper question-answering assistant using Large Language Models (LLMs). Throughout the development and evaluation process, we explored various categories of questions—such as factual, contextual, reasoning-based, and summarization—and evaluated our system's ability to retrieve accurate answers based on chunked representations of research articles.

Our evaluation highlighted that the system performs well on direct factual queries, with reasonably accurate retrievals and coherent summaries. However, questions that required reasoning across multiple sections or integration of information from non-textual elements such as figures or tables proved more challenging. This observation emphasizes the limitations of current chunking strategies and the absence of deeper document structure awareness.

From a real-world perspective, the system holds promising applications in academic and research workflows. It can serve as an intelligent assistant for:

- **Graduate students** performing literature reviews by surfacing key insights without reading entire papers.
- **Researchers** who wish to quickly compare methodologies or results across multiple works.
- **Educators** who want to generate comprehension questions or assist students with understanding complex topics.

These applications underline the system's potential to democratize access to dense academic content, especially in interdisciplinary fields or for non-native English speakers.

Nevertheless, the system has several limitations. As shown in our results and reflected in user interaction, the chunking granularity can often be misaligned with the semantic flow of the paper, leading to incomplete context or retrieval of irrelevant information. Moreover, the current system lacks any understanding of non-textual elements, such as figures, equations, or tables—which are often crucial for conveying experimental results or model architectures.

In conclusion, while our system demonstrates solid performance in answering textual questions from research papers, it also surfaces open challenges in long-document comprehension and knowledge extraction. Addressing these will be critical in advancing the system toward production-grade academic assistants.