RAG-based conversational tool for scientific materials
Final report
NLP Course, Winter 2024

Michał Gromadzki
Warsaw University of Technology
michal.gromadzki.stud@pw.edu.pl

Kacper Skonieczka
Warsaw University of Technology
kacper.skonieczka.stud@pw.edu.pl

Grzegorz Zakrzewski
Warsaw University of Technology
grzegorz.zakrzewski.stud@pw.edu.pl

Jakub Piwko
Warsaw University of Technology
jakub.piwko2.stud@pw.edu.pl

supervisor: Anna Wróblewska
Warsaw University of Technology
anna.wroblewska1@pw.edu.pl

Abstract

This project explores the usage of Large Language Models (LLMs) enhanced by Retrieval-Augmented Generation (RAG) to answer specific questions from user-provided scientific papers. The system generates accurate, context-aware responses supported by source citations by combining LLMs' language understanding with RAG's dynamic retrieval of relevant documents. This approach addresses the limitations of LLMs, such as outdated knowledge and hallucinations, ensuring reliable outputs. The novelty lies in tailoring the system to process user-defined collections of scientific articles, enabling domain-specific adaptability. The deliverable is a functional chatbot that simplifies accessing and understanding complex research materials, evaluated for accuracy and relevance. The chatbot performance is assessed with many embedding and large language models.

1 Introduction

Large Language Models (LLMs) have revolutionized fields like data science, demonstrating remarkable abilities in code generation and academic writing tasks. However, challenges like factual inaccuracies and hallucinations limit their effectiveness in domain-specific applications.

Retrieval-Augmented Generation (RAG) addresses these limitations by integrating LLMs with external knowledge bases. RAG retrieves relevant documents from a structured database using embeddings and indexing techniques, combining this retrieved information with user input to provide context-aware and reliable responses. Unlike traditional methods, RAG eliminates the need for retraining by dynamically updating the model's knowledge through retrieval, making it especially valuable for tasks in rapidly evolving domains.

This capability opens up transformative possibilities for automating manual tasks, such as searching for specific information across diverse materials. By improving accuracy, relevance, and adaptability, RAG systems enable applications like intelligent chatbots and tools that enhance research, business, and education productivity.

2 Scientific Goal

This project aims to create a chatbot powered by Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG) to help students and researchers find information in scientific materials such as papers and books and evaluate its correctness and faithfulness. By combining the static knowledge of LLMs with the dynamic retrieval of external sources, the chatbot will provide accurate and relevant answers to user questions.

The main challenge we aim to address is the difficulty of quickly and accurately locating specific information in extensive collections of scientific documents. Traditional searching methods can be time-consuming, and LLMs alone struggle with hallucinations and outdated knowledge. Using RAG, the chatbot will retrieve relevant documents from a database to generate reliable answers. It will also show the sources of information, making it easier for users to cite and review materials.

Our project focuses on scientific articles in Data Science, especially those relevant to our master's

theses. This allows us to test the system in a challenging environment with complex and cutting-edge content. The chatbot will also be designed to work with any set of user-provided documents, making it adaptable for various domains in the future.

Evaluation is a key part of the project. We will test the chatbot's answers' accuracy and reliability and compare them to results from different pre-trained LLMs. Automated tools like LLM-based metrics will help us measure correctness and faithfulness. These tests will show how well RAG improves the quality of answers and how effectively it handles complex topics in specialized fields.

This project demonstrates RAG's potential to improve people's access and use of information, enabling faster and more accurate research in academic and professional fields. At the same time, we are trying to incorporate the latest methods for assessing the quality of our solutions qualitatively and quantitatively.

3 Related Works

Large Language Models (LLMs) have rapidly emerged as transformative tools in data science and other disciplines, showcasing remarkable abilities in tasks like code generation, academic writing, and conversational systems. These advancements are underpinned by the massive datasets used for training, which allow LLMs to generalize across a broad range of topics. As a result, LLMs have already demonstrated significant potential in diverse fields such as telecommunications (Zhou et al., 2024), biomedicine (Pal et al., 2024), and many others. Their impact on research has been profound, enabling new methods of inquiry and accelerating innovation across multiple domains (Antu et al., 2023; Sallam, 2023).

Despite their versatility, LLMs face notable challenges and limitations that hinder their application in more specialized domains. Among the most critical issues are their propensity to produce hallucinations (Huang et al., 2023; Rawte et al., 2023)—fabricated or inaccurate information—and their lack of specialized domain knowledge (Abu-Rasheed et al., 2024). These shortcomings make it difficult for LLMs to provide reliable outputs in contexts where accuracy and domain-specific expertise are paramount (Friha et al., 2024; Hadi et al., 2023; Hadi et al., 2024; Minaee et al., 2024). Addressing these issues has become a central focus of ongoing research.

Retrieval-Augmented Generation (RAG) has emerged as a promising solution to extend the capabilities of LLMs by integrating external knowledge sources. Instead of relying solely on the model's static parametric knowledge, RAG dynamically retrieves relevant information from structured databases, allowing the model to provide context-aware and up-to-date responses (Gao et al., 2023; **?**). The RAG architecture involves creating a vectorized representation of documents using embeddings (Almeida and Xexéo, 2019; Liu et al., 2020), storing these vectors in an efficient indexing structure, and leveraging similarity search to retrieve the most relevant documents based on user queries. These retrieved documents are concatenated with the user input and passed as context to the LLM, enabling it to generate responses grounded in factual evidence.

The flexibility and adaptability of RAG make it highly effective for applications where knowledge evolves rapidly. By eliminating the need for retraining, RAG enables LLMs to incorporate new information seamlessly, which is particularly advantageous for real-time applications. As a result, RAG has been applied successfully in various fields, including business intelligence (Arslan and Cruz, 2024), typo correction (Cho et al., 2024), and scientific literature analysis (Singh, 2023; **?**). One of its most impactful applications has been in the development of conversational agents or chatbots (Kulkarni et al., 2024; Vakayil et al., 2024), where RAG addresses key challenges like hallucination and improves performance in dynamic, specialized domains.

Evaluating RAG-based systems, especially in the context of chatbots, poses unique challenges. The correctness and faithfulness of responses depend heavily on the quality of the retrieval component and the LLM's ability to integrate the retrieved content effectively. Current evaluation methods for RAG systems focus on two primary areas: embeddings and end-to-end performance. Techniques such as hit rate, Normalized Discounted Cumulative Gain (nDCG), and Mean Average Precision (MAP) are widely used to assess the retrieval quality of embeddings (Zhou, 2024; Caspari et al., 2024). These metrics ensure that the embeddings accurately capture the semantic meaning of documents and retrieve relevant information effectively. Additionally, some novel

approaches involve using LLMs to evaluate RAG outputs, leveraging metrics like contextual precision, recall, and faithfulness to assess the overall response quality (Salemi and Zamani, 2024; Shankar et al., 2024; Zheng et al., 2023).

The growing interest in RAG systems has paved the way for numerous practical applications that automate labor-intensive tasks, such as searching for specific information across large document corpora. To our knowledge, no currently available tool serves as a literature assistant for researchers. Our tool will dynamically adapt to user article databases, making it a multi-domain system. We will also provide an evaluation strategy to test the quality of responses and the search engine. This ensures that our solution is reliable.

4 Methodology

Our project adopts a technical and experimental approach to evaluate the effectiveness of retrieval-augmented generation (RAG) systems for scientific literature retrieval and question-answering. The objective is to build a functional application that retrieves relevant content from scientific papers and generates accurate and coherent responses. This section outlines the underlying scientific methodology, methods, techniques, tools, and resources used in our project.

The guiding principle of our project is to leverage RAG to address the challenges of working with domain-specific knowledge, such as scientific literature. Our approach involves constructing a database of articles related to our master's theses, designing an efficient retrieval pipeline, and implementing a chatbot for response generation. Our methodology incorporates various techniques and tools to implement the RAG system effectively.

Dataset. A critical component of this project is the dataset. Instead of relying on publicly available datasets, we will construct our own database of scientific articles. These articles will primarily consist of papers collected during the preparation of our master's theses, ensuring the dataset reflects the latest knowledge in the field of data science. This approach provides a challenging benchmark for our chatbot and ensures that we are familiar with the content, enabling more effective quality assessment and evaluation of the chatbot's responses.

This collection expanded as we discovered more relevant articles during the semester. It has grown from around 40 articles at the beginning of the project to 75 papers. This dynamic growth aligns perfectly with the capabilities of Retrieval-Augmented Generation (RAG), which is designed for knowledge bases that evolve continuously. This setup served as an excellent test case, simulating real-world scenarios where the knowledge base must adapt to changing information needs.

Eventually we collected 75 scientific papers. They will be available through course repository.

Technological tools

- **Frontend:** `Streamlit`, a Python-based framework, to provide an intuitive user interface.
- **Backend:** `LangChain`, for modular integration of LLMs, embeddings, and retrieval components.
- **LLM Serving:** `Ollama`, framework designed to facilitate the deployment of large language models on local environments
- **Vector Storage:** `FAISS`, for storing and retrieving document embeddings efficiently.
- **Evaluation:** The `DeepEval` package, to measure response quality.

Techniques

- **Embedding Generation:** Converting textual documents into dense numerical vectors using embedding models.
- **Document Retrieval:** Leveraging FAISS (Facebook AI Similarity Search) for fast and scalable vector search.
- **Response Generation:** Combining retrieved documents with user queries to produce coherent outputs using LLMs.
- **Evaluation Pipeline:** Iteratively testing different configurations of embeddings and LLMs using predefined metrics.

To evaluate the components of our chatbot, we adopt a multi-step evaluation process focusing on embeddings, LLMs, and retrieval mechanisms. We analyze the results of our system using both quantitative and qualitative methods. Quantitative metrics such as hit rate, MMR, nDCG, and

MAP measure the retrieval component's effectiveness, while DeepEval metrics assess the quality of generated responses. Qualitative analysis involves manually reviewing chatbot responses to ensure relevance, coherence, and factual accuracy. Comparative analysis will demonstrate improvements in RAG-enhanced responses over baseline LLM outputs.

Embedding Models We test multiple embedding models to determine their effectiveness in retrieving relevant information. The embedding models under consideration include:

- Ollama - `snowflake-arctic-embed`
- Ollama - `mxbai-embed-large`
- Ollama - `nomic-embed-text`
- HuggingFace - `instructor-xl`

The embeddings will be evaluated using the following metrics:

- **Hit Rate:** The percentage of queries where the retrieved documents contain the correct answer.
- **Maximal Marginal Relevance (MMR):** Measures the diversity and relevance of retrieved documents.
- **Normalized Discounted Cumulative Gain (nDCG):** Evaluates the ranking quality of retrieved documents.
- **Mean Average Precision (MAP):** Summarizes retrieval precision across queries.

Language Models We test several cutting-edge LLMs for their ability to generate coherent and accurate responses. We will focus on models provided in the Ollama library, namely:

- `qwen2.5:3b`
- `qwen2.5:7b-instruct-q4_0`
- `llama3.1:3b`
- `llama3.2:8b`

Each model is evaluated using the `DeepEval` package, which provides a framework for obtaining qualitative and quantitative assessments of responses of one LLM provided by a different llm. We will try to focus on the following metrics:

- **Answer Relevancy Metric:** Evaluates the relevance of responses to queries.
- **Faithfulness Metric:** Measures factual accuracy against source materials.
- **Contextual Precision and Recall:** Assesses precision and recall of responses in the context of retrieved documents.
- **Contextual Relevancy:** Detects unsupported or fabricated content in responses.

5 SciBot

In this section, we want to present details about the implementation of Scibot itself. SciBot is a chatbot designed for advanced question-answering tasks, integrating contextual retrieval to provide concise and accurate responses. The core functionalities of SciBot are outlined below:

Initialization SciBot employs the `qwen2.5:7b-instruct-q4_0` language model for conversational tasks and session-based chat history management to maintain context across interactions.

Query Contextualization User queries are reformulated into standalone questions using a system prompt. This ensures that queries can be understood and processed independently of prior chat history, improving the relevance of retrieved results.

Database Ingestion SciBot processes pre-indexed document databases using `FAISS`, a high-performance vector store. Text embeddings are generated using the `HuggingFace InstructXL` model, enabling similarity-based search. The retriever operates with Maximum Marginal Relevance (MMR) to balance relevance and diversity. We provided a script that allows the user to feed any paper in a PDF format to a potential database, making SciBot very elastic and open to any domain articles.

Retrieval and Question Answering A history-aware retriever combines chat history with the reformulated query to fetch contextually relevant document chunks. The retrieved content is passed to the question-answering chain, guided by a concise system prompt. SciBot transparently states that it does not know the answer if insufficient information is available.

Conversational RAG Chain The chatbot employs a Retrieval-Augmented Generation (RAG) chain, combining retrieval and generation. This chain is enhanced with session-specific chat histories to ensure continuity in multi-turn conversations.

Session Management and Interaction Each session is identified by a unique ID, allowing SciBot to store and retrieve chat histories for seamless interactions. Users interact by querying through the `ask` method, which invokes the RAG chain and returns concise, contextually accurate responses.

SciBot's architecture ensures robust performance in applications requiring high-quality retrieval and concise responses.

5.1 Application

As part of our work, we also developed a user-friendly graphical interface using Streamlit, which serves as a chatbot application powered by SciBot. This interface is a personal AI assistant, allowing users to interact with SciBot and ask questions about scientific articles. The chatbot leverages the underlying vector store, built from embeddings generated by the `Instructor-xl` model from Hugging Face, and uses the LLM model `qwen2.5:7b-instruct-q4_0` for generating responses.

The interface is designed to be simple yet effective, providing users with a convenient way to interact with SciBot. It hides all backend complexities, allowing users to focus solely on their queries. Figure 7 shows an example of the interface, displaying a short conversation with the chatbot.

This application is tailored for users who prefer working with ready-to-use tools, offering an experience similar to many popular chat systems on the market. Thanks to the addition of history awareness, the chatbot can delve deeper into user questions, ensuring a more coherent and contextually rich conversation about scientific articles. The simplicity of the design does not compromise its functionality, making it an effective tool for research and inquiry.

6 Experiments

This section outlines conducted experiments. We describe the technical setup used in the development of the solution, detailing the techniques employed. Following that, we will present the results of our studies, offering an analysis and conclusions based on these outcomes.

6.1 Embeddings Evaluation

Approach To create embeddings, we first prepared our PDF files. This process included reading the files using specialized parsers and splitting them into smaller, manageable chunks suitable for large language models (LLMs). Academic papers are often extensive documents that can easily exceed the context window limits of such models. We opted for a chunk size of 384 tokens with an overlap of 96 tokens. The overlap assures that the information from the end of one chunk carries into the next one. Simultaneously, we saved metadata for each chunk to maintain a precise mapping between the chunks and the original documents, ensuring traceability to the source information. Once the chunks were prepared, we generated embeddings for them.

We compared four different embedding models:

- Ollama - `snowflake-arctic-embed`
- Ollama - `mxbai-embed-large`
- Ollama - `nomic-embed-text`
- HuggingFace - `instructor-xl`

The chunks were processed using each of these models, and we created vector stores using FAISS for fast indexing and similarity search. As a result, we obtained four distinct vector stores.

For the evaluation, we needed queries to compare against the retrieved content from the vector stores. To generate these, we randomly sampled 100 chunks from the papers and used the LLM model `llama3.1:8b` to rephrase their content. This approach provided altered fragments of the documents to serve as test queries for the vector stores's accuracy. We also saved metadata for each modified chunk, including its index and context within the original document, to trace back to the source. After generating these queries, we embedded them using each embedding model and retrieved results from the respective vector stores. The higher (on the list of retrieved chunks) the original chunk used to create the query, the better the model's ability to retrieve relevant information. We evaluated the performance of each embedding model using the following four metrics:

- **Hit Rate** - Measures whether the correct document is included in the retrieved results.
- **Mean Reciprocal Rank (MRR)** - Assesses the rank of the correct document, rewarding higher placements.
- **Normalized Discounted Cumulative Gain (nDCG)** - Evaluates the quality of rankings, considering both the position and relevance of retrieved documents.
- **Mean Average Precision (MAP)** - Measures precision at each rank where a correct document is retrieved, averaged across all queries.

Results and Discussion All four metrics were calculated for each embedding model using the modified document queries. The results are presented as bar plots, summarizing the performance of each model for each metric. These plots are shown in Figure 10 in the Appendix.

The analysis reveals that Snowflake achieves the highest hit rate, closely followed by Mxbai. InstructorXL demonstrates slightly lower performance, while Nomic shows the lowest hit rate among the models. This indicates that Snowflake and Mxbai are more reliable in ensuring relevant documents are included in the retrieval results.

On the MRR metric, Snowflake again leads, highlighting its ability to rank relevant documents higher. Mxbai and InstructorXL perform comparably but lag behind Snowflake, while Nomic performs significantly worse, indicating challenges in prioritizing relevant documents.

For nDCG, which considers the position and relevance of retrieved documents, Snowflake continues to lead, showcasing its effectiveness in producing high-quality rankings. Mxbai follows, albeit with a noticeable performance gap. InstructorXL demonstrates results comparable to Mxbai, while Nomic performs the worst, suggesting limited robustness in its ranking methodology.

Snowflake once again outperforms the other models in terms of MAP, reflecting superior precision across its rankings. Mxbai and InstructorXL show moderate performance, with InstructorXL slightly ahead. Nomic, however, ranks the lowest, indicating difficulty maintaining precision throughout its rankings.

Snowflake consistently outperforms the other embedding models across all four metrics, making it the most reliable and practical choice for retrieving and ranking relevant documents. Mxbai and InstructorXL perform moderately well, with InstructorXL showing slight improvements in MAP and nDCG over Mxbai. However, Nomic demonstrates the weakest performance, indicating significant retrieval accuracy and ranking quality limitations. Therefore, Snowflake is the most suitable embedding model for applications requiring high retrieval and ranking performance. Mxbai and InstructorXL may still be viable options for specific use cases, whereas Nomic appears less appropriate, given its relatively poor results.

6.2 Chatbot Evaluation

Approach The second part of our evaluation process focused on the SciBot chatbot itself. We tested the version that will be included in the final application, as described in Section 5. This version uses a vectorstore based on the `Instructor-XL` model. We evaluated several large language models integrated with the chatbot, specifically:

- `qwen2.5:3b`
- `qwen2.5:7b-instruct-q4_0`
- `llama3.1:3b`
- `llama3.2:8b`

To assess these models, we created a set of 60 diverse questions targeting specific topics covered in the scientific papers within our database. These questions ranged from inquiries about particular methods described in the papers to questions about general concepts, advantages, and disadvantages of approaches discussed in the literature. The complete list of questions is available in the project repository.

Manually evaluating the responses to such a large number of questions would have been highly time-intensive. To streamline the process, we used the `deepeval` package, which facilitates automated evaluation of retrieval-augmented generation (RAG) systems. This package employs an evaluator LLM to score responses based on input (question), actual output (model response), expected output (ground truth), and context (retrieved paper chunks). In addition to scores, the evaluator provides textual justifications for the ratings.

For each model and question, we saved the response, response time, and retrieved-context (six paper chunks). We employed the powerful `GPT-4o-mini` model to generate ground truth responses. We provided `GPT-4o-mini` with the questions and their corresponding contexts, enabling it to generate high-quality expected answers. Its strong performance in generating accurate responses made it a reliable benchmark for evaluating SciBot's outputs.

We assessed the chatbot's performance using five key metrics using `GPT-4o-mini` as an evaluator:

- **Answer Relevancy Metric** – Evaluates how well the response aligns with the question.
- **Faithfulness Metric** – Measures factual consistency with the provided context.
- **Contextual Precision** – Assesses the proportion of relevant information in the response.
- **Contextual Recall** – Evaluates the chatbot's ability to extract relevant information from the context.
- **Contextual Relevancy** – Combines precision and recall to assess overall contextual alignment.

These metrics collectively evaluate the chatbot's responses' accuracy, contextual relevance, and factual consistency.

Results and Discussion We analyzed the quantitative performance of different LLMs using bar plots to summarize the scores for each metric.

- Answer Relevancy presented in the Figure 9: Qwen2.5:7b performed best with a score of 0.94, followed closely by LLaMA3.1:8b at 0.95. LLaMA3.2:3b and Qwen2.5:3b showed slightly lower performance, indicating they provide relevant answers but are less consistent in maintaining high relevance.
- Faithfulness depicted in the Figure 11: Qwen2.5:7b led with 0.84, reflecting strong factual accuracy. LLaMA3.1:8b and LLaMA3.2:3b scored 0.82 and 0.80, respectively, while Qwen2.5:3b trailed at 0.79.
- Contextual Precision whown in the Figure 12: Qwen2.5:7b excelled with 0.79, slightly outperforming LLaMA3.1:8b, LLaMA3.2:3b, and Qwen2.5:3b, which scored between 0.75 and 0.76.
- Contextual Recall presented in the Figure 13: LLaMA3.2:3b achieved the highest score at 0.96, with Qwen2.5:7b and LLaMA3.1:8b close behind at 0.94. Qwen2.5:3b scored slightly lower at 0.92.
- Contextual Relevancy depicted in the Figure 14: Qwen2.5:7b outperformed other models with a score of 0.56, followed by LLaMA3.1:8b and LLaMA3.2:3b at 0.53. Qwen2.5:3b ranked lowest at 0.52.

The analysis highlights Qwen2.5:7b as the most robust model across all metrics, excelling in Faithfulness, Contextual Precision, and Contextual Relevancy. LLaMA3.1:8b and LLaMA3.2:3b also performed well, particularly in Contextual Recall. Qwen2.5:3b demonstrated reasonable performance but consistently ranked lower.

These results offer valuable insights into the strengths and weaknesses of each model, guiding the selection of the most suitable chatbot model for specific applications.

Case Study In addition to conducting a quantitative evaluation of the chatbot, we also reviewed its responses to assess its sensibility. This section presents examples of our chatbot's responses, the corresponding ground truth answers, and textual assessments generated by the `deepeval` package to demonstrate its capabilities.

Given the extensive number of test questions and the length of some responses, we decided to showcase only a selection of the most interesting cases. To maintain clarity and conciseness, we excluded the retrieved context from these examples, as it tends to be lengthy. However, all data, including responses, retrieved-context, and quality assessments provided by the evaluator LLM, are available in our repository.

In Figure 1, we present an example of a question, the chatbot's response, and the ground truth. Both responses address the same topic, focusing on the key aspects of the model architecture described in the query. This was a very specific question, as it pertained to a particular model from one of the papers in our database. Nevertheless, our

Question 1: What is the model architecture in the deep learning asset pricing model made by Luyang Chen?

Responding model: `llama3.1:8b`

Model Response:
The paper "Deep Learning in Asset Pricing" by Luyang Chen, Markus Pelger, and Jason Zhu uses a deep neural network architecture that combines an adversarial approach to construct informative test assets with a fully flexible form that accounts for time variation. The key innovations include using the fundamental no-arbitrage condition as the criterion function.

Ground Truth:
The deep learning asset pricing model constructed by Luyang Chen and colleagues utilizes a deep neural network architecture that incorporates a flexible non-linear factor model. This model aims to estimate stock returns and takes into account various conditioning information while accounting for time variation. The innovations of their approach include the integration of the fundamental no-arbitrage condition in the criterion function, the use of an adversarial approach to construct informative test assets, and the extraction of economic states from macroeconomic time series. The model is designed to outperform traditional methods in terms of Sharpe ratio, explained variation, and pricing errors, ultimately identifying key factors that drive asset prices.

Figure 1: First example of question and comparison of chatbot response and ground truth

Question 1 - Automatic evaluation

Answer Relevancy Score: 1.0
Justification: The score is 1.00 because the response directly addresses the question about the equivalent of a factor model with structural breaks in factor loading without any irrelevant statements.

Faithulness Score: 1.0
Justification: The score is 1.00 because there are no contradictions present, indicating that the actual output perfectly aligns with the retrieval context.

Contextual Precision Score: 1.0
Justification: The score is 1.00 because the relevant node ranked first provides a clear and direct answer to the query, stating that 'A factor model with a break in its factor loadings is observationally equivalent to a model without changes in the loadings but with a change in the variance of its factors.' In contrast, the subsequent nodes ranked second to sixth, discuss unrelated topics, such as standard factor models and statistical data, which do not address the question about structural breaks, thus reinforcing the relevance of the first node.

Contextual Recall Score: 1.0
Justification: The score is 1.00 because every aspect of the expected output is fully supported by the relevant node(s) in the retrieval context, effectively linking all key concepts together.

Contextual Relevancy Score: 0.67
Justification: The score is 0.67 because while there are relevant statements like 'A factor model with a break in its factor loadings is observationally equivalent to a model without changes in the loadings but with a change in the variance of its factors,' the retrieval context also contains irrelevant information such as 'goodness-of-fit assessment tools' that detracts from the specific question about equivalence.

Figure 2: Automatic evaluation scores and justification for first example question

chatbot successfully provided a correct and relevant answer.

The automatic evaluator confirmed this by assigning perfect scores (1.0) for answer relevancy, faithfulness, contextual precision, and contextual recall. These scores indicate that the context was well-retrieved and utilized, with no irrelevant information. However, the contextual relevancy score was slightly lower (0.67). This discrepancy arose because some parts of the retrieved context were not directly related to the question, as the information was concentrated in a specific section of one paper. This is an expected limitation when dealing with highly specific queries.

Overall, most responses to our test questions achieved similarly high scores. These results align with our quantitative assessment, further validating that SciBot can retrieve accurate context and provide informative responses to user queries.

Although most of the chatbot's responses were accurate, a few cases presented more significant challenges, which are particularly interesting for analysis. In the example given in Figure 3, the answer provided by SciBot does not align with the ground truth. The response introduces two completely different ideas related to the abbreviation "GGA." Unfortunately, the chatbot makes an error here, as the ground truth correctly identifies "GGA" as referring to a genetic algorithm with genetic engineering (GGA), which is unrelated to the AAC domain mentioned in the question.

Looking at the automatic evaluation in Figure 4, we can see that the evaluator detected this inconsistency, noting that the content of the chatbot's response is largely irrelevant to the question. While the evaluator assigned a relatively high score for "contextual precision" and "contextual recall," the "answer relevancy" score was 0.88, and the "faithfulness" score was 0.86, reflecting the chatbot's failure to provide an accurate answer. These scores may be considered somewhat lenient, given that the chatbot's response was significantly off-topic.

One interesting observation is that the retrieved context was positively assessed, as shown in the evaluation. The model indicated that the context contained relevant papers with the necessary information. This suggests that the issue lies not in the retrieval process but in SciBot's ability to utilize the retrieved context effectively. The chatbot failed to identify the key idea behind the question

Question 2: What is the idea behind the AAC approach called "GGA"?

Responding model: `qwen2.5:7b`

Model Response:
"The term "GGA" in the context of Augmentative and Alternative Communication (AAC) likely refers to a specific method or strategy within AAC that focuses on Goal, Group, and Action. However, it's important to note that there isn't a widely recognized standard abbreviation called "GGA" for this exact approach.
If you're referring to a specific framework or methodology developed by a particular organization or researcher, here's what the components might typically represent...

Ground Truth:
The GGA (Genetic Algorithm with Genetic Engineering) approach, as described in the context, involves enhancing the traditional genetic algorithm by integrating a surrogate model to evaluate the quality of new configurations. The surrogate model predicts the rank an individual would achieve in a tournament rather than directly predicting solution quality or runtime, which is important in finding high-quality configurations. GGA employs a competitive and non-competitive population structure, where successful configurations from competitions are recombined through a crossover operator. This adaptation aims to focus on high-performing individuals, utilizing specialized splitting criteria within a random forest model to refine the search process in automatic algorithm configuration.

Figure 3: Second example of question and comparison of chatbot response and ground truth

Question 2 - Automatic evaluation

Answer Relevancy Score: 0.88
Justification: The score is 0.88 because while the response mostly addresses the AAC approach, the irrelevant statement about 'GGA' not being a widely recognized standard detracts from the clarity and focus on the main concept.

Faithulness Score: 0.86
Justification: The score is 0.86 because the actual output incorrectly asserts the existence of a widely recognized standard abbreviation 'GGA', which is not mentioned in the retrieval context.

Contextual Precision Score: 1.0
Justification: The score is 1.00 because the relevant nodes ranked highest provide direct information about the GGA approach, stating that 'GGA to include a surrogate model' and detailing its integration with a crossover operator. The irrelevant nodes ranked lower discuss topics unrelated to GGA, such as GMM equations, variations of GA, and Generative Adversarial Networks, ensuring that they do not interfere with the relevance of the first node.

Contextual Recall Score: 1.0
Justification: The score is 1.00 because every element of the expected output directly aligns with the information provided in node 1 of the retrieval context, demonstrating a perfect match in concepts related to the GGA approach.

Contextual Relevancy Score: 0.29
Justification: The score is 0.29 because while there are relevant statements about GGA, such as its adaptation in 'GGA++' and its population structure, the majority of the context focuses on unrelated AAC methods and approaches, which does not address the core idea behind GGA.

Figure 4: Automatic evaluation scores and justification for the second example question

and, as a result, provided an inaccurate response despite the availability of relevant data.

The question in Figure 5 illustrates another example of SciBot's failure to answer correctly. In this case, the question asked about MSE SHAP. This specific metric calculates the Mean Squared Error (MSE) on previously calculated SHAP values to evaluate the explainability of machine learning models. However, SciBot's response is not accurate. The model starts with "I don't know," indicating that it could not retrieve the relevant context to address the question. This suggests that the issue lies not with the language model but with the retrieval process (RAG), which failed to fetch the correct context.

This issue is confirmed by the automatic evaluation, as seen in Figure 6. The evaluator scores the contextual metrics very low, providing justifications highlighting how the retrieved context discusses MSE and SHAP metrics separately but not in the combined context of "MSE SHAP." This failure to retrieve the correct context meant that the ground truth model could not correctly identify the definition of the "MSE SHAP" metric and instead described MSE and SHAP individually.

This case serves as a clear example of errors in the RAG process that can harm the rest of the SciBot pipeline. Without the correct context, it becomes challenging, if not impossible, for the bot to answer the user's query accurately, even when the information is present in the database.

7 Work Division

This section provides an overview of the tasks completed, along with a breakdown of work distribution among team members. We are pleased to report that the project's schedule was fulfilled. The timeline outlined in the project proposal has been successfully followed, with some minor adjustments incorporated. Below, we detail the tasks completed until now, their respective timelines, and the members responsible for each.

- **Task 1:** Conduct a literature review on state-of-the-art RAG systems and their applications. **Timeline:** November 2024 **Assignees:** Jakub, Michał, Grzegorz

- **Task 2:** Collect a dataset of scientific papers. **Timeline:** Nov 20 - Nov 24, 2024 **Assignees:** Grzegorz, Michał

Question 3: What is MSE SHAP?

Responding model: `llama3.2:3b`

Model Response:
I don't know. The provided context does not mention MSE SHAP specifically, but it mentions Mean Squared Error (MSE). It also discusses SHAP values and their application in feature importance measures.

Ground Truth:
"The term "MSE SHAP" does not appear in the provided documents specifically. However, MSE (Mean Squared Error) is a common metric used to measure the accuracy of predictions in a model, calculated as the average of the squared differences between predicted and actual values. SHAP (SHapley Additive exPlanation) values, on the other hand, are a measure of feature importance that explains the contribution of each feature to the model's predictions based on game-theoretic principles.
While there is no direct mention of "MSE SHAP" in the snippets, it could imply a context where both MSE (or other error metrics) are being evaluated alongside SHAP values to provide insights into model performance and feature contributions. In practice, researchers may analyze how different features (using SHAP) influence the MSE or other error metrics, but a specific definition or methodology termed "MSE SHAP" is not delineated in the provided excerpts."

Figure 5: Third example of question and comparison of chatbot response and ground truth

Question 3 - Automatic evaluation

Answer Relevancy Score: 0.75
Justification: The score is 0.75 because while the output provides some useful information about MSE SHAP, it includes irrelevant statements that do not directly contribute to answering the question, such as the mention of the context lacking information about MSE SHAP.

Faithulness Score: 0.75
Justification: The score is 0.75 because the actual output references Mean Squared Error (MSE) which is not explicitly mentioned in the context, creating ambiguity regarding its relevance.

Contextual Precision Score: 0.41
Justification: The score is 0.41 because while there are relevant nodes present, they are ranked lower than multiple irrelevant nodes. Specifically, the first node discusses SHAP values without any connection to MSE, and the second node is focused on unrelated topics like R-Tree index, which lowers the overall effectiveness of the retrieval context. The third node provides a definition for MSE, which is essential, and the fifth and sixth nodes offer valuable insights into SHAP and its significance, but their relevance is overshadowed by the earlier irrelevant nodes.

Contextual Recall Score: 0.67
Justification: The score is 0.67 because while the terms MSE and SHAP are supported by multiple documents in the retrieval context (nodes 1, 3, 5, and 6), the specific term 'MSE SHAP' is not explicitly mentioned, indicating a partial alignment with the expected output.

Contextual Relevancy Score: 0.38
Justification: The score is 0.38 because, although there are statements about SHAP values like 'SHAP values attribute to each feature the change in the expected model prediction,' they fail to specifically define or explain what MSE SHAP is, leading to a lack of direct relevance.

Figure 6: Automatic evaluation scores and justification for the third example question

- **Task 3:** Implement pdf loader and splitter of documents into chunks. **Timeline**: Nov 24 - Dec 1, 2024 **Assignees:** Kacper, Michał

- **Task 4:** Create embeddings from document chunks using a pre-trained model **Timeline:** Dec 1, 2024 - Dec 5, 2024 **Assignees:** Michał, Kacper

- **Task 5:** Integrate the retrieval system with a generative model to develop a functional chatbot prototype. **Timeline:** Dec 6, 2024 - Dec, 2024 **Assignees:** Michał, Jakub

- **Task 6:** Prepare mid-term progress report, document our findings, and prepare a presentation. **Timeline:** Dec 6, 2024 - Dec 11, 2024 **Assignees:** Jakub, Grzegorz

- **Task 7:** Extend the articles database and provide new example questions **Timeline:** Dec 12, 2024 - Dec 15, 2024 **Assignees:** Kacper, Jakub

- **Task 8:** Evaluate embeddings quality using provided metrics **Timeline:** Dec 16, 2024 - Dec 22, 2025 **Assignees:** Kacper, Jakub

- **Task 9:** Evaluate the chatbot using LLM-based metrics to measure response accuracy and reliability **Timeline:** Dec 16, 2024 - Dec 22, 2025 **Assignees:** Michał, Grzegorz

- **Task 10:** Optimize and fine-tune the chatbot to improve its performance in challenging scenarios. **Timeline:** Dec 22, 2025 - Dec 31, 2025 **Assignees:** Michał, Kacper

- **Task 11:** Prepare chatbot application **Timeline:** Jan 1, 2025 - Jan 5, 2025 **Assignees:** Michał

- **Task 12:** Document project findings, describe experiments, results, and conclusions, and prepare the final report and presentation **Timeline:** Jan 13, 2025 - Jan 20, 2025 **Assignees:** Jakub, Kacper, Grzegorz, Michał

The tasks above show that the team has successfully followed the planned timeline. The collaboration among team members has ensured that each task was completed effectively and on time.

8 Potential future directions:

In this section, we discuss potential avenues for the continued development of this project. Although the pipeline has been fully implemented, with an end-to-end system in place, several aspects could be expanded or improved upon in the future:

- **Improving Embedding Quality** - The quality of embeddings plays a critical role in the system's performance. In the future, experimenting with newer or more advanced embedding models could enhance the retrieval accuracy. Models that better capture the semantic meaning of the data, especially as they continue to evolve, could lead to more precise and contextually relevant results.

- **Exploring Alternative Vectorstores** - In our work, we used FAISS for managing the vector store. However, other solutions, such as ChromaDB or Pinecone, might offer performance improvements for the Retrieval-Augmented Generation (RAG) process. While this was outside the scope of our current project, experimenting with these alternatives could provide valuable insights into optimizing RAG system performance.

- **Ehancing SciBot** - A natural next step would be to refine the SciBot itself, potentially by integrating more advanced models. While we currently use open models, there may be opportunities in the future to incorporate models like ChatGPT (or its future iterations, such as `ChatGPT-4o mini`), which have shown strong performance in providing high-quality responses during the ground truth evaluation phase. Such advancements could lead to even better question-answering capabilities.

9 Conclusions

The implementation of the Retrieval-Augmented Generation (RAG) system has demonstrated promising results in the context of our scientific document analysis application. The system proved well-suited for the intended use case, as evidenced by its ability to integrate additional information from research articles in response to user queries.

We began by indexing documents and creating a vector store to represent the content of scientific papers. We used embedding models to generate

vector representations of document chunks, which were then utilized as context for a large language model (LLM) during question answering. A chat system was developed that leverages this vector store, allowing users to interact with the system via a simple graphical interface.

Our experiments with different embedding models on modified document chunks revealed that the `snowflake-arctic-embed` model outperformed other candidates across all evaluated metrics. The performance gap between the other models was insignificant, indicating that the available embedding models provide a solid foundation for building RAG systems.

Regarding SciBot's performance, we also explored various LLM models as responders to user questions. To assess the quality of these responses, we employed a novel evaluation system using the DeepEval package, which allowed for automated evaluation by different LLMs. The evaluation revealed that responses were generally accurate, with some exceptions where the model struggled to properly utilize context or where the RAG process itself failed to retrieve the most relevant context.

Overall, SciBot demonstrated good performance. The project successfully illustrated that current tools make it feasible to create domain-specific applications using customizable paper databases. Such a system can significantly benefit students and researchers who need to process and analyze large volumes of scientific literature, providing helpful explanations of complex concepts found in academic papers.

However, while the system shows promise, complete accuracy is not guaranteed. As with any AI-driven solution, users should remain cautious and verify uncertain responses, as demonstrated in our analysis of case studies. Further work is required to address edge cases and improve the model's ability to utilize context accurately.

References

[Abu-Rasheed et al.2024] Hasan Abu-Rasheed, Christian Weber, and Madjid Fathi. 2024. Knowledge graphs as context sources for llm-based explanations of learning recommendations. *arXiv preprint arXiv:2403.03008*.

[Almeida and Xexéo2019] Felipe Almeida and Geraldo Xexéo. 2019. Word embeddings: A survey. *arXiv preprint arXiv:1901.09069*.

[Antu et al.2023] Shouvik Ahmed Antu, Haiyan Chen, and Cindy K Richards. 2023. Using llm (large language model) to improve efficiency in literature review for undergraduate research. *LLM@ AIED*, pages 8–16.

[Arslan and Cruz2024] Muhammad Arslan and Christophe Cruz. 2024. Business-rag: Information extraction for business insights. *ICSBT 2024*, page 88.

[Caspari et al.2024] Laura Caspari, Kanishka Ghosh Dastidar, Saber Zerhoudi, Jelena Mitrovic, and Michael Granitzer. 2024. Beyond benchmarks: Evaluating embedding model similarity for retrieval augmented generation systems. *arXiv preprint arXiv:2407.08275*.

[Cho et al.2024] Sukmin Cho, Soyeong Jeong, Jeongyeon Seo, Taeho Hwang, and Jong C Park. 2024. Typos that broke the rag's back: Genetic attack on rag pipeline by simulating documents in the wild via low-level perturbations. *arXiv preprint arXiv:2404.13948*.

[Friha et al.2024] Othmane Friha, Mohamed Amine Ferrag, Burak Kantarci, Burak Cakmak, Arda Ozgun, and Nassira Ghoualmi-Zine. 2024. Llm-based edge intelligence: A comprehensive survey on architectures, applications, security and trustworthiness. *IEEE Open Journal of the Communications Society*, 5:5799–5856.

[Gao et al.2023] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

[Hadi et al.2023] Muhammad Usman Hadi, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, Seyedali Mirjalili, et al. 2023. A survey on large language models: Applications, challenges, limitations, and practical usage. *Authorea Preprints*.

[Hadi et al.2024] Muhammad Usman Hadi, Qasem Al Tashi, Abbas Shah, Rizwan Qureshi, Amgad Muneer, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, et al. 2024. Large language models: a comprehensive survey of its applications, challenges, limitations, and future prospects. *Authorea Preprints*.

[Huang et al.2023] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

[Kulkarni et al.2024] Mandar Kulkarni, Praveen Tangarajan, Kyung Kim, and Anusua Trivedi. 2024. Reinforcement learning for optimizing rag for domain chatbots. *arXiv preprint arXiv:2401.06800*.

[Liu et al.2020] Qi Liu, Matt J Kusner, and Phil Blunsom. 2020. A survey on contextual embeddings. *arXiv preprint arXiv:2003.07278*.

[Minaee et al.2024] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey. *arXiv preprint arXiv:2402.06196*.

[Pal et al.2024] Soumen Pal, Manojit Bhattacharya, Sang-Soo Lee, and Chiranjib Chakraborty. 2024. A domain-specific next-generation large language model (llm) or chatgpt is required for biomedical engineering and research. *Annals of biomedical engineering*, 52(3):451–454.

[Rawte et al.2023] Vipula Rawte, Amit Sheth, and Amitava Das. 2023. A survey of hallucination in large foundation models. *arXiv preprint arXiv:2309.05922*.

[Salemi and Zamani2024] Alireza Salemi and Hamed Zamani. 2024. Evaluating retrieval quality in retrieval-augmented generation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2395–2400.

[Sallam2023] Malik Sallam. 2023. The utility of chatgpt as an example of large language models in healthcare education, research and practice: Systematic review on the future perspectives and potential limitations. *MedRxiv*, pages 2023–02.

[Shankar et al.2024] Shreya Shankar, JD Zamfirescu-Pereira, Björn Hartmann, Aditya Parameswaran, and Ian Arawjo. 2024. Who validates the validators? aligning llm-assisted evaluation of llm outputs with human preferences. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, pages 1–14.

[Singh2023] Jaswinder Singh. 2023. Combining machine learning and rag models for enhanced data retrieval: Applications in search engines, enterprise data systems, and recommendations. *Journal of Computational Intelligence and Robotics*, 3(1):163–204.

[Vakayil et al.2024] Sonia Vakayil, D Sujitha Juliet, Sunil Vakayil, et al. 2024. Rag-based llm chatbot using llama-2. In *2024 7th International Conference on Devices, Circuits and Systems (ICDCS)*, pages 1–5. IEEE.

[Zheng et al.2023] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

[Zhou et al.2024] Hao Zhou, Chengming Hu, Ye Yuan, Yufei Cui, Yili Jin, Can Chen, Haolun Wu, Dun Yuan, Li Jiang, Di Wu, et al. 2024. Large language model (llm) for telecommunications: A comprehensive survey on principles, key techniques, and opportunities. *arXiv preprint arXiv:2405.10825*.

[Zhou2024] Ren Zhou. 2024. Advanced embedding techniques in multimodal retrieval augmented generation a comprehensive study on cross modal ai applications. *Journal of Computing and Electronic Information Management*, 13(3):16–22.

Appendix

Images

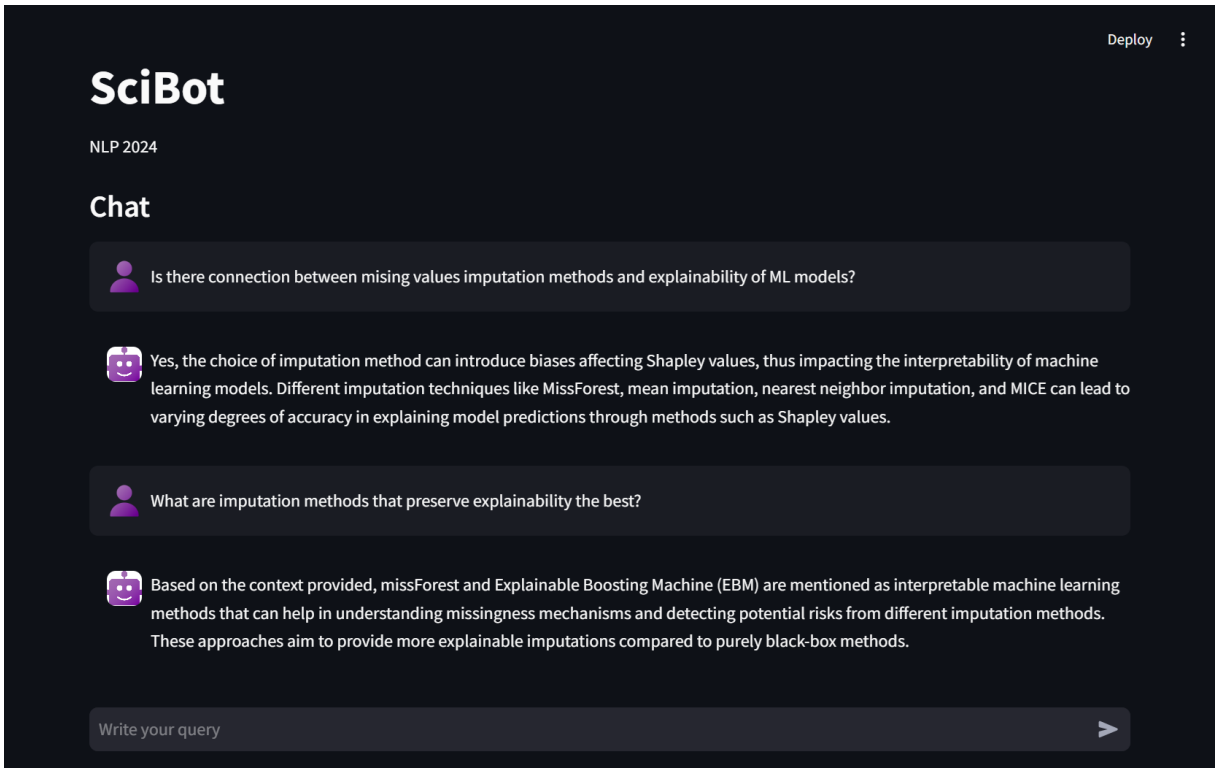

Figure 7: SciBot GUI and example conversation

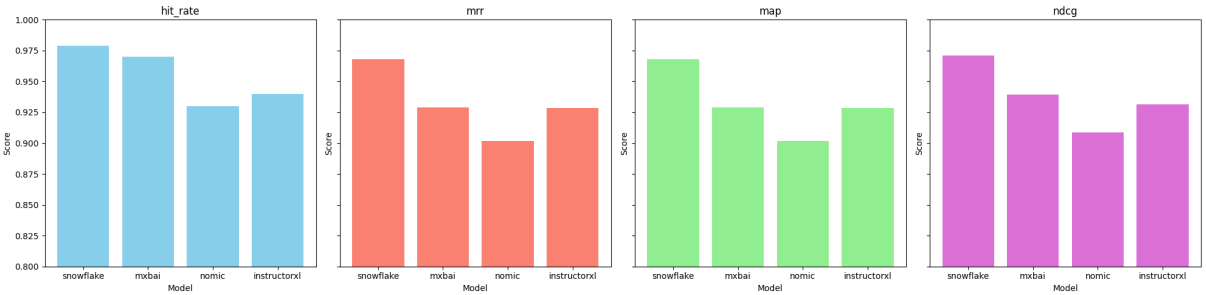

Figure 8: Evaluation results of different embedding models

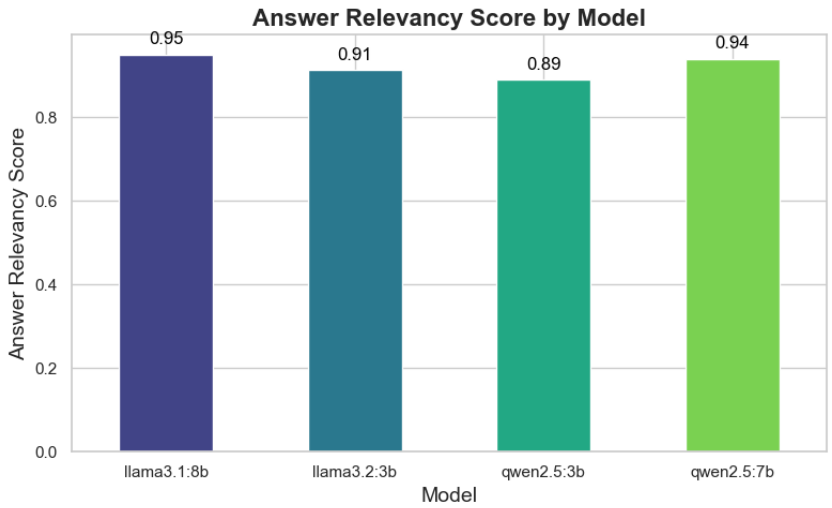

Figure 9: Answer Relevancy scores for different LLMs

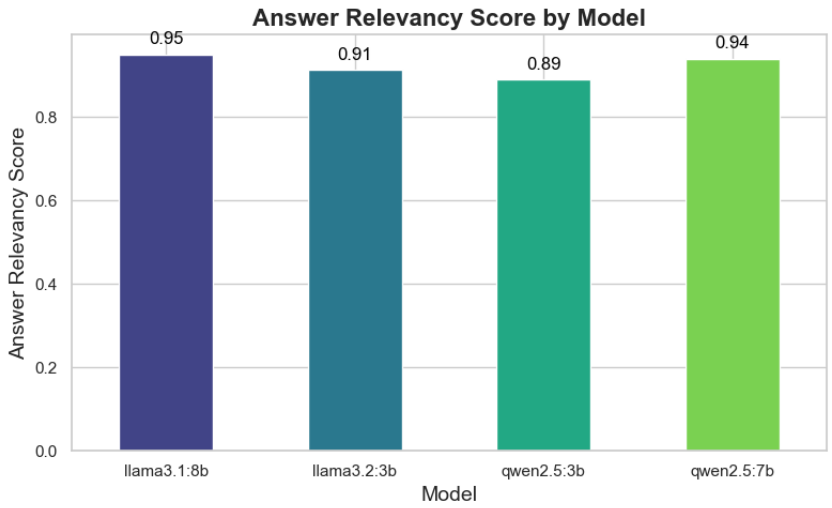

Figure 10: Answer Relevancy scores for different LLMs

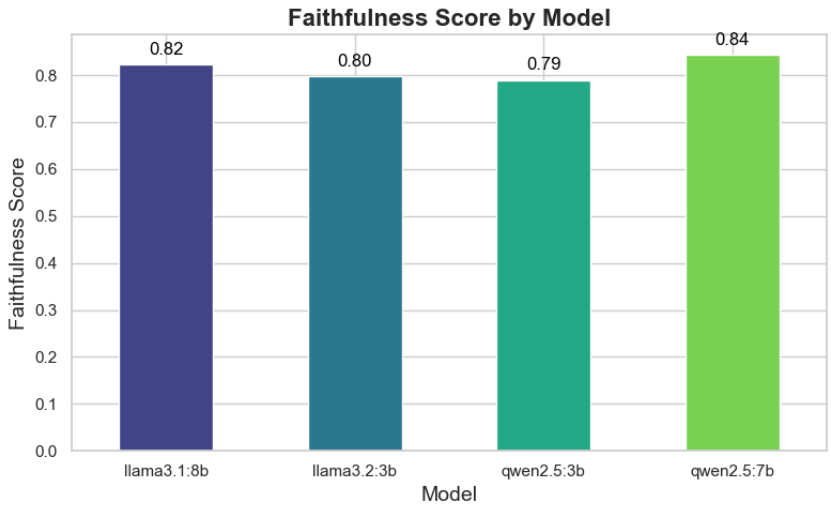

Figure 11: faithulness scores for different LLMs

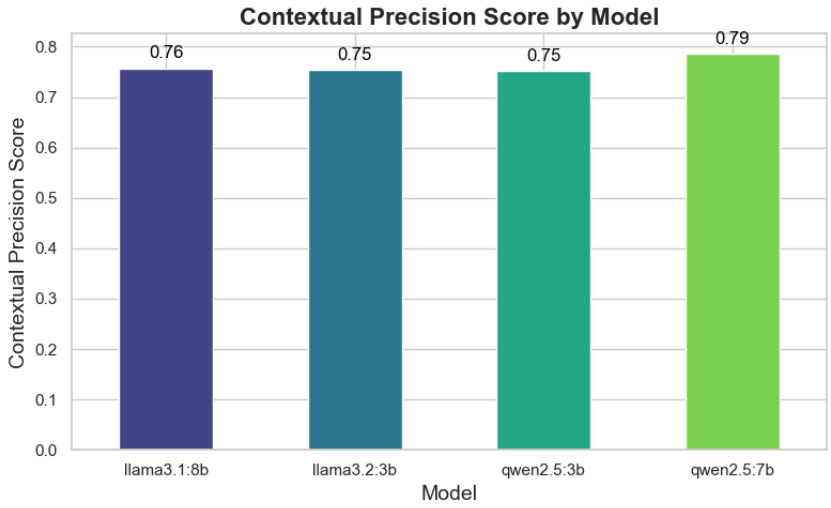

Figure 12: Context Precision scores for different LLMs

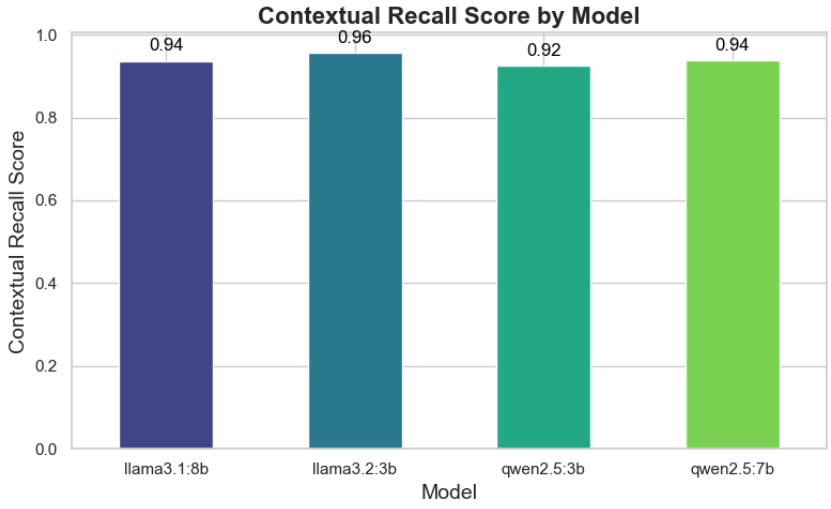

Figure 13: Context Recall scores for different LLMs

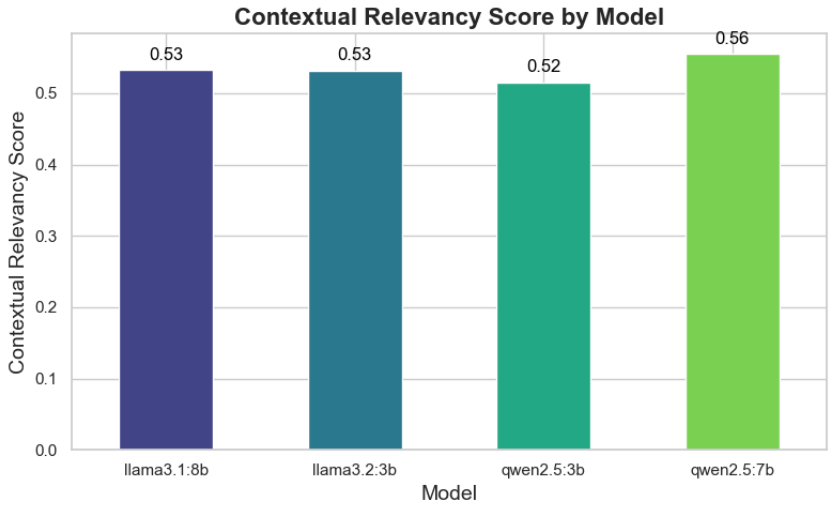

Figure 14: Context Relevancy scores for different LLMs

Reproducibility checklist

Description	**Our Results**
Model description	Your results here Embedding models: • Ollama - `snowflake-arctic-embed` • Ollama - `mxbai-embed-large` • Ollama - `nomic-embed-text` • HuggingFace - `instructor-xl` LLMs used in SciBot testing (All from Ollama): • `qwen2.5:3b` • `qwen2.5:7b-instruct-q4_0` • `llama3.1:3b` • `llama3.2:8b` LLMs for generating ground truth and evaluating: • OpenAI - `GPT-4o mini`

Description	**Our Results**
Infrastructure	• CPU: 13th Gen Intel(R) Core(TM) i5-13500H (16 CPUs), 2.6GHz • RAM: 16GB • NIVIDIA GeForce RTX 4060 • Windows 11 • Python 3.12
Runtime parameters	• Splitting pdfs into chunks. ca. 3 min • Creating vector stores. (per model) 5-8 min • Response to 1 question. 5-25 seconds • Generating ground truth responses. ca. 20 min • Generating automatic evaluation. ca. 4 hours

Description	Our Results
Metrics	The embeddings were be evaluated using the following metrics: • **Hit Rate:** The percentage of queries where the retrieved documents contain the correct answer. • **Maximal Marginal Relevance (MMR):** Measures the diversity and relevance of retrieved documents. • **Normalized Discounted Cumulative Gain (nDCG):** Evaluates the ranking quality of retrieved documents. • **Mean Average Precision (MAP):** Summarizes retrieval precision across queries. LLMs were evaluated using the following metrics: • **Answer Relevancy Metric:** Evaluates the relevance of responses to queries. • **Faithfulness Metric:** Measures factual accuracy against source materials. • **Contextual Precision and Recall:** Assesses precision and recall of responses in the context of retrieved documents. • **Contextual Relevancy:** Detects unsupported or fabricated content in responses.
Data stats	Database of scientific articles collected by authors. 75 different articles with number of pages varying from 8 to 50.