RAG-based conversational tool for scientific materials
Proof of Concept 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 usage of Large Language Models (LLMs) enhanced by Retrieval-Augmented Generation (RAG) to answer specific questions from user-provided scientific papers. The planned system will generate 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 This approach addresses the limitations of LLMs, such as outdated knowledge and hallucinations, ensuring reliable outputs.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.

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 then 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 large 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 pretrained 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 in a qualitative and quantitative manner.

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; Lewis et al., 2020). 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 that are 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; Wilcock, 2024). 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, Discounted Cumulative Gain (DCG), 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 also 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 not only provides a challenging benchmark for our chatbot but also ensures that we have direct familiarity with the content, enabling more effective quality assessment and evaluation of the chatbot's responses.

Additionally, this collection will expand over time as we discover more relevant articles during the semester. This dynamic growth aligns perfectly with the capabilities of Retrieval-Augmented Generation (RAG), which is designed for knowledge bases that evolve continuously. This setup will serve as an excellent test case, simulating real-world scenarios where the knowledge base must adapt to changing information needs.

We are providing a link to a zip file containing almost 40 selected papers for now: **Articles database**.

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.
- **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, DCG, 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.
- **Discounted Cumulative Gain (DCG):** 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.
- **Hallucination Metric:** Detects unsupported or fabricated content in responses.

5 Experiments

This section outlines the progress made on the project so far. We provide a description of the technical setup used in the development of the solution, detailing the techniques employed. Following that, we will present the initial tests and results of our solution, offering an analysis and conclusions based on these outcomes. It is important to note that this section includes only the content that has been implemented and tested up to this point. Future updates will expand on these findings as the project progresses.

5.1 Work Progress

So far, we have successfully implemented several components of our solution. The primary framework used for developing the solution is Langchain, which offers an intuitive interface for building scalable applications. Below, we describe the key components that have been developed:

Document Loading The `PyPDFLoader` class is used to load content from scientific articles in PDF format. This component parses the PDFs to extract both the textual content and relevant metadata. Metadata, such as the source of the file, is retained, which enables context-aware retrieval. The loader processes all PDF files within a specified directory and combines the text from individual pages into complete documents. Each document is represented as a `Document` object, encapsulating the text and associated metadata.

Document Splitting To handle large documents and accommodate the constraints of transformer-based models, we use the `RecursiveCharacterTextSplitter` to divide the text into manageable chunks. Currently, we focus on using the Hugging Face `AutoTokenizer` with the `hkunlp/instructor-xl` model to align with tokenization constraints. Each chunk inherits metadata from the parent document and is enriched with additional metadata, such as a unique `chunk_idx`, which ensures precise traceability during retrieval.

Embedding Generation and Indexing Embeddings are generated for each text chunk using the `HuggingFaceInstructEmbeddings` module with the `hkunlp/instructor-xl` model. This model transforms textual data into highdimensional vector representations, enabling efficient semantic similarity calculations. The generated embeddings are stored in a FAISS (Facebook AI Similarity Search) index, an efficient data structure for approximate nearest neighbor search. The FAISS index serves as the backbone of the retrieval system, enabling fast and contextually accurate semantic searches across a large collection of scientific articles. This setup ensures scalability and supports the chatbot's ability to provide accurate, context-aware responses to complex scientific queries.

Chatbot Prototype The chatbot, named **SciBot**, is initially built using the `llama3.1:latest` model, which offers strong instruction-following capabilities. It maintains conversational history to preserve context across multiple turns of interaction. To ensure precise and contextually relevant responses, SciBot reformulates user queries into standalone questions when needed, leveraging prior interactions to enhance understanding.

RAG Approach The chatbot employs a Retrieval-Augmented Generation (RAG) approach, utilizing a FAISS database pre-generated from scientific documents. The FAISS index enables efficient retrieval of relevant text chunks based on Maximal Marginal Relevance (MMR), a technique that ensures the retrieved context is both diverse and highly relevant to the user's query. This approach combines the strengths of information retrieval and generative models to enhance the chatbot's accuracy and informativeness.

Response Generation Responses are generated by combining the retrieved context with the user's query. The chatbot is designed to provide concise and accurate answers, explicitly acknowledging when sufficient context is unavailable to generate a reliable response. This design minimizes the risk of hallucinations, ensuring that the chatbot prioritizes accuracy and clarity, offering users well-informed assistance.

Thanks to these components, we are now able to provide a functional database of articles and interact with the system effectively. Currently, we are using the `llama3.1:latest` LLM and the `hkunlp/instructor-xl` embedding model. However, the code is designed to be modular, allowing for easy replacement or updating of these components in the future.

5.2 Testing

The created chatbot was tested using 25 questions formulated by us based on the content of scientific articles. Our knowledge of the content helped us create questions that users could realistically pose, as we can also be users of this solution in the future. The main goal in creating the questions was to design those that could showcase the chatbot's ability to retrieve information from articles, such as methods, metrics, and examples. We recorded all questions along with the response time, the exact model response, the six best-retrieved documents, and the content used by the model to generate its response. For now, we evaluated the responses using our own knowledge and visual inspection and compared them to the content of the retrieved articles. Unfortunately, this process requires a lot of manual assessment.

Additionally, to evaluate the chatbot's performance, we sought a comparison response. To achieve this, we fed the context from the retrieved documents provided by the RAG system to another language model — OpenAI's `GPT-4o-mini` — along with the same question. This means that the chatbot's response will be compared to the response from the OpenAI model, which had the context from the articles, and we will treat this response as the ground truth. This comparison allows us to assess how well the chatbot handles the entire pipeline: retrieving the correct context from the article database and using it to generate an informative and concise response.

The testing output generated during the process is too large to be presented fully in this report, so we will only provide a few interesting examples that demonstrate the model's capabilities. We will also analyze the results, pointing out the most important observations and common issues found in the responses. It is important to note that these examples represent only a subset of the larger picture of responses.

We will focus on showing the answers and the chatbot responses alongside the ground truth, as the context retrieved from the articles is too large to display here. The full table with all the questions will be available in the course repository.

In Figure 1, we can see an example of a question, the response of the chatbot, and the ground truth, which is the response from `GPT-4o-mini` model provided with the context of articles retrieved by the RAG system. This example shows

Question 5.1: Please give some examples of metaheuristics.

Model Response:
Some examples of metaheuristics include Simulated Annealing (SA), Tabu Search (TS), Evolutionary Algorithms (EAs), Ant Colony Optimization algorithm (ACO), and Particle Swarm Optimization (PSO).

Ground Truth:
Some examples of metaheuristics include Simulated Annealing (SA), Tabu Search (TS), Evolutionary Algorithms (EAs), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO).

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

very similar responses with almost no difference. The chatbot's response is accurate and truthful, based on the content provided in the articles, but also supported by general knowledge of metaheuristics. This is an example of a good response, which is common in our chatbot, where the answers are sufficient and informative. However, there can still be occasional issues, as seen in some of the other examples.

The question presented in Figure 2 concerns the GGA (Gender-Based Genetic Algorithm) used in the context of automatically configuring solvers. As demonstrated, our chatbot failed to correctly identify the algorithm. Furthermore, the response begins with an ironic statement, "A very specific question," which does not contribute to answering the query and highlights insufficient understanding. Instead of addressing the correct algorithm, the chatbot provided an explanation for an algorithm from a completely unrelated domain—density functional theory. This demonstrates a significant instance of hallucination, where the chatbot generated irrelevant content by incorrectly associating the abbreviation "GGA" with a concept outside the provided context. On the other hand, the ground truth response successfully captured the core concept of the GGA algorithm described in the retrieved context. However, it incorrectly expanded the GGA abbreviation as "Genetic Algorithm with Genetic Engineering," which is inaccurate. This error is likely a

Question 5.2: "What is the idea behind the AAC approach called "GGA"?

Model Response:
A very specific question!
After some research, I found that GGA stands for Generalized Gradient Approximation. It's an approximation method used in density functional theory (DFT) to calculate the exchange-correlation energy of a system. In DFT, the total energy of a system is divided into three parts: the kinetic energy, the electron-electron interaction energy, and the exchange-correlation energy. The GGA approach is a refinement of earlier local density approximation (LDA) methods that try to capture the effects of gradient terms in the exchange-correlation functional. GGA approximations are used in various fields, including materials science, chemistry, and physics, to study the electronic structure and properties of systems such as solids, liquids, and molecules. They provide a more accurate description of the system compared to LDA methods, especially for systems with strong gradients or anisotropies. Is this the GGA you were thinking of?

Ground Truth:
The GGA (Genetic Algorithm with Genetic Engineering) approach, as described in the context, integrates a surrogate model to evaluate the quality of new configurations within its operation. It combines a competitive population, which competes in races, with a non-competitive population, allowing the winning configurations from the competitive races to be recombined with individuals from the non-competitive group. The crossover operator generates new individuals based on parameter tree crossover and uses the surrogate model to assess their ranks in a tournament-like setting. The individuals with the best ranks are then selected to form the next generation. Unlike traditional random forest methods, GGA focuses specifically on high-quality configurations, utilizing specialized splitting criteria that emphasize these areas of the input space.

Figure 2: Second example of question and comparison of chatbot response and ground truth

result of insufficient detail in the retrieved context, where the exact name of the algorithm was not explicitly stated. This behavior underlines an important limitation of the retrieval system: when critical information is missing, the system struggles to generate a fully accurate response. When context is inadequate or ambiguous, the model generates content unrelated to the query, which reduces the credibility of the system. Even when the retrieval system surfaces relevant chunks, it might fail to retrieve or highlight explicit and essential details (e.g., the correct full name of the algorithm), which can impact the model's response accuracy.

The response shown in Figure 3 contains an irrelevant statement regarding the lack of information on 'PAP' and 'GLOBAL', which detracts from directly addressing the user's query. While the relevant nodes in the document provide valuable insights into the PAP approach called 'GLOBAL', the chatbot fails to extract and use this relevant information correctly. In this case, the chatbot's response does not match the expected information from the retrieved context, indicating an issue with the retrieval or the model's understanding of the context. This points to the necessity for further fine-tuning and improvements in both retrieval and response generation.

It is important to note that most responses were of high quality, with the chatbot providing information that closely matched the content of the articles and aligned well with the ground truth responses. The examples presented here aim to highlight common issues that we need to address. While many responses were accurate and informative, some failed to correctly identify relevant content in the retrieved context. Some responses indicated that not all relevant information was retrieved, and a few seemed to hallucinate details. This evaluation process also revealed the inefficiency of manual testing. As a result, we strongly recognize the need for automated evaluation methods, which would allow for faster, more scalable, and quantifiable assessments rather than relying solely on human judgment.

6 Work Division

In this section, we provide an overview of the tasks completed so far, along with a breakdown of work distribution among team members. Our team has made significant progress, and we are pleased to

Question 5.3: What is the idea behind the PAP approach called "GLOBAL"?

Model Response:
I don't have information on a concept or methodology called 'PAP' that includes an approach called 'GLOBAL'. It's possible that it might be specific to particular fields, industries, or research. If you could provide more context or details about what PAP and GLOBAL refer to in your query, I'll do my best to offer a helpful response.

Ground Truth:
The PAP approach called "GLOBAL" focuses on optimizing performance by exploiting the advantages of different population-based algorithms through a parallel framework. Its main goal is to improve the overall performance across a diverse range of multi-objective problems by employing multiple algorithms that operate independently and in parallel. This strategy allows the algorithm portfolio to explore the search space globally, providing a higher chance of finding better solutions than any single algorithm could achieve on its own. Through the communication and migration of solutions among subpopulations, the PAP framework facilitates the sharing of beneficial information and enhances diversity in the search process. This method aims to effectively reduce the risk associated with solving various problems by improving cooperation between different member algorithms while maintaining their individual search biases.

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

report that the project is ahead of schedule. The timeline outlined in the project proposal has been successfully followed, with some additional steps incorporated, allowing us to deliver more than initially expected. Below, we detail the tasks completed up 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

- **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 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

As seen in the tasks above, the team has successfully followed the planned timeline, and work has been progressing efficiently. The collaboration among team members has ensured that each task was completed effectively and on time. With all milestones met so far, we are confident that we will continue to make excellent progress toward the final stages of the project.

7 Future Works:

In this section, we outline the next steps and plans for the continued development of the project. While significant progress has been made, several critical tasks remain to achieve the final solution that was initially envisioned. The following are key areas that will require attention in the near future:

- **Expansion of article database and questions:** For the final version of our solution, it is essential to expand the database of articles. While the current results show promise, increasing the size and diversity of the article set will better simulate a dynamically changing knowledge database. This will allow us to test the retrieval capabilities of the system in a broader context, as the model will need to search through a larger variety of topics and identify the most relevant information from more complex and varied content. Additionally, we will prepare more diverse and challenging questions to evaluate how well the system handles information retrieval across the expanded dataset.

- **Evaluation of different embedding models and LLMs:** A crucial part of future development will be evaluating the performance of different embedding models and language models (LLMs). As discussed in Section 4, we will assess the effectiveness of these models using various evaluation methods. We will use metrics such as Hit Rate, Maximal Marginal Relevance (MMR), Discounted Cumulative Gain (DCG), and Mean Average Precision (MAP) to measure the quality of retrieved information. Furthermore, we will apply LLM-based evaluation methods, using tools like the DeepEval package, to assess response quality. These evaluations will include metrics such as Answer Relevancy, Faithfulness, Contextual Precision and Recall, and the Hallucination Metric. This comprehensive evaluation process will help us identify the best-performing models, enabling us to select the optimal embedding model and LLM for the chatbot, ensuring the highest quality of responses.

- **Front-end of application:** Another key development step will involve the creation of a user-friendly front-end application using Streamlit. This application will allow users to interact with the chatbot through a visually appealing interface. By separating the backend logic from the front-end interface, we will ensure that the system is modular and scalable. The front end will enhance the accessibility and usability of the solution, providing a seamless experience for users to interact with the chatbot in a web application format.

Below, we provide a potential division of tasks for the remaining development stages, along with their respective timelines and team assignments:

- **Task 1:** Extend the articles database and provide new example questions **Timeline:** Dec 12, 2024 - Dec 15, 2024 **Assignees:** Kacper, Jakub

- **Task 2:** Evaluate embeddings quality using provided metrics **Timeline:** Dec 16, 2024 - Dec 22, 2025 **Assignees:** Kacper, Jakub

- **Task 3:** Evaluate the chatbot using LLM-based metrics to measure response accuracy and reliability **Timeline:** Dec 16, 2024 - Dec 22, 2025 **Assignees:** Michał, Grzegorz

- **Task 4:** Optimize and fine-tune the chatbot to improve its performance in challenging scenarios. **Timeline:** Dec 22, 2025 - Dec 31, 2025 **Assignees:** Michał, Kacper

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

- **Task 6:** 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

These tasks will guide the next phase of the project, ensuring that all necessary steps are taken to finalize the chatbot's development. By following this structured approach, we are confident that the final solution will meet the project's objectives.

8 Conclusions

The implementation of the Retrieval-Augmented Generation (RAG) system has shown promising results thus far. The RAG system appears well-suited for our intended applications based on the articles we have collected and the example questions we developed.

We began by indexing documents and creating a vector store to represent the content of the scientific papers in our dataset. We generated embeddings for the document chunks using embedding models, which could then be provided to a large language model (LLM) as additional context. This process enabled us to build a functional system capable of utilizing the context from articles to respond to user queries.

Our initial testing involved predefined questions, which allowed us to assess the system's performance. While the chatbot performed well in most cases, there were some instances where issues were identified, such as the inaccurate use of context or the generation of untruthful content. Unfortunately, the evaluation process was manual, relying on our subjective assessment of the responses. Although this approach was sufficient for identifying general performance trends, it proved to be inefficient and inadequate for providing detailed, objective feedback.

In light of these challenges, it is clear that automating the evaluation process is crucial for improving efficiency and providing more measurable insights. By implementing automated evaluation methods, we will be able to more effectively assess the chatbot's performance and identify areas for further refinement.

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*.

[Lewis et al.2020] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

[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.

[Wilcock2024] Graham Wilcock. 2024. New technologies for spoken dialogue systems: Llms, rag and the genai stack. In *14th International Workshop on Spoken Dialogue Systems*.

[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.