Natural Language Processing & Information Extraction

Retrieval Augmented Generation for Academic Question Answering

Mohammadali Dehghani - 12432957
Amir Saadati - 12434679
Amina Kadic - 12439016
Meliha Kasapovic - 12439367

Course: NLP & IE
Academic Year: 2025-2026
January 23, 2026

Management Summary

Objective and Task Overview

The objective of this project was to design and evaluate a **retrieval-augmented generation (RAG)** system for answering predefined research questions related to machine learning. The focus of the work was not only on answer correctness, but also on assessing the reliability of generated answers, particularly with respect to **hallucinations**, answer grounding in source documents, and alignment with the original question intent. This work implements two RAG systems and three versions of their pipelines in order to assess how different setups change the outcome of the experiment.

Solution Overview

The implemented solution is based on a retrieval-augmented generation (RAG) pipeline that combines document retrieval with large language model-based answer generation. The goal of this approach is to ensure that generated answers are grounded in relevant source documents rather than relying solely on the model's internal knowledge.

A curated collection of scientific publications was used as the underlying knowledge source. Several document retrieval strategies were explored and compared in order to identify an approach that consistently surfaces relevant and informative content for the given questions. Based on both quantitative indicators and qualitative inspection, the most reliable retrieval strategy was selected for further use in the system.

Multiple versions were implemented in order to analyze differences in obtained results. The first approach was implementing two RAG systems on one set of questions with labeling of chunks done by a single person. These two systems of RAG used off-the-shelf retrieval model from Huggingface. To enhance the retrieval, existing questions were reformulated, the labeling was done by three different large language models and the retreival model was trained on those labelings.The two RAG systems were also run on this new set of questions and labels. The last version of two RAG systems were implemented using embedding retrieval that was trained on relabeled chunks, which was done manually by three people.

Two complementary approaches to answer generation were evaluated to better understand trade-offs between answer fluency and reliability:

- **Vanilla RAG**, which allows free-form generation and prioritizes fluent and natural responses, and
- **Strict Constrained RAG**, which restricts generation to retrieved evidence and discourages unsupported or speculative claims when sufficient information is not available.

External Resources

The project relied on several concrete external tools and resources. Scientific publications were collected using the **arXiv API** and served as the primary knowledge source for the system.

For document retrieval, dense text embeddings were generated using the **sentence-transformers/all-MiniLM-L6-v2** model, and similarity search was performed with the **FAISS** library to efficiently retrieve relevant document chunks. Traditional retrieval baselines, including keyword-based retrieval and TF–IDF, were also implemented for comparison.

Answer generation was performed using the **TinyLlama/TinyLlama-1.1B-Chat-v1.0** language model. Large language models were additionally used to support automatic relevance annotation and qualitative evaluation of system outputs.

Key Challenges

Several challenges were encountered during the project. The most significant issue was the tendency of language models to generate fluent but factually unsupported answers, even when relevant documents were retrieved. Ensuring alignment between the question, retrieved evidence, and generated answer proved difficult, particularly when source documents did not explicitly state the requested information.

Another challenge involved the creation of reliable relevance annotations. Automatically generated labels enabled scalability but introduced noise, while manual annotation required substantial effort and time.

One of the ideas was to implement the **RAGAS** framework for hallucination detection and RAG systems evaluation. However, due to hardware limitations, multiple timeout errors were encountered. Therefore, hallucinations and systems were qualitatively evaluated.

Results and Limitations

The evaluation reveals systematic differences between unconstrained and constrained generation strategies. Vanilla RAG consistently produces fluent and well-structured answers, but exhibits a high rate of hallucinations, including unsupported claims and deviations from the original question intent. This behavior indicates that surface-level coherence does not reliably correlate with factual correctness.

Strict Constrained RAG substantially reduces explicit fabrication by limiting generation to retrieved evidence. However, this improvement comes at the cost of answer specificity and completeness, with responses frequently becoming generic, repetitive, or incomplete when the available context does not directly support the query.

Overall, the results demonstrate that retrieval quality and prompt-based constraints, while necessary, are not sufficient to ensure reliable question answering. Limitations related to answer alignment, faithful interpretation of retrieved content, and robustness remain unresolved, highlighting the need for additional validation and verification mechanisms.

Next Steps

Future work should focus on **improving question design**, selecting **higher-quality and more focused source documents**, and introducing systematic evaluation methods for hallucination detection and answer faithfulness, such as the **RAGAS framework** on different, more powerful hardware. From an application perspective, integrating verification mechanisms, clearer explanations of answer uncertainty, and structured outputs would significantly improve user trust and decision-making support.

Team Contributions

Amir Saadati worked on tokenization, parsing, writing questions and labeling the first version, as well as the followup version chunking and improvements. Mohammadali Dehghani worked on paper retreival, rule-based retreivers as well as improving consequent versions of retreival. Meliha Kasapovic worked on machine learning baselines, as well as the constrained RAG approach, and initial conll-u conversion. Amina Kadic worked on baseline comparisons, Vanilla RAG implementation, and improvements on both RAG systems. All team members worked on manual labeling interchangeably, and qualitative evaluations of different parts of the project pipeline, as well as improvements in chunking, question reformulation, and minor improvements.