A question-answering pipeline for congressional hearings using Retrieval-Augmented Generation

Alexandros Aristomenis Nikoloudis[1]

Supervisors: Stephanie Tan[1]**, Edgar Salas Gironés**[1]

[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

June 22, 2025

Name of the student: Alexandros Aristomenis Nikoloudis
Final project course: CSE3000 Research Project
Thesis committee: Stephanie Tan, Edgar Salas Gironés, Odette Scharenborg

***Abstract*—Understanding and reasoning over complex policy discussions, such as those found in congressional hearings, has been difficult for individuals due to lack of traditional punctuation and word order. Although recent advances in the area of Natural language processing (NLPs), particularly the development of Large Language Models (LLM), have significantly improved general text understanding, applying these models directly to hearing transcripts results in irrelevant or shallow responses. This work explores the use Retrieval-Augmented Generation (RAG), a method that improves response quality by combining document retrieval with LLM-based text generation. RAGs offer a promising paradigm for addressing this issue, as it provides a solution to domain-specific applications that contain specialized terminology and context, reducing hallucination. Through this research, we evaluate and implement different retrieval strategies for relevant information and chunking configurations. We explore how different retrieval strategies influence the completeness (coverage given contextual information), relevance (focus on original query), and faithfulness (accuracy of response given contextual information) of responses generated from the RAG. In addition to improving knowledge of RAG configurations in policy research, and providing a useful application to congressional hearings, the paper establishes the foundation for further applications in other fields where structured document interpretation is necessary.**

I. Introduction

Each year, the U.S. Congress addresses over 20,000 legislative items across approximately 1,500 hearings data [4], [5]. Congressional hearings contain complex policy discussions, which often lack consistent structure and syntactic regularity. This makes them challenging to analyze—not only for individuals, but also for existing NLP systems. As the government aims for greater transparency and accountability, there is a growing demand for tools that allow citizens, journalists, and researchers to easily access and interrogate legislative records.

To meet this demand, question answering pipelines have been developed in recent years by utilizing LLMs. While these models show promise, their effectiveness depends heavily on the relevance of the retrieved information and how well it is integrated into the generation process. Notably, a study Ling et al. [8] showed the limitations of LLMs for domain-specific applications, which present unique challenges due the heterogeneity of domain data, sophisticated domain knowledge, unique objectives, and contextual complexities, requiring specialized adaptation techniques to perform well in specialized fields.

Such cases have led to the increasing use of RAG, a framework that combines information retrieval with language generation to produce context-aware, grounded responses. Figure 1 illustrate the core architecture of a typical RAG pipeline. A user question and an associated article are first processed by a retriever, which searches a document collection to return the top-k relevant results. These results are then passed into a prompt construction stage that formulates the input to the LLM. The LLM uses only this prompt to generate a contextually grounded answer. RAG aim to enhance the accuracy of LLM outputs by incorporating data only from external sources. Research on how specific retrieval and prompting strategies of RAG frameworks affect the quality of generated responses has received attention, but remains limited in certain fields. In particular, the use of RAG systems to extract or summarize information from complex, political documents like U.S. Congressional hearing transcripts has not been thoroughly explored.

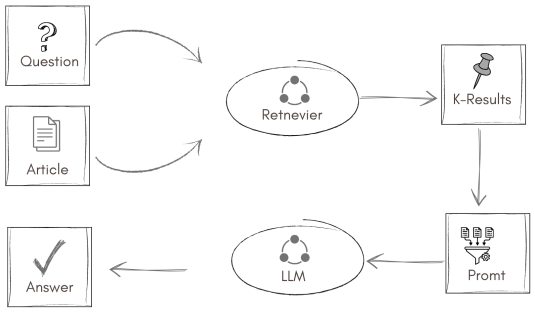

Fig. 1. Retrieval-Augmented Question Answering Workflow.

This study investigates the application of RAG frameworks using congressional hearings as a case study. Given the importance of accurate information retrieval in civic tech and political discourse analysis, understanding the impact of retrieval configurations on response quality is essential for building a trustworthy and effective system. As RAG approaches become more integrated into research and decision-making tools, this study provides a much-needed systematic evaluation of their strengths and limitations within this domain.

This study addresses this gap by investigating the following primary research question: "How do variations in retrieval strategies influence the accuracy and quality of responses generated by a RAG system when applied to U.S. Congressional hearing transcripts?".

This pipeline is tailored to questions based on U.S. Congressional hearing transcripts, generating responses strictly based on retrieved textual evidence, with no use of prior model knowledge. However, other programs can also use it as basis to develop a RAG for other fields. We aim to understand which combination of strategies is more efficient, allowing not only the validation of the retrieval setup but also supports better design and evaluation of similar QA pipelines in specialized domains.

To guide this investigation, we focus on the following subquestions:

- How do different retrieval methods (BM25, dense retrieval, reranking) compare in retrieving contextually relevant passages from congressional hearing transcripts?
- How does the size of textual chunks (e.g., 4, 5, 6, 10 sentences) affect the completeness and faithfulness of answers across retrieval strategies?
- How does the inclusion of open-ended questions impact model performance across the different retrieval strategies?
- How does the use of multiple-choice questions affect model performance across various the retrieval strategies?

To evaluate this, we develop a question answering pipeline tailored to congressional hearings, which supports experimentation with different RAG retrieval configurations. The pipeline is built using the LlamaIndex framework, with ChatGPT-4o-mini serving as the frozen language model responsible for response generation.

The system is evaluated under a range of retrieval strategies and document chunking settings to understand how each configuration influences answer quality. By comparing both open-ended and multiple-choice question types, the study aims to generate a comprehensive understanding of how different retrieval mechanisms perform in extracting information from complex political documents. This investigation not only evaluates performance across technical metrics but also offers insights that can inform the design of more effective domain-specific question-answering tools.

To summarize, this study presents a modular RAG pipeline tailored for U.S. congressional hearing transcripts, integrating sentence-window chunking, dense embeddings, and Chroma-based indexing. It includes a comparative evaluation of BM25, dense vector retrieval, and LLM-based reranking across multiple chunk sizes using a frozen GPT-4o-mini model. Additionally, it provides an empirical analysis of how retrieval strategies and prompt formats influence answer quality in domain-specific QA tasks.

II. Background and Related Works

The study employs a RAG approach, combining LLMs with external knowledge bases to enhance the accuracy and contextual relevance of generated responses, addressing common issues such as hallucinations and outdated information in traditional LLMs. This section presents preliminary information regarding the field and previous research conducted outlining their approaches and key findings.

A. Fundamentals of the RAG technique

RAGs were initially introduced in by Patrick Lewis et al. [2] during 2020 to improve the performance of Knowledge-Intensive NLP Tasks. RAG demonstrates significant improvements over existing methods by effectively leveraging both parametric and non-parametric memory components. This provided a solution to LLM issues such as hallucinations, outdated knowledge, and limited reasoning capabilities. Compared to standard LLMs, which rely solely on provided pre-trained knowledge to generate answers, RAG systems enhance LLMs by integrating relevant information, from external knowledge sources, , in our case the congressional hearings database.

RAG can be described as a model that combines information retrieval with text generation, aiming to enhance the factual accuracy and depth of generated responses by grounding them in external knowledge sources. The architecture integrates a retriever and a generator. In most RAG implementations, including this, the generator is a frozen LLM, meaning its parameters remain unchanged during downstream tasks. This design allows the model to maintain its general language capabilities while relying on the retriever and document index for task-specific execution, enhancing modularity, efficiency, and adaptability across domains. Regarding the framework used for the development of the RAG system, the selection of LlamaIndex was made due to its strong performance in retrieval tasks. It was faster and more efficient than other frameworks like LangChain and Haystack, while also being widely used, making it a better choice for building a scalable and responsive RAG system [11], [12].

B. RAG & Domain Specific question answering

Many approaches have been introduced to develop QA pipelines and research strategies for Retrieval-Augmented Generation (RAG), to tackle knowledge-intensive tasks. Lun-Chi Chen et al. [9] designed a novel customized model with a RAG-based LLM as a sustainable solution for industrial integration. Throughout the research, multiple retrieval configurations were evaluated, leading to the identification of the most optimal setup. The in depth comparison of the recent LLM based applications is used as bases for the selection of OpenAIs models for this project, for answer generation, reranking and evaluation of results. The scoring standard uses 5-point system, which is similarly introduced further in this project.

Further, Mateusz Płonka et al. [13] provided with an extensive research on the effectiveness of document splitters for large language models in legal contexts. With legal document being the most related to the political discussion that are being process throughout my research, Mateusz Płonka set the bases for the splitter that is used for chunking the congressional hearings. Given that the window-based splitter yielded the best results in the research, this approach was adopted for implementation of this pipeline. However, it is important to note that congressional hearings and policy discussions still have differences than the more structured legal texts. Unlike legal documents, which often present arguments in a concise and direct manner, policy discussions are typically more discursive, with ideas unfolding gradually across multiple speaker turns. To accommodate the difference structure, the window size was set larger than the optimal size identified during Płonka reseach, beginning at 4 sentences instead of 3. To experiment further and explore potential trade-offs, a 10-sentence window was also tested in this research. This longer window was tested to examine wether more context would help the system follow the flow of the discussion and better capture how arguments are developed.

The selection of the specified retrieval methods and embedding models was made following an extensive literature review on RAG pipeline development for domain-specific applications. These choices were informed not only by previously discussed sources, but also by additional studies—such as the work of Yanyan Lu et al. [9], who developed a RAG-based QA system for the Electric Power Industry. Although their approach did not experiment with multiple retrieval strategies, they effectively leveraged document structure by organizing information into tree-like knowledge hierarchies to support

more accurate response generation. In a similar manner, Sun et al. [13] proposed a domain-specific RAG pipeline designed to answer questions related to Pittsburgh and Carnegie Mellon University. Their system combined BM25 and dense vector retrieval, using a reranking mechanism to improve the relevance and contextual fit of the retrieved content.

III. Methodology

As the name suggests, the RAG pipeline is structured around three core components: Retrieval, Augmentation, Generation. These components are implemented using the LlamaIndex framework. This section presents the methodology used to develop the model, as illustrated in Figure 2, along with the prompting formats and the LLM-based evaluation approach.

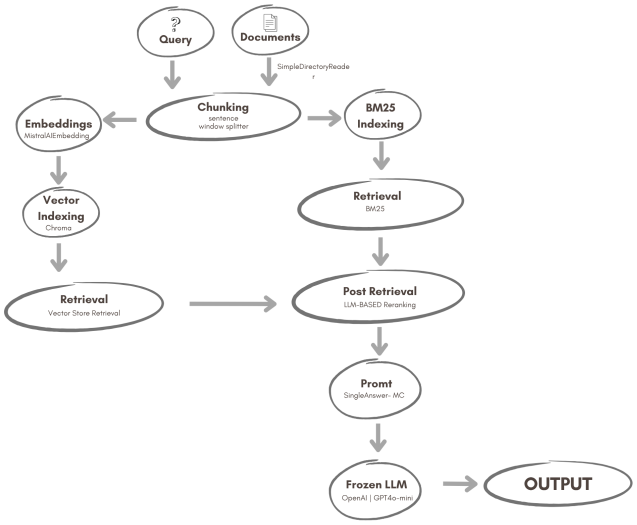

Fig. 2. Overview of the RAG-based Question Answering Pipeline.

Illustration of the flow from pre-processing to retrieval and generation using the LlamaIndex framework. Various retrieval methods (BM25, vector search, reranking) and window sizes are tested.

A. Framework

With multiple frameworks being available for the development of RAG models, with LlamaIndex and Langchain being the most known and used. LangChain is well-suited for integrating diverse document sources and chaining retrieval with generation, making it ideal for more complex applications. LlamaIndex, in contrast, follows a more lightweight and structured architecture, making it preferable for workflows with a smaller scope, such as text-based documents or hierarchical document structures [14]. This project uses LlamaIndex as the underlying framework for building the retrieval-augmented generation (RAG) pipeline. LlamaIndex handles key components such as document loading, sentence-window chunking, embedding generation, and indexing with Chroma. It supports both BM25 and dense vector retrieval, as well as post-retrieval reranking, allowing flexible experimentation across retrieval strategies. The framework was chosen for its modularity, ease of integration with OpenAI models, and its ability to manage custom prompt templates and query flows. Its design simplifies development while ensuring transparency and reproducibility in retrieval-based NLP tasks.

B. Retrieval & Augmentation

The retrieval and augmentation components of the system are responsible for narrowing a large set of congressional hearing transcripts into a smaller, relevant subset of passages and transforming these into a structured format suitable for use during the generation phase.

Initially, the transcripts in PDF format are loaded into the pipeline using the ‘SimpleDirectoryReader‘ class, converting them into LlamaIndex document objects. These documents are then preprocessed and segmented into smaller chunks to facilitate retrieval. Each transcript is chunked into smaller segments, using a sliding window approach over the sentences, to maintain contextual continuity and improve retrieval quality. Four different chunking strategies were tested, using 4, 5, 6, and 10 sentences per chunk, to evaluate the impact of chunk size on retrieval relevance and performance.

Once chunked, these segments are transformed into high-dimensional vector representations using the MistralEmbedding model [16] via the LlamaIndex API, which is designed to produce semantically meaningful embeddings. The resulting vectors are stored and indexed in Chroma, a persistent vector database optimized for fast semantic search.

Retrieval methods are then applied to narrow down the dataset to passages most relevant to a given user query. Two types of retrieval strategies were employed: sparse and dense. Sparse retrieval is a word-based method, relying on term frequencies and inverted indexes, focusing on exact keyword matches. Since sparse retrievers operate through exact-matching between query and document terms, they do not suffice when there is a semantic gap between the query and the corpus language. Dense retrieval, by contrast, uses contextualized pre-trained transformer models to map queries and documents into a shared embedding space. It applies cosine similarity in the embedding space to identify semantically similar chunks, even when the exact terms do not appear.

As most sparse ranking methods typically employ BM25, it was similarly adopted in this project. BM25 is a traditional retrieval algorithm which ranks a set of documents based on the query terms appearing in each document, regardless of their proximity within the document. Further, for the dense retrieval, a Vector Store Retrieval was implemented, which uses cosine similarity to retrieve semantically similar chunks, even when exact keywords are not matched.

To further refine the retrieved results, a post-retrieval LLM-based reranking stage is applied. In this step, top passages retrieved by both BM25 and vector search are merged, and the LLM is used to select the most relevant passage based on the

query. For this task, OpenAI's GPT-4.5 Turbo is employed as it currently represents its most capable and reliable model.Given the importance and dificulty of accurate reranking in the pipeline, selecting the best available model ensures high-quality, contextually grounded passage selection, resulting to a better response. This reranking enhances the likelihood of grounding the final answer in accurate and relevant evidence, especially in cases where dense retrieval returns semantically close but contextually irrelevant results.

These retrieval configurations — including chunk size, retrieval method, and the use of reranking — form the basis of the experimental design and evaluation used in this study.

C. Generation

After the retrieval of the most relevant passages from the indexed congressional transcripts, the system proceeds to the generation phase. The pipeline employs OpenAI's GPT-4o-mini as its frozen LLM [6], to ensure consistent results across configurations, as it was design to offer strong performance, low latency, and cost efficiency, making it suitable for experimentation at scale. Its API is well-documented, reliable for RAG applications and easy to integrate with LlamaIndex [7].

The transition from retrieval to generation uses the top-ranked passages retrieved from the BM25, vector search, reranked, or a combination of these methods. The top-k retrieved passages is always set to 3 throughout the experiment, due to high computational cost, however, it can easily be changed and tested as it is a global parameter. The selected passages, along with the user query, are formatted into a predefined prompt template and are inserted in the LLM, that utilize them to generate the response.

The input to the LLM consists of:

- The user question, which is either open ended, or provide possible answers in the form of multiple choice.
- Top 3 retrieved passages.
- A structured prompt, which can be either open-ended or single-answer multiple-choice, depending on the type of question.

To avoid hallucinations, the model is explicitly instructed to generate responses grounded entirely on the context of congressional hearings, without relying on external knowledge or assumptions. Given that the LLM is frozen, all adaptation to the congressional domain happens in the retrieval phase. In cases where the retrieved passages do not contain sufficient information to answer a users query, the RAG responds with: *"The answer is not available in the provided texts"*.

Overall, the generation stage is responsible for transforming retrieved evidence into high quality and answerable output - ensuring that the system produced reliable responses.

D. Prompting

Prompting has a critical role in shaping how the LLM responds to the query of the user. This determines not only the structure of the question posed to the model but also influences how the LLM interprets and synthesizes the input.

In this project, two prompting strategies are explored to evaluate their impact on performance:

- Open-ended prompting is phrased to encourage free-form explanation or summary generation, suitable for interpretive or exploratory analysis. This type of prompting is best suited for opinion-based questions, where the answer may require reasoning, contextual understanding, or synthesis of multiple ideas.
- Single-answer multiple-choice (MC) prompting restricts the model to select one correct option from a predefined list. These options are provided as part of the prompt and are designed for fact-based or closed-ended questions, where only one answer is correct based on the retrieved context.

By testing both prompt types across the retrieval configurations, the system allows for a controlled evaluation of prompt sensitivity—understanding how the format and structure of a prompt influence the model's ability to provide accurate and contextually faithful answers.

E. LLM-Based Prompt Evaluation

As part of the overall evaluation process, an automated LLM-based assessment is conducted on generated question–answer pairs. OpenAI's GPT-3.5 Turbo is used both for creating the questions and for evaluating the model responses. A different model is used compared to the answer generation and reranking stages, to avoid evaluation bias from the generation model itself and ensure more objective assessment. Each response is assessed for completeness, relevance, and faithfulness based on the retrieved context. The evaluation is automated through the mentioned prompts, and results are averaged to compare retrieval strategies and prompt types. This setup allows for consistent and scalable analysis without human annotation for every response. Further details on the complete evaluation will be provided in Section IV, which will also include human evaluation, annotation guidelines, and scoring procedure.

IV. Evaluation & Results

This section outlines the evaluation methodology used to assess the model across various parameters, and presents the results of the experiments split into LLM-based and human evaluation.

A. Experimental settings and system configuration

The system configuration for setting up the QA pipeline can be found summarized in Table I.

The strategies evaluated in this study include variations in chunk size—specifically chunks of 4, 5, 6, and 10 sentences—as well as different retrieval methods: sparse BM25, dense vector retrieval, and an LLM-based reranker that combines both. To maintain consistency and ensure cost and computational efficiency, the top-k value for retrieved passages is fixed at 3 (i.e., the top 3 relevant passages are always used) and the LLM model (GPT-4-Turbo) remains unchanged across all experiments. However, both parameters can be easily adjusted

Category	Component	Value
Retrieval	LLM chunk size	4–5–6–10
	Question Generation chunk size	10
	Top-k passages	3
	Embedding model	`MistralAIEmbedding`
	Vector Store	`ChromaVectorStore`
OpenAI models	Answer Generation: model	`gpt-4o-mini`
	Answer Generation: temperature	0.1
	Reranking: model	`gpt-4-turbo`
	Reranking: temperature	0
	Answer Evaluation: model	`gpt-3.5-turbo`
	Answer Evaluation: temperature	0

TABLE I
SYSTEM CONFIGURATION FOR RETRIEVAL AND OPENAI MODELS

within the implementation in case of replication or further experimentation.

To evaluate the mentioned retrieval strategies, two evaluation approaches were employed: LLM-based automatic evaluation and human expert evaluation.

The LLM-based evaluation methodology was analyzed in section III-E. It is used to automatically generate one question for every 10 sentences of the input transcript yielding to a total of 701 open-ended questions. Our model was then used to answer each generated question with the specified retrieval methods and chunk sizes. All LLM-based question-answer pairs are publicly available in the project's GitHub [15] repository.

For the human evaluation, a domain expert manually created a set of 24 questions, 9 open-ended and 15 multiple-choice. To provide a systematic evaluation and reproducibility the open-ended where manually retrieved from the LLM generated dataset. These were used to assess the model's ability to retrieve relevant information and provide accurate answers. All questions used can be found in A (Appendix)

With multiple-choice questions being more straightforward, they were manually evaluated quantitatively: a correct answer was awarded 2 points, no answer received 0 points, and an incorrect answer resulted in a deduction of 1 point. Open-ended responses however, required qualitative assessment. These were evaluated given three dimensions: completeness, relevance, and faithfulness, as outlined in Table 3. Each criterion was rated on a 5-point scale using a predefined rubric, provided in A (Appendix) . For the automated evaluation, the LLM was tuned to score the responses given the same rubric, while also providing an explanation on the scoring. In the human evaluation, a domain expert manually rated each answer.

B. LLM-based automatic Evaluation

1) Open-ended Questions: After completing all interactions with the RAG system across the various retrieval configurations, a total of 701 responses were evaluated using the three defined metrics. The aggregated results are presented in Table IV, which reports the scores for each metric across all tested retrieval methods and chunk size.

Completeness	Evaluates how fully the generated answer addresses the user's question using specific information from the provided context.
Relevance	Assesses how well the generated answer stays focused on the user's original query.
Faithfulness	Measures whether the answer stays true to the facts provided in the context.

TABLE II
DEFINITIONS OF EVALUATION METRICS FOR GENERATED ANSWERS

Retrieval Method	Chunk Size	Completeness	Relevance	Faithfulness
BM25	4	3.09	3.63	3.19
	5	3.25	3.84	3.42
	6	3.24	4.80	3.33
	10	3.50	4.13	4.60
Vector Store Retrieval	4	3.38	4.00	3.52
	5	3.56	4.21	3.68
	6	3.56	4.20	4.64
	10	3.63	4.28	4.74
LLM-based reranking	4	3.56	4.22	3.68
	5	3.78	4.68	3.81
	6	3.67	4.33	3.76
	10	3.95	4.8	4.82

TABLE III
SUMMARY OF EVALUATION METRICS ACROSS RETRIEVAL STRATEGIES AND CHUNK SIZES

Across all metrics, LLM-based reranking consistently outperforms both BM25 and vector retrieval, with the 10-sentence chunk size having the best results. This configuration answered a total of 691/701 questions, being unable to provide answers to 10 of them. Overall, performance appears to also increase slightly with chunk size, suggesting that larger windows preserve more context, especially for semantic retrieval, however this does not happen with the window size of 6, which will be discussed and analyzed further in Section VI-A

C. Human Evaluation

As mentioned the human evaluation is separated in open-ended and multiple choice questions.

1) Open-ended Questions: A total of nine questions were being retrieved from the LLM generated questions, three from each congressional hearing and can be found on A (Appendix). The model was then used with each retrieval configuration to answer each question. Each answer was then scored by three different the domain expert, based on the rubric, which aligns with the one the LLM is using for the grading.

The evaluation of all three experts for the mentioned open-ended questions can be found on the provided Github.

2) Multiple-Choice Questions: For the multiple-choice (MC) evaluation, the system was assessed on its ability to correctly select one answer from a set of four expert-provided options. A total of 15 MC questions were generated by a domain expert. The system was tasked with answering each of them across all retrieval configurations. The complete results of the evaluation can be found in Appendix A.

Retrieval Method	Chunk Size	Completeness	Relevance	Faithfulness
BM25	4	3.44	5.00	4.11
	5	3.74	5.00	4.19
	6	3.81	5.00	4.41
	10	4.37	5.00	4.89
Vector Store Retrieval	4	3.41	4.89	3.93
	5	3.89	5.00	4.41
	6	3.78	4.85	4.11
	10	4.33	4.56	4.56
LLM-based reranking	4	4.07	5.00	4.44
	5	3.93	5.00	4.33
	6	3.96	5.00	4.37
	10	4.56	5.00	5.00

TABLE IV
ROUNDED AVERAGE SCORES BY RETRIEVAL METHOD AND CHUNK SIZE FROM EXPERTS EVALUATION

To complement this, Figure 4 presents a heatmap that visualizes the accuracy rates achieved across different retrieval strategies and chunk sizes. The y-axis denotes the chunking strategies (4, 5, 6 and 10 sentences), while the x-axis lists the retrieval methods — BM25, vector store retrieval, and the LLM-based reranker. The color intensity in each cell corresponds to the accuracy percentage, providing an intuitive overview of how performance varies across configurations.

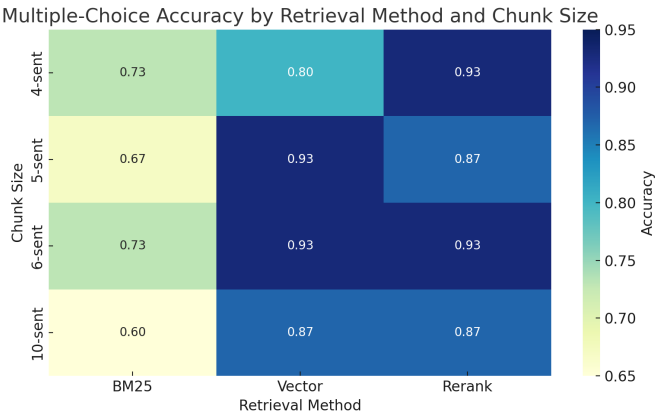

Fig. 3. MCQ Accuracy Heatmap Across Retrieval Strategies and Chunk Sizes.

The heatmap in Figure 3 illustrates a noticeable improvement in responses with 5- or 6-sentence chunks compared to those with 4, likely due to the added contextual information. Interestingly, the performance at the 10-sentence chunk size fell over for all retrieval methods, suggesting that longer contexts do not necessarily improve the answer correctness in this type of questions.BM25 method showed a further drop in accuracy at this size, reinforcing its limitations with longer passages where keyword precision is diluted. Moreover, the overall largest performance spread is observed in the hard questions, while performance between easy and medium questions remains relatively similar.

V. Responsible Research & Ethical Implications

With the growing use and influence of automated QA systems to its users, it is essential to set ethical principles through the design and evaluation process of the system. This section outlines the ethical considerations addressed in this study, from the data retrieved, data sensitivity and the responsible use of the provided results.

A. Ethical Use of Data

All collected data used in the project is public, sourced from official U.S. Congress platform [**?**]. Given that the congressional hearings are from a public domain, as per U.S law (17 U.S.C. §105), they are available unrestricted for any academic and research use. Congressional representatives and invited speakers are required to agree on the publication of these hearings. As such, their contributions are part of the public legislative record and do not require additional consent for reuse in analytical research. No sensitive personal data or private information is included in the dataset.

B. Mitigating Hallucinations and Ethical Reflections on Use

To ensure the reliability of generated responses, the system is designed to explicitly generate answers based on retrieved passages from congressional hearing transcripts. In this way, hallucinations are reduced to prevent fabrication of unsupported claims, or unrelated and biased context. By using a frozen LLM and enforcing strict retrieval constraints, the pipeline aims to promote verifiability of the information and maintain a clear trace from answer to source.

Despite the safeguards that this model provides, LLMs in general still have a risk of misinterpretation or misuse in sensitive policy domains. If used without proper oversight, generated summaries or answers could oversimplify complex debates or omit minority viewpoints. Therefore, this work emphasizes transparency in both retrieval and prompting stages, and it is intended to support, and not replace, expert analysis or manual research of congressional hearings.

C. Research Reproducibility

To maximize the reproducibility of the study, efforts of three aspects have been made: All source code for the development and evaluation of the QA pipeline, together with the requirements file been open-sourced and hosted on public repositories on Github [15]. Hence, the project can be easily rebuilt. All parameters that can be tuned, such as the number of chunks, the top-k retrieved passages, and the OpenAI LLM model, have been set to global, and can be easily changed at the beginning of the notebook, allowing for further evaluation of the model. Furthermore, a detailed README file is provided for easier understanding of the code-base and the integration of the model. The study protocol is also documented with detail.

It should be noted, that since the project relies on third-party APIs, such as OpenAI and Mistral, for embedding generation and language modeling, it is required for the user to create individual accounts which often involve usage costs. Therefore, even if the system's structure and settings are

completely reproducible, certain parts rely on premium APIs that the user must manually integrate and configure.

VI. Discussion

This section analyzes how retrieval and generation strategies affect answer quality—specifically completeness, relevance, faithfulness for open-ended questions, and accuracy for multiple-choice formats. It also highlights current limitations, such as fixed context windows and static evaluation, and outlines future improvements to better capture the complexity of political discourse.

A. Reflection on Results

1) Reflection of retrieval method and Chunk size on Open-ended questions: The analysis of open-ended question responses revealed a clear and consistent pattern in the LLM-based evaluation: both retrieval method and chunk size significantly influenced performance. Larger chunk sizes, especially 10-sentence windows, led to the best overall results across all metrics. This outcome underscores the benefit of extended context, which allows the system to retrieve passages that capture the full scope of a speaker's point or explanation, rather than isolated facts. However, an intresting result was the transition from 5- to 6-sentence chunks. The results yielded did not show the expected improvement—performance remained unchanged for vector retrieval (completeness held at 3.52, faithfulness at 4.33), and even slightly dropped for reranking, where completeness decreased from 3.53 to 3.47 and faithfulness from 4.45 to 4.35.. One possible explanation is that the 6-sentence chunks may have occasionally included partially relevant or distractor material that diluted the main answer, making it harder for rerankers to prioritize the most answer-relevant content, given that more answers were actually answered with the 6-chunk size. While BM25 also benefited marginally from larger chunks (faithfulness rising from 4.04 to 4.33), it remained less competitive than vector-based methods, as it lacks the ability to leverage semantic structure.

In the human evaluation, similar trends were observed, reinforcing the LLM-based results. Human raters consistently favored responses retrieved with 10-sentence chunks, especially those reranked using LLMs, citing improved completeness, better flow, and more accurate context. Compared to 5-sentence chunks, completeness increased by approximately 6% points on average , while faithfulness saw an even larger relative gain. Interestingly, humans also found 6-sentence responses slightly less complete than expected, mirroring the results seen in the automatic evaluation. This suggests that beyond a certain threshold, chunk size must balance between providing enough context and avoiding irrelevant content. In shorter chunks (4–5 sentences), responses often lacked sufficient background or continuity to fully ground the answer, while mid-range chunks (6 sentences) sometimes introduced content overlap or noise without a corresponding semantic gain. Overall, both evaluations confirmed that 10-sentence chunks combined with reranking consistently yield the most faithful and complete open-ended responses, while intermediate chunk sizes may suffer from a lack of contextual focus or signal dilution.

2) Reflection of retrieval method and Chunk size on Multiple-choice questions: The inclusion of MCQs in the evaluation process offered a structured and more objective way to compare retrieval strategies under controlled conditions. The format allowed for clear scoring and revealed sensitive variation in outcomes across chunk sizes and retrieval methods. For instance, models using dense vector retrieval and LLM-based reranking generally performed better with longer chunks (5–6 sentences), benefiting from increased contextual information. However, BM25 exhibited a drop in accuracy for the chunk size of 5 sentences and remained stable for the 6 sentences, likely due to its dependence on exact keyword matches, which become harder to find in longer text chunks. Interestingly, a decrease in performance was observed also for the reranker at the 5-sentence chunk size, a result that was unexpected given the general trend as well as the score of the vector retriever. One possible explanation is that the LLM used for reranking may have made suboptimal passage selections, excluding relevant content that dense retrieval retained.

Interestingly, open ended questions performance decreased at the 10-sentence chunk size across all retrieval methods. This result contrasts with the open-ended evaluation setting, where 10-sentence chunks led to the best performance. A likely reason is that MCQs often depend on pinpointing exact facts or keywords, which become less accessible in larger chunks as relevant information is diluted among more context. Additionally, longer chunks could overwhelm the reranking models with excessive or tangential information, leading to lower precision in selecting answer-relevant spans. Another possibility is that with longer chunks, the chance of "distractors" or off-topic sentences increases, potentially misleading both the retriever and the reranker. Overall, however, the system demonstrated strong performance, achieving high accuracy scores, indicating that the underlying retrieval and reasoning mechanisms are effective. It should be mentioned however, that despite its value for benchmarking, the MCQ format is of limited practical utility for end users. This is because answering MCQs with the system requires the user to supply not only the correct answer but also plausible "distractors", which undermines the intended purpose of automated question answering.

B. Limitations

1) Retrieval Scope Constraints: This project was conducted within a limited time frame of approximately 8 weeks, with development, experimentation, and analysis performed by a single researcher. As a result, the breadth of configurations tested was necessarily constrained. Only three retrieval strategies (BM25, dense retrieval, and reranking) and four chunk sizes (4, 5, 6 and 10 sentences) were evaluated. Other potentially impactful factors — such as variations in the top-k parameter, passage overlap rates, or embedding models — could not be extensively explored and were set to be constant through the project. Furthermore, the use of commercial APIs (e.g.,

OpenAI and Mistral) introduced cost limitations, restricting to high extend the number of prompt variations and evaluation rounds.

These constraints also limited the ability to experiment with newer or more specialized model features (e.g., adaptive reranking or custom embeddings), which may have improved performance but fell outside the project's practical scope.

2) Evaluation Bias and Subjectivity: This project employed both automated LLM-based scoring and human evaluation to assess the quality of open-ended responses. However, subjectivity of scoring open-ended questions could not be entirely eliminated. Even if the rubric is clearly defined and quite strict on the scoring, still leaves room for interpretation, especially in metrics like completeness where responses can but accurate but miss peripheral details.

Automated evaluation using ChatGPT-3.5 Turbo was chosen for its consistency and efficiency. Nonetheless, it still inherits its own reasoning patterns and biases, which potentially impact fairness and objectivity of scoring. Human evaluation, while more nuanced, was conducted by three domain experts — answering a total of 10 questions out of the 701 LLM-generated. While this improves reliability compared to a single evaluator, some degree of subjectivity remains, as individual interpretations and expectations may still influence scoring, particularly for open-ended responses.

3) Participant Recruitment: Human evaluation in this study was conducted by three domain experts familiar with both the content and the pre-defined evaluation rubric. While the use of different annotators improves the reliability of results compared to single-rater assessments, time constraints and limited resources restricted their involvement.

As mentioned, only 9 out of the 701 LLM-generated question–answer pairs were manually evaluated, rather than the full dataset This limited sample size constrains the depth of comparison between human and automated evaluation and may not fully capture broader trends in model performance. However, no broader participant group (ie. policy analysts, researchers, or general users) were involved in testing and analyzing further the question-answering system. As such, the pipeline's usability, perceived value, and accessibility remain untested in real-world settings.

C. Implications and Future Work

As intended, the research project demonstrates the feasibility and potential of RAG systems using the LLamaIndex framework in processing U.S. congressional hearing transcripts through a QA pipeline. By combining traditional and semantic retrieval techniques with prompting strategies, the pipeline effectively delivers structured answers grounded in official legislative documents. These findings highlight the opportunity for deploying RAG-based systems in civic, policy, and legal contexts—particularly where transparency, factual grounding, and query complexity are needed. Of-course, for the sake of experimentation more retrieval methods could be used, to asses the best combination for such case.

Further on, the chunk size of 10 sentences evaluated to be better than the smaller ones, preserving more context. However, this could dilute term precision, particularly for sparse retrievers. Future systems could dynamically adjust chunk size based on document structure or question type, improving retrieval across varied legislative formats as every type of text may contain required information in a smaller or larger chunk size. Additionally, the use of multiple retrievers—along with LLM-based reranking—proved promising, but not without flaws. In some cases, the reranker excluded relevant passages. Implementing another reranker such as a Cross-Encoder reranker could be useful, to compare the results of the two.

The evaluation framework, while effective, focused on technical and content-alignment metrics such as completeness, faithfulness, and relevance. Future work could extend this with user-centered such as utility, or clarity, to better understand how these systems perform in practical environments. Moreover, while human evaluation added depth, broader user testing is necessary to see the confidence that a stakeholder has while using it, and how much trust there is towards the model.

Finally, future work should expand the dataset beyond the three transcripts used in this study. Incorporating a broader range of hearings—such as budgetary, investigative, or informal sessions—will help evaluate generalization across structural and thematic variation. Moreover, the development of a shared benchmark for legislative QA tasks would allow for standard comparisons and foster collaboration in this emerging application area.

VII. Conclusion

This study examined the development and evaluation of a Retrieval-Augmented Generation (RAG) question-answering pipeline for the context of US congressional hearings. In order to assess the impact of various retrieval methods and chunking sizes on the quality of generated answers, the project combined a frozen LLM for response generation with retrieval strategies BM25, Vector Store Retriever, and LLM-based reranking. The results showed that retrieval strategy and chunk size significantly affect the faithfulness, completeness, and relevance of responses using both LLM-based and human evaluation techniques. The best results in open-ended question settings were obtained specifically with LLM-based reranking and 10-sentence chunks, demonstrating the importance of semantically rich retrieval in conjunction with extended context. However, multiple-choice questions that contained five to six sentence chunks performed the best, highlighting the significance of maintaining accuracy and reducing noise in keyword-based queries. Additionally, the project produced a reproducible and modular pipeline using LlamaIndex that can be used as a model for future RAG-based systems in the policy and civic domains. The findings were strengthened by the inclusion of both automated and manual evaluation, and the validity of the evaluation framework is supported by the observed agreement between expert judgements and LLM-based scores. Even though the system worked well in the majority

of configurations, there are still some issues, especially with scalability, dataset diversity, and end-user validation. To better understand the system's usability and reliability in practical policy applications, future research could build on this work by implementing adaptive chunk sizing, sophisticated reranking techniques (like cross-encoders), and more thorough user studies. In conclusion, this study establishes the groundwork for future research into LLM applications in structured, domain-specific text understanding and shows that RAG pipelines are both feasible and efficient for legislative QA tasks.

REFERENCES

[1] H. Naveed *et al.*, "A comprehensive overview of large language models," *arXiv preprint*, Jul. 2023. [Online]. Available: https://arxiv.org/abs/2307.05772

[2] W. Fan *et al.*, "A survey on RAG meeting LLMs: Towards retrieval-augmented large language models," *arXiv preprint arXiv:2307.06435*, Jul. 2023. [Online]. Available: https://arxiv.org/abs/2405.06211

[3] P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020. [Online]. Available: https://www.scopus.com/record/display.uri?eid=2-s2.0-85108449607

[4] M. V. Lee Badgett and M. J. Malbin, "Vital statistics on Congress," Brookings Institution, Nov. 2024. [Online]. Available: https://www.brookings.edu/wp-content/uploads/2024/11/6-1-Full.pdf [Accessed: Jun. 19, 2025].

[5] Congress.gov, "Advanced search: Legislation – 118th Congress," U.S. Library of Congress. [Online]. Available: https://www.congress.gov/advanced-search/legislation?congresses%5B0%5D=118 [Accessed: Jun. 19, 2025].

[6] OpenAI, "GPT-4o mini: Advancing cost-efficient intelligence," Jul. 2024. [Online]. Available: https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence

[7] LlamaIndex, "Starter tutorial (using OpenAI)," *LlamaIndex Documentation*. [Online]. Available: https://docs.llamaindex.ai/en/stable/getting_started/starter_example/ [Accessed: Jun. 19, 2025].

[8] C. Ling *et al.*, "Domain specialization as the key to make large language models disruptive: A comprehensive survey," *arXiv preprint arXiv:2309.00623*, 2023.

[9] L.-C. Chen *et al.*, "Application of retrieval-augmented generation for interactive industrial knowledge management via a large language model," *Journal of Manufacturing Systems*, May 2024. [Online]. Available: https://doi.org/10.1016/j.jmsy.2024.05.001 [Accessed: Jun. 22, 2025].

[10] Y. Lu *et al.*, "A retrieval-augmented generation framework for electric power industry question answering," in *Proc. 2024 2nd Int. Conf. on Electronics, Computers and Communication Technology (CECCT '24)*, Jan. 2025. [Online]. Available: https://doi.org/10.1145/3705754.3705771 [Accessed: Jun. 22, 2025].

[11] SciPhi, "Benchmarking retrieval-augmented generation (RAG) frameworks: LangChain, LlamaIndex, and Haystack," *SciPhi.ai*, Apr. 22, 2024. [Online]. Available: https://www.sciphi.ai/blog/benchmarking [Accessed: Jun. 22, 2025].

[12] S. Shah and A. Kumar, "Scalability and performance benchmarking of LangChain, LlamaIndex, and Haystack for enterprise AI customer support systems," *ResearchGate*, Feb. 2024. [Online]. Available: https://www.researchgate.net/publication/383866832 [Accessed: Jun. 22, 2025].

[13] M. Płonka, Y. Zhang, and M. Grabmair, "A comparative evaluation of the effectiveness of document splitters for large language models in legal contexts," *Expert Systems with Applications*, vol. 239, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0957417425003331 [Accessed: Jun. 22, 2025].

[14] T. Tamanna, "LangChain vs. LlamaIndex: A Comprehensive Comparison for Retrieval-Augmented Generation (RAG)," *Medium*, May 12, 2024. [Online]. Available: https://medium.com/@tam.tamanna18/langchain-vs-llamaindex-a-comprehensive-comparison-for-retrieval-augmented-generation-rag-0adc119363fe [Accessed: Jun. 22, 2025].

[15] A. Nikoloudis, "Research-Project," *GitHub*, 2025. [Online]. Available: https://github.com/akanikoloudis/Research-Project [Accessed: Jun. 22, 2025].

[16] Mistral AI, "Embeddings Overview," *Mistral Documentation*, 2024. [Online]. Available: https://docs.mistral.ai/capabilities/embeddings/overview/ [Accessed: Jun. 22, 2025].

[17] U.S. Congress, "Congress.gov," *Library of Congress*, 2025. [Online]. Available: https://www.congress.gov/ [Accessed: Jun. 22, 2025].

APPENDIX

Congressional Hearing 1

- What steps does the drug-shortages staff take to ensure that manufacturers of critical medical products, such as generic sterile injectables like penicillin, are qualified and have appropriate processes in place to prevent issues like cross-contamination?
- What strides has the United States taken in strengthening its public-health infrastructure in response to the COVID-19 pandemic over the past 4 years?
- What potential consequences could arise from reducing funding to the National Institutes of Health (NIH) for essential research, particularly in the areas of pathogen surveillance, genomic sequencing, informatics, and clinical-trial network infrastructure, as a reactionary response to the hypothesis that the novel coronavirus originated from a laboratory incident?

Congressional Hearing 2

- How many inadmissible aliens have been reported to have been released into the country since January 2021, according to the House Judiciary report mentioned in the passage?
- How does the presence of large numbers of less educated, low-income illegal immigrants impact the economy's social services spending?
- What percentage of the agents' time is spent in their vehicles while fully operational, according to the passage?

Congressional Hearing 3

- What specific accusation was made regarding health consumption, and how is that characterized by the speaker in the passage provided?
- What potential issue is Mr. Stansbury raising regarding the use of government property for electoral politics in the passage?
- Do you think the high cost of electric vehicles, averaging nearly $57,000, poses a financial challenge for the average American family in the current economy?

TABLE V
LIST OF MULTIPLE-CHOICE QUESTIONS CATEGORIZED BY CONGRESSIONAL HEARING

Retrieval Strategy	Window Size	Correctness
BM25	4	11
	5	10
	6	11
Vector Store Retrieval	4	12
	5	14
	6	14
LLM-based reranking	4	14
	5	13
	6	14

TABLE VI
CORRECTNESS SCORES ACROSS RETRIEVAL STRATEGIES AND WINDOW SIZES FOR MULTIPE-CHOICE QUESTIONS