NUST Omega at RIRAG 2025: Investigating Context-Aware Retrieval and Answer Generation-Lessons Learned and Challenges

Huma Ameer[*], Muhammad Hannan Akram[*], Seemab Latif, and Mehwish Fatima[†]
School of Electrical Engineering and Computer Science (SEECS),
National University of Sciences and Technology (NUST),
Islamabad, Pakistan
[hameer.msds20seecs,makram.bsds23seecs,
seemab.latif,mehwish.fatima]@seecs.edu.pk

Abstract

NUST Omega participates in RIRAG Shared Task. Regulatory documents pose unique challenges in retrieving and generating precise and relevant answers due to their inherent complexities. We explore the task by proposing a progressive retrieval pipeline and investigate its performance with multiple variants. Some variants include different embeddings to explore their effects on the retrieval score. Some variants examine the inclusion of keyword-driven query matching technique. After exploring such variations, we include topic modeling in our pipeline to investigate its impact on the performance. We also study the performance of various prompt techniques with our proposed pipeline. With empirical experiments, we find some strengths and limitations in the proposed pipeline. These findings will help the research community by offering valuable insights to make advancements in tackling this complex task.

1 Introduction

Regulatory documents, issued by governmental bodies, define the rules and standards for legal compliance across industries. These texts are often lengthy and complex, requiring specialized expertise to interpret, with non-compliance carrying heavy penalties (News, 2023). Advancements in NLP have led to the emergence of Regulatory Natural Language Processing (RegNLP), a multidisciplinary subfield aimed at simplifying access to and interpretation of regulatory texts (Gokhan et al., 2024).

Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) leverages LLMs by integrating external knowledge sources, enabling up-to-date, domain-adaptable capabilities (Asai et al., 2023; Siddharth and Luo, 2024; Sahlman et al., 2023). The RIRAG shared task consists of two phases: (1) Retrieval and (2) Answer Generation. Accurate retrieval, crucial for effective generation, employs techniques like chunking, query expansion, metadata annotation (Setty et al., 2024; Zhang et al.), and topic modeling to segment regulatory texts for improved precision (Tran and Litman, 2024; Rezaei et al., 2024). Advanced prompting strategies, such as few-shot prompting (Wang et al., 2020) and Chain-of-Thought (CoT) (Wei et al., 2022), further enhance response quality in the generation phase.

In our approach to the RIRAG task, we explore multiple methodologies. We begin with metadata-based keyword retrieval and refine it using topic modeling for coherent segmentation. For answer generation, we leverage few-shot and CoT prompting to enhance accuracy and coherence. Our results emphasize the critical role of retrieval quality in boosting generation performance while highlighting limitations that pave the way for future research.

2 Progressive Retrieval Pipeline

We propose a pipeline, Progressive Retrieval Pipeline (ProReg), for this shared task by adopting an iterative and structured approach. Figure 1 illustrates the architecture of ProReg[1].

2.1 Retrieval

2.1.1 Embeddings

The effectiveness of a retrieval system is correlated with its embeddings, which encapsulates the semantic and contextual information of the text. So, we experiment with multiple embedding models to assess the retrieval performance: (1) OpenAI [2], (2) Gemini [3], and (3) LegalBERT (Chalkidis et al.,

[*] Equal contribution.

[†] Corresponding author: mehwish.fatima@seecs.edu.pk

[1] https://github.com/MehwishFatimah/NUST-Omega.git

[2] OpenAI: New Embedding Models and API Updates

[3] Gemini: Embeddings

Proceedings of the 31st International Conference on Computational Linguistics, pages 85–90
January 19–24, 2025. ©COLING 2025

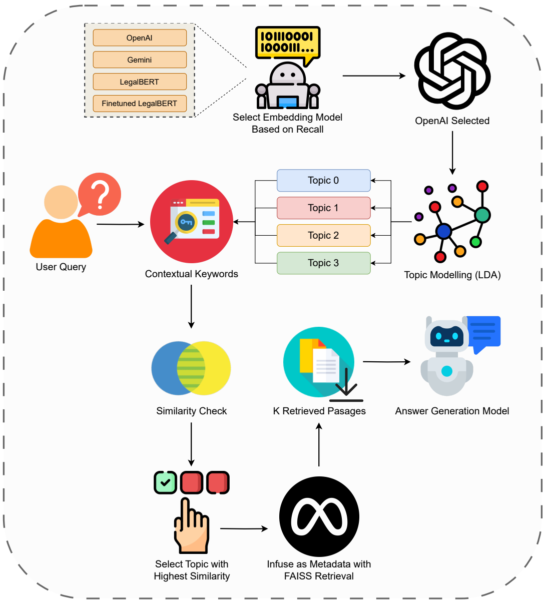

Figure 1: Progressive Retrieval Pipeline (ProReg)

2020). Building on Gokhan et al. (2024)'s fine-tuned LegalBERT embeddings (used for RePaSs), we also use these embeddings referred to as Fine-Tuned LegalBERT, in our experiments.

Retrieval: We use Facebook AI Similarity Search (FAISS) [4] for fast and dense similarity search.

2.1.2 Passage Filtering

The Structured document dataset contains 720 such instances where "Passage" were empty, and 1744 such instances in which passages consist of headings like "Introduction", "General", "Objectives" etc. Notably, they do not contextually contribute to the outcome,therefore, we test the best embedding model retrieval results by removing such passages which are less than equal to five words.

2.1.3 Metadata-Driven Query Matching

Keywords are extracted from each passage using KeyBERT (Grootendorst, 2020), and included as metadata. The rationale behind the exploration is to enhance the retrieval process by aligning query with the extracted keywords. We experimented with two approaches, firstly, the passages in the retrieval are considered if atleast one of the keywords matches with the query. Secondly, we use semantic similarity with varying thresholds *i.e.*, 0.5,0.7.

2.1.4 Retrieval with Topic Modeling

In our efforts to enhance the retrieval, we also explored a structured methodology by introducing topic modeling into the pipeline. Therefore, we conducted extensive experimentation with various parameters of BERTopic [5]and Latent Dirichlet Allocation (LDA) (Blei et al., 2003; Zoya et al., 2021) to segment the data into topics aiming to make the retrieval system more structured.

Next, the extracted contextual keywords of the passages under each topic are matched with the contextual keywords of the user query. Based on this approach, the topic with the highest score is identified. Subsequently, the topic includes meta data which is then included in FAISS retriever. It then ensures that relevant chunks are received. The steps are illustrated in Algorithm 1.

2.2 Answer Generation

In the answer generation phase, OpenAI's Generative Pre-Training Transformer (GPT) model is used and tested with three major prompting strategies. We experimented first with Simple Prompting by providing just initial instructions to answer the question. Then with Few-shot Prompting in which few examples are provided. Lastly, we explored CoT encouraging the model to break down its reasoning steps and structuring the generation process.

3 Experiments

3.1 Dataset

A comprehensive overview of the shared task and dataset are presented to Appendix C for brevity.

3.2 Models

BM25 serves a baseline for retrieval proposed by (Gokhan et al., 2024). In our initial experiments, embedding models including Gemini, OpenAI, LegalBERT, and fine-tuned LegalBERT are used alongside FAISS as a retriever. In the next iteration, keyword-driven methods such as Exact Match and Semantic Matching with OpenAI embeddings and FAISS retriever is explored. The results are presented in the table 1. For answer generation, the baseline combines BM25 for retrieval either using Passage-Only (PO) or Rank Fusion (RF) with GPT-4 for answer generation. Our experiment combines OpenAI embedding with FAISS retriever for retrieval and GPT with three Prompting techniques as shown in table 2.

[4] FAISS: Vector Database

[5] BERTopic: Github Link

Retrieval	**R@10**	**M@10**
Baseline		
BM25	0.76	0.62
Embeddings		
Gemini	0.68	0.09
OpenAI	0.71	0.09
LegalBERT	0.38	0.05
FT-LegalBERT	0.11	0.01
OpenAI + Pass.Filter	0.71	0.09
OpenAI(Unseen Ques)	0.58	0.09
Keyword-Driven Query Matching		
Exact Match	0.33	0.14
Semantic [0.7]	0.71	0.09
Semantic [0.5]	0.71	0.09

Table 1: Retrieval performance across Embeddings and Keyword-Driven Query Matching.

3.3 Evaluation Metrics

For the retrieval module, we use RIRAG shared task evaluation metrics (Gokhan et al., 2024). For retrieval, Recall@10 and Mean Average Precision (MAP@10) are used, and for answer generation, Regulatory Passage Answer Stability Score (RePASs) is used that combines entailment, contradiction and obligation coverage.

4 Results

4.1 Embeddings Impact on Retrieval

We evaluate multiple embeddings to identify the most effective one for the task and assess its impact on retrieval performance. Table 1 shows that OpenAI embeddings outperform other models, with recall@10 (R@10) and mean average precision@10 (M@10) as the evaluation metrics. After applying passage filtering, the differences in results are negligible. Notably, domain-specific embeddings like LegalBERT perform poorly. Additionally, we include the fine-tuned LegalBERT embeddings from the base paper in our experimentation, which yield suboptimal results.

Since OpenAI embeddings are trained on diverse and large datasets, it captures better respresentation of the text across various domains. However, it is worth noting that LegalBERT did not perform well and a potential reason could be that it may have been trained on specific legal jargon that is contextually different than the provided dataset.

4.2 Metadata-Driven Query Matching

To enhance retrieval results, we implement a metadata-driven query matching approach as outlined in Subsection 2.1.3. However, as shown in Table 1, the exact query matching method underperforms, and experiments with similarity scores fail to achieve significant improvements. Consequently, this approach proves ineffective for the task. It is noteworthy that in table 1, Exact Match refers to query keywords exactly matching passage keywords. Semantic [0.7] refers to passages retrieved based on semantic similarity with a threshold of 0.7. Lastly, Semantic [0.5] refers to passages retrieved based on semantic similarity with a threshold of 0.5.

Models	$\mathbf{E_S}$	$\mathbf{C_S}$	$\mathbf{OC_S}$	$\mathbf{Re}$
Baseline				
BM25(PO)+GPT-4	0.77	0.24	0.22	0.58
BM25(RF)+GPT-4	0.77	0.24	0.20	0.58
Prompting Method				
Few-Shot	0.53	0.16	0.11	0.49
CoT	0.49	0.23	0.19	0.49
Simple Prompt	0.45	0.17	0.15	0.48
CoT(Unseen Ques)	0.48	0.23	0.16	0.43

Table 2: Evaluation of Answer Generation.

The metadata keywords appear insufficiently informative for the retrieval task, and the embeddings may lack semantic richness specific to this subdomain. While these limitations are evident, it is premature to dismiss other potential avenues before resorting to model fine-tuning, which is resource-intensive. A logical next step involves leveraging contextual keywords with a more targeted approach and gaining a deeper understanding of the data to refine the retrieval process.

4.3 Prompting Strategies

Next, we evaluate different prompting strategies using OpenAI embeddings and FAISS as the retriever. Table 2 shows that few-shot prompting achieves the highest entailment score $(\mathbf{E_S})$, indicating its strength in maintaining factual consistency. However, Chain of Thought (CoT) prompting demonstrates improved obligation coverage $(\mathbf{OC_S})$ but results in the highest contradiction score $(\mathbf{C_S})$, reflecting the complexity introduced in its reasoning steps. Additionally, $\mathbf{Re}$ in Table 2 represents the overall relevance, which serves as a holistic measure of the prompt's effectiveness across these metrics.

The high contradiction score in CoT indicates that the model struggles to handle the complexity of the domain effectively. In contrast, the few-shot approach performs better as it introduces the model to domain knowledge through carefully se-

Retrieval	R@10	M@10
Simple	0.86	0.09
Keyword	0.31	0.05

Table 3: Retrieval performance comparison on a sampled test set.

Topic	0	1	2	3
Passages	3,544	2,252	2,512	2,960

Table 4: Distribution of passages across topics after segmentation.

lected examples. These examples consist of a few question-answering samples derived from the test set. Moving forward, a hybrid approach that balances the strengths of both techniques could enhance answer generation by leveraging structured reasoning from CoT while maintaining the contextual grounding of few-shot learning.

4.4 Retrieval with Topic Modelling and Contextual Keywords

We revisit the retrieval phase with a structured approach to address the lack of significant improvements in retrieval results. This iteration focuses on segmenting the dataset into distinct topics, identifying the probable topic of a query, and incorporating this information as metadata into the FAISS retriever. For dataset segmentation, we experiment extensively with topic modeling techniques, including BERTopic and Latent Dirichlet Allocation (LDA). Both LDA and BERTopic are evaluated using coherence scores and intertopic distance maps, testing various parameter combinations to optimize topic diversity and coherence, achieving a maximum coherence score of 0.41. The statistical method proves more effective for the given dataset, allowing us to segment the data into clearly defined topics, as illustrated in Figure 2.

It is important to highlight that passage filtering is a crucial step in the pipeline, as it prevents the grouping of duplicated passages containing common terms across different files. Without this step, passages with repetitive words, such as “Introduction”, would be incorrectly clustered into a single topic, negatively impacting the quality of topic modeling. By filtering out such passages, the pipeline ensures more accurate and meaningful topic differentiation.

The next step maps the query to the most relevant topic. Since LDA does not provide contextual topic terms, we extract contextual keywords for passages within each topic using GPT. To test the effectiveness of this structured approach, we select 50 passages from each topic, extract their contextual keywords, and use a sample of 50 questions from the test set. To validate this approach, we also evaluate the outcome of these questions without applying a contextual keyword filter. However, the results, as shown in Table 3, contradict our hypothesis, indicating no significant improvement in retrieval scores. Upon revisiting the data segmentation, although the four topics are distinct, the distribution of passages per topic in Table 4 suggests potential overlap and heterogeneity among passages.

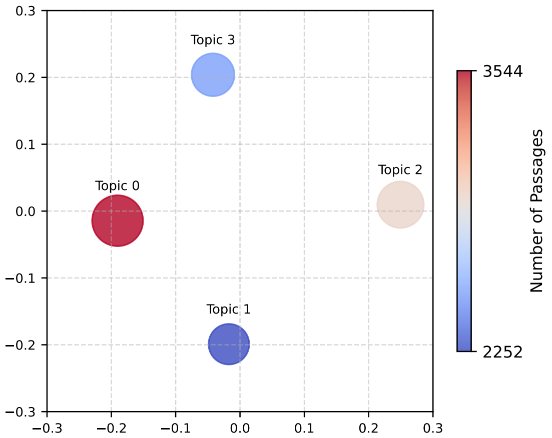

Figure 2: LDA Based Topic Modeling

5 Conclusion

In this study, we have explored the applicability of RAG for regulatory documents. We approach the task by systematically exploring the performance of embedding models, keyword supported query matching, and topic modeling in compliance with contextual keywords. Key Lessons from our experiments include the significance of embedding models with respect to the retrieval. The unsuccessful outcome of query matching led us to approach the problem by ingesting topic modeling in the pipeline. Moving forward, focusing on sub-topic modeling could provide deeper insights. Additionally, fine-tuning the model may improve performance, but experimenting with a more hierarchical RAG pipeline could unlock significant potential.

Acknowledgments

We sincerely thank the organizers and reviewers for their valuable contributions, constructive feedback, and support, which have significantly enhanced the quality of our work.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *Preprint*, arXiv:2310.11511.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Tuba Gokhan, Kexin Wang, Iryna Gurevych, and Ted Briscoe. 2024. Regnlp in action: Facilitating compliance through automated information retrieval and answer generation. *arXiv preprint arXiv:2409.05677*.

Maarten Grootendorst. 2020. Keybert: Minimal and easy keyword extraction with bert. Accessed: 2024-11-29.

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.

Bloomberg News. 2023. Rbc hit with fine for breaking canadian money laundering rules. Accessed: 2024-11-30.

Mohammad Reza Rezaei, Maziar Hafezi, Amit Satpathy, Lovell Hodge, and Ebrahim Pourjafari. 2024. At-rag: An adaptive rag model enhancing query efficiency with topic filtering and iterative reasoning. *arXiv preprint arXiv:2410.12886*.

WA Sahlman, AM Ciechanover, and E Grandjean. 2023. Khanmigo: Revolutionizing learning with genai. *Harvard Business School Case*, pages 824–059.

Spurthi Setty, Katherine Jijo, Eden Chung, and Natan Vidra. 2024. Improving retrieval for rag based question answering models on financial documents. *arXiv preprint arXiv:2404.07221*.

L Siddharth and Jianxi Luo. 2024. Retrieval augmented generation using engineering design knowledge. *Knowledge-Based Systems*, page 112410.

Nhat Tran and Diane Litman. 2024. Enhancing knowledge retrieval with topic modeling for knowledge-grounded dialogue. *arXiv preprint arXiv:2405.04713*.

Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. 2020. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E Gonzalez. Raft: Adapting language model to domain specific rag, 2024. *URL https://arxiv. org/abs/2403.10131*.

Zoya, Seemab Latif, Faisal Shafait, and Rabia Latif. 2021. Analyzing lda and nmf topic models for urdu tweets via automatic labeling. *IEEE Access*, 9:127531–127547.

A Limitation

The scope of this study is limited to basic RAG pipeline experiments, serving as a base to navigate to targeted approaches. It is also limited to the base embeddings of the models to assess their capabilities, however, a domain specific embedding has a potential to improve these results.

B Training Considerations

Our framework leverages pre-trained retrieval techniques to enhance efficiency, bypassing the need for custom model training. This approach reduces resource demands while ensuring high relevance for regulatory Question Answering tasks.

C Task and Data

The Regulatory Information Retrieval and Answer Generation (RIRAG) Shared Task is an interesting take-on advancing RegNLP which seeks to develop efficient systems for retrieval and precise answer generation from regulatory documents. The task consists of two sub-tasks: (1) Regulatory Information Retrieval primary focus is to retrieve passages with highest relevancy given the user query. (2) Regulatory Answer Generation refers to developing systems to generate concise and accurate answers. The authors, (Gokhan et al., 2024) introduces the Obligation-based Question Answering (ObliQA) dataset, derived from Abu Dhabi Global Markets (ADGM) financial regulations. The dataset consists of structured regulatory documents in json format making upto a total of 13,732 passages and 640,000 words. The synthetic question answer pairs are prepared which are validated by Natural Language Inference (NLI) and it uses nli-deberta-v3-xsmall model is used for semantic similarity.

D Algorithm for Enhancing Retrieval through Topic Modeling

Algorithm 1: Enhancing Retrieval through Topic Modeling with Cosine Similarity

```
Input: Dataset $\mathcal{D}$, Query $q$, Topic
       Modeling Method $T$ (LDA)
Output: Relevant Chunks $\mathcal{C}_{\text{relevant}}$
Step 1: Train Topic Model
Train the topic model $T$ on the dataset $\mathcal{D}$ to
  generate topics $\mathcal{T} = \{T_1, T_2, \dots, T_k\}$;
Step 2: Extract Contextual Keywords for
  Topics
foreach Topic $T_i \in \mathcal{T}$ do
    1. Retrieve passages $P_{T_i}$ associated
      with topic $T_i$ from the dataset $\mathcal{D}$;
    2. Use a pre-trained language model
      (e.g., GPT-4) to extract the most
      relevant contextual keywords $\mathcal{K}_{T_i}$ from
      the passages $P_{T_i}$;
    $$\mathcal{K}_{T_i} = f_{\text{LM}}(P_{T_i})$$
    Where:
      • $P_{T_i}$ are the passages for topic $T_i$,
      • $f_{\text{LM}}$ is the pre-trained language model (e.g.,
        GPT-4) for keyword extraction,
      • $\mathcal{K}_{T_i}$ are the relevant contextual keywords
        extracted for topic $T_i$.
end
Step 3: Extract Query Keywords and
  Compute Similarity
Extract contextual keywords $\mathcal{K}_q$ from the
  query $q$;
foreach Topic $T_i \in \mathcal{T}$ do
    Compute the similarity score $S(T_i, q)$
      using cosine similarity:
    $$S(T_i, q) = \frac{\sum_{k \in \mathcal{K}_{T_i} \cap \mathcal{K}_q} w_k^{(T_i)} \cdot w_k^{(q)}}{\sqrt{\sum_{k \in \mathcal{K}_{T_i}} \left(w_k^{(T_i)}\right)^2} \cdot \sqrt{\sum_{k \in \mathcal{K}_q} \left(w_k^{(q)}\right)^2}}$$
      Where:
      • $w_k^{(T_i)}$ is the weight (e.g., TF-IDF score) of keyword $k$
        in topic $T_i$,
      • $w_k^{(q)}$ is the weight of keyword $k$ in query $q$.
end
Step 4: Identify Best-Matching Topic
Find the topic $T^*$ with the highest similarity
  score:
    $$T^* = \arg\max_{T_i \in \mathcal{T}} S(T_i, q)$$
Step 5: Retrieve Relevant Chunks
Add $T^*$ as metadata to the FAISS retriever;
Retrieve relevant chunks $\mathcal{C}_{\text{relevant}}$ associated
  with $T^*$;
return $\mathcal{C}_{\text{relevant}}$
```