NUST Alpha at RIRAG 2025: Fusion RAG for Bridging Lexical and Semantic Retrieval and Question Answering

Muhammad Rouhan Faisal*, Faizyab Ali Shah*, Muhammad Abdullah*,
Shalina Riaz*, Huma Ameer, Seemab Latif, and Mehwish Fatima†
School of Electrical Engineering and Computer Science (SEECS),
National University of Sciences and Technology (NUST),
Islamabad, Pakistan
[rfaisal.bscs21seecs,fshah.bscs21seecs,abdullah.bscs21seecs,sriaz.bscs21seecs,
hameer.msds20seecs,seemab.latif,mehwish.fatima]@seecs.edu.pk

Abstract

NUST Alpha participates RIRAG and proposes FUSIONRAG that combines OpenAI embeddings, BM25, FAISS, and Rank-Fusion to improve information retrieval and answer generation. We also explore multiple variants of our model to assess the impact of each component in overall performance. The strength of fusionRAG comes from our rank fusion and filter strategy. Rank fusion integrates semantic and lexical relevance scores to optimize retrieval accuracy and result diversity, and filter mechanism remove irrelevant passages before answer generation. Our experiments demonstrate that FusionRAG offers a robust and scalable solution to automate regulatory document analysis, improve compliance efficiency, and mitigate associated risks. We further conduct an error analysis to explore the limitations of our model's performance.

1 Introduction

The RIRAG shared task advances Question Answering (QA) by challenging teams to develop models for accurate query responses over complex regulatory datasets. Our team aim to tackle key challenges in retrieval and reasoning while addressing limitations in existing techniques.

Despite advancements in information retrieval (IR) and answer generation, regulatory information remains underexplored. Research has enhanced retrieval-augmented generation (RAG) systems using tools like FAISS for efficient high-dimensional searches (Han et al., 2023; Douze et al., 2024; Krisnawati et al., 2024; George and Rajan, 2022), MiRAGDB for gene regulation (Desai et al., 2022), and Neo4j for modeling complex relationships in domains like social networks and recommendation systems (Miller, 2013; Hodler and Needham, 2022; Saad et al., 2023). Dense retrieval models like Contriever (Izacard et al., 2022) excel in semantic understanding but struggle with exact keyword matching, while sparse models like BM25 handle lexical matching well but falter with ambiguous queries (Finardi et al., 2024). Re-ranking methods, such as cross encoders, enhance contextual relevance, and innovations like HyDE enrich query generation for ambiguous inputs (Setty et al., 2024). Training strategies, like incorporating irrelevant documents, reduce bias and improve robustness. Adapter layers, such as linear adapters, fine-tune embeddings for task-specific precision in RAG (Liu, 2023; Shen et al., 2024; Jostmann and Winkelmann, 2024), though methods like ReAct (Reason + Act) show limited industrial applicability (Veturi et al., 2024; Huly et al., 2024; Yao et al., 2023).

The regulatory domain poses challenges due to complex compliance, evolving laws, and regional standards. FusionRAG addresses this by combining dense (FAISS) and sparse (BM25) retrieval models for nuanced text handling. Integrated with ChatGPT-3.5, it generates contextually relevant responses tailored to regulatory queries.

2 FusionRAG

Figure 1 illustrates our model[1], which integrates vector-based (FAISS) and text-based (BM25) retrieval methods to retrieve the most relevant passages. We use a custom rank fusion technique, combining FAISS for semantic relevance and BM25 for lexical matching, enhancing retrieval accuracy and diversity. An LLM-based prompt (GPT3.5 turbo (OpenAI, 2023)) filters the top-k passages, from which GPT3.5 Turbo generates contextually accurate answers, ensuring reliable responses for regulatory queries.

[*] Equal contribution.

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

[1] https://github.com/MehwishFatimah/Nust-Alpha.git

Proceedings of the 31st International Conference on Computational Linguistics, pages 79–84
January 19–24, 2025. ©COLING 2025

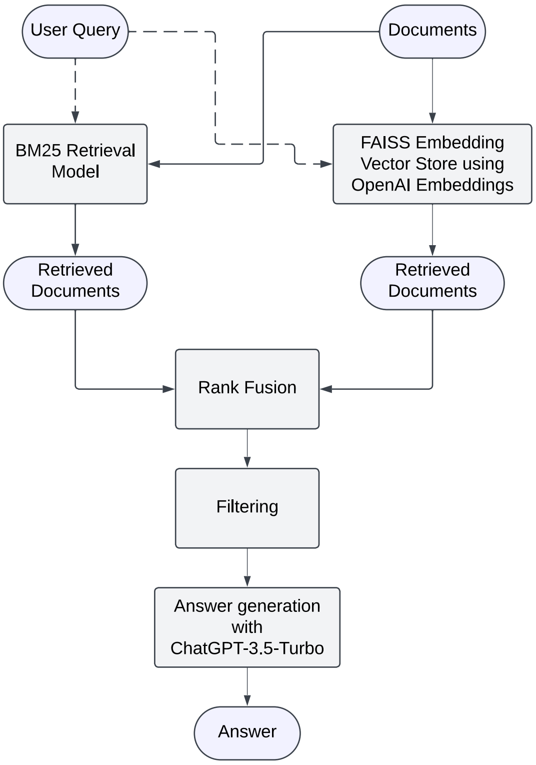

Figure 1: Architecture Diagram of FusionRAG

2.1 Retrieval

The process starts with a user query, which can either be a question or a topic of interest, forming the basis for information retrieval. We combine two approaches for this purpose: FAISS for vector-based retrieval and BM25 for text-based ranking.

FAISS: We use vector embeddings of OpenAI-text-embedding-3-large (OpenAI, 2022), enabling semantic-based retrieval. The query is passed to the FAISS retriever to perform similarity search and retrieve the top-10 most relevant documents. FAISS uses vector search to match the query against indexed document embeddings, returning a list of documents ranked by their relevance to the query, along with similarity scores.

BM25: All passages are indexed using BM25, a traditional information retrieval model. The query is processed by removing punctuation and stop words before being passed to the BM25 retriever. BM25 ranks documents based on term frequency and inverse document frequency, generating another set of relevant results.

2.2 Rank fusion

Rank fusion combines the strengths of multiple retrieval systems by aggregating scores from both FAISS and BM25. This unified ranking boosts the scores of highly ranked documents in both systems, addressing individual limitations like vocabulary mismatch in BM25 and embedding imprecision in FAISS. The fusion improves retrieval quality by prioritizing documents that perform well in both, reducing noise and enhancing diversity. This leads to more reliable results for tasks such as passage ranking and answer generation. We employ a custom scoring method for rank fusion, as described below:

$$S = (10 - R_{BM25}) + 0.8 \times (10 - R_{FAISS})$$

Where R_{BM25} denotes the rank of a document among those retrieved by BM25, while R_{FAISS} indicates the rank in the FAISS results. The document score ranges from a maximum of 18 to a minimum of 1.8. We do not normalize our scores as doing so would have no effect whatsoever on the ordering. BM25 maintains better ordering of the results as compared to FAISS, hence the decay of FAISS score by 0.8. This value is decided based on the results of development set.

2.3 Filtering

The filtering strategy involves using a relevance evaluation step to select the most pertinent passages from the top 10 retrieved by Rank Fusion. We design a prompt that instructs GPT-3.5 to assess which passages are relevant to the query, returning only the IDs of the relevant ones. If none are relevant, two passages are randomly selected. We use GPT-3.5 for both evaluating relevance and generating answers based on the selected context. This approach ensures the model operates within token limits while maintaining relevance and efficiency.

2.4 Generation

We use GPT-3.5 to create concise and contextually accurate responses based on the retrieved passages. Ensuring domain-specific relevance, prompts are carefully designed to include explicit instructions that guide the model in generating legal-context-aware answers. The prompts incorporate key legal terminology, a brief summary of the retrieved context, and specific tasks such as identifying obligations or providing clarifications, ensuring precision and alignment with user query. A fine-tuned legal-specific obligation classifier identifies obligation-related sentences within the passages and generated answers, enhancing their focus. A pre-trained natural language inference model evaluates the responses using entailment and contradiction scores to ensure logical consistency and alignment with the context. These scores, combined with an obligation coverage metric assessing the extent to which legal obligations are addressed, form a composite score that measures the reliability, consistency, and domain relevance of the generated responses.

3 Experiments

3.1 Dataset

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

3.2 Models

For the passage retrieval task, the baseline system uses BM25+GPT-4, a lexical-based retrieval model known combined with an LLM for answer generation. Additionally, a variation of the baseline uses BM25+RANK-FUSION (RF)+GPT-4 (BM25+RF+GPT-4) combining the lexical plus neural retrievers (Gokhan et al., 2024).

FusionRAG consists of OpenAI embeddings+ BM25+FAISS+Rank-Fusion (RF) for information retrieval. We also investigates some other variations of our pipeline, such as: (i) all-miniLLM-l6-v2+ FAISS: MINI+FAISS, (ii) all-miniLLM-l6-v2+BM25+FAISS+ Rank-Fusion (RF)+Reranker (R): MINI+FAISS+RF+R, and (iii) all-miniLLM-l6-v2+BM25+ Rank Fusion (RF): MINI+BM25+RF, to explore the impact of combining these approaches on model performance.

For text generation, we integrate FusionRAG with GPT-3.5 turbo: FUSIONRAG+GPT-3.5. We also investigate other variants: (i) FusionRAG with Gemini Flash: FUSIONRAG+GEMINI, and (ii) FusionRAG with LLaMA 3.1-8B: FUSIONRAG+LLAMA. These variants help to evaluate the impact of different generation models on the system's overall performance.

3.3 Evaluation Metrics

We use MAP@10 and RECALL@10 for passage retrieval, and RePaSs (Re) for answer generation (Gokhan et al., 2024).

4 Results

4.1 Retrieval Performance

Table 1 presents the results of retrieval models for RECALL@10 and MAP@10, calculated on the unseen dataset consisting of 446 questions.

The results highlight the significant performance improvements achieved by FusionRAG. FusionRAG outperforms BM25 with RECALL@10 = 78.2 and MAP@10 = 63.4. This is a remarkable increase over the baselines that demonstrates the robustness and impactfulness of FUSIONRAG. By integrating FAISS, a highly efficient similarity search algorithm, with BM-25, FUSIONRAG successfully captures nuanced query-document relationships, resulting in superior retrieval performance. The fusion of these retrieval strategies allows FUSIONRAG to maintain high efficiency while enhancing its ability to understand deeper semantic connections between queries and documents. Moreover, the addition of CHATGPT-3.5 as a sophisticated filtering mechanism further refines the retrieved results. This filtering step ensures that only the most relevant passages are retained, discarding those that do not contribute meaningfully to the query, thus boosting precision and reinforcing the overall performance of FUSIONRAG.

Models	**RECALL@10**	**MAP@10**
BM25 passage-only	64.2	50.9
BM25+RF	64.2	51.0
MINI+FAISS	49.1	31.2
MINI+FAISS+RF+R	72.4	49.1
MINI+BM25+RF	72.4	61.2
FUSIONRAG	78.2	63.4
FUSIONRAG	67.2	52.1

Table 1: Performance Comparison of Retrieval Models. The last row presents the results from organizers.

Additionally, the results from Team Alpha offer further insights into retrieval performance, demonstrating a RECALL@10 = 67.2 and MAP@10 = 52.1. While these figures fall below FusionRAG's benchmarks, they provide a valuable comparative baseline for understanding the efficacy of other retrieval methods in this shared task. These results underscore the challenges faced in designing retrieval systems that effectively balance semantic understanding with precision, further validating the innovations embedded in FUSIONRAG.

4.2 Generation Performance

Table 2 compares the performance of two baseline methods: BM25+GPT-4 and BM25+GPT-4 rank fusion (RF) against FusionRAG and its variants: FUSIONRAG+GPT-3.5, FUSIONRAG+GEMINI, and FUSIONRAG+LLAMA. All evaluations were conducted on the unseen dataset.

BM25+GPT-4 achieves a REPASS score of 0.58, demonstrating its strong capability for retrieving relevant passages, as evidenced by its high entailment score of 0.77. However, its moderate OBLIGA-

Models	E_s	C_s	OC_s	RE
BM25+GPT-4	0.77	0.24	0.22	0.58
BM25+RF+GPT-4	0.77	0.24	0.20	0.58
FUSIONRAG+LLAMA	0.25	0.58	0.09	0.26
FUSIONRAG+GEMINI	0.27	0.49	0.13	0.32
FUSIONRAG+GPT-3.5	0.58	0.15	0.13	0.52
FUSIONRAG+GPT-3.5	0.50	0.11	0.10	0.50

Table 2: Comparison of Answer Generation Performance (Unseen Data). The last row presents the results from organizers.

TION COVERAGE score of 0.22 and CONTRADICTION SCORE of 0.24 indicate potential inconsistencies in the retrieved information, where conflicting details may undermine the coherence and reliability of the generated responses.

In comparison, FUSIONRAG+GPT-3.5 achieves a slightly lower RePASs score of 0.518. Despite this, its results reflect a more focused and precise retrieval strategy. With an obligation coverage score of 0.13 and a lower contradiction score of 0.15, FUSIONRAG+GPT-3.5 prioritizes accuracy and coherence over broad coverage. This tradeoff ensures that only the most relevant and consistent passages are included, thereby minimizing the introduction of conflicting or irrelevant details. Consequently, while its overall REPASS score is slightly reduced, its commitment to maintaining accuracy and relevance establishes it as a reliable choice for scenarios where precision is crucial. The results from Team Alpha add additional context, showcasing a REPASS score of 0.498 alongside an entailment-score of 0.505 and a contradiction score of 0.109. These results highlight the nuanced differences in retrieval performance across various methods, emphasizing the challenges in balancing obligation coverage (0.098) with overall coherence and relevance. These findings validate the importance of carefully designed retrieval strategies, such as those employed by FUSIONRAG, to achieve optimal results in both consistency and precision.

4.3 Error Analysis

We conduct an in-depth error analysis on 446 unseen questions to identify Hybrid Framework's limitations. The system successfully generates answers for 192 questions but fails for 254 due to a retrieval filter blocking irrelevant passages. This demonstrates that FusionRAG's performance heavily depends on the quality of the retrieval process, as it cannot generate answers without retrieving relevant passages.

Manual Analysis: We find a clear distinction between answered and unanswered questions. Answered questions are typically more specific with clear contextual cues, referencing regulatory guidelines or domain-specific concepts such as "ADGM", "compliance", or "authorised". These factors facilitate the retrieval of relevant passages and eventually enable accurate response generation. While, unanswered questions are often more general or abstract lacking sufficient context, containing vague terms like "could" or "under what circumstances". Many of such queries also pose hypothetical scenarios, complicating the retrieval process and limiting the model's ability to generate responses.

Topic Modeling: To explore further, we use LDA to uncover topic patterns in the questions for generating five topics for answered and unanswered questions. LDA reveals distinct patterns in topic distributions illustrated in Figures 2 and 3 in Appendix D.

In summary, the error analysis highlights the critical role of specificity and contextual clarity in determining the model's success. Answered questions tend to be grounded in actionable, domain-specific information, whereas unanswered questions are broader, theoretical, or vague. To improve performance, we recommend enhancing the retrieval process to handle abstract and hypothetical queries more effectively while refining the model's ability to interpret less specific questions.

5 Conclusions

We present FUSIONRAG for the Regulatory Information Retrieval and Answer Generation (RIRAG) Shared Task, combining OpenAI embeddings,SmallUpperCaseBM25, FAISS, and RankFusion (RF) to improve both retrieval and answer generation. Our rank fusion strategy merges semantic and lexical relevance scores to enhance accuracy and diversity. We filter top-ranked passages to remove irrelevant results before generating answers. While FUSIONRAG achieves notable improvements in regulatory document analysis, the Repass score for generation (0.52) is slightly lower due to a focus on relevance, which impacted entailment and obligation coverage.

Acknowledgments

We sincerely thank the organizers and reviewers for their valuable contributions, constructive feedback, and support.

References

Sagar Sanjiv Desai, Saurabh Whadgar, Sathees C. Raghavan, and Bibha Choudhary. 2022. Miragdb: A knowledgebase of rag regulators.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library. *arXiv preprint arXiv:2401.08281*.

Paulo Finardi, Leonardo Avila, Rodrigo Castaldoni, Pedro Gengo, Celio Larcher, Marcos Piau, Pablo Costa, and Vinicius Caridá. 2024. The chronicles of rag: The retriever, the chunk and the generator. *Preprint*, arXiv:2401.07883.

Godwin George and Rajeev Rajan. 2022. A faiss-based search for story generation. In *2022 IEEE 19th India Council International Conference (INDICON)*, pages 1–6. IEEE.

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

Yikun Han, Chunjiang Liu, and Pengfei Wang. 2023. A comprehensive survey on vector database: Storage and retrieval technique, challenge. *Preprint*, arXiv:2310.11703.

Amy E Hodler and Mark Needham. 2022. Graph data science using neo4j. In *Massive Graph Analytics*, pages 433–457. Chapman and Hall/CRC.

Oz Huly, Idan Pogrebinsky, David Carmel, Oren Kurland, and Yoelle Maarek. 2024. Old ir methods meet rag. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2559–2563.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Preprint*, arXiv:2112.09118.

Marten Jostmann and Hendrik Winkelmann. 2024. Evaluation of hypothetical document and query embeddings for information retrieval enhancements in the context of diverse user queries.

Lucia D Krisnawati, Aditya W Mahastama, Su-Cheng Haw, Kok-Why Ng, and Palanichamy Naveen. 2024. Indonesian-english textual similarity detection using universal sentence encoder (use) and facebook ai similarity search (faiss). *CommIT (Communication and Information Technology) Journal*, 18(2):183–195.

Jerry Liu. 2023. Fine-tuning a linear adapter for any embedding model.

Justin J. Miller. 2013. Graph database applications and concepts with neo4j. In *SAIS 2013 proceedings*.

OpenAI. 2022. Openai embeddings: text-embedding-ada-002.

OpenAI. 2023. Gpt-4 technical report. Technical report, OpenAI.

Mohamed Saad, Yingzhong Zhang, Jinghai Tian, and Jia Jia. 2023. A graph database for life cycle inventory using neo4j. *Journal of Cleaner Production*, 393:136344.

Spurthi Setty, Harsh Thakkar, Alyssa Lee, Eden Chung, and Natan Vidra. 2024. Improving retrieval for rag based question answering models on financial documents. *Preprint*, arXiv:2404.07221.

Tao Shen, Guodong Long, Xiubo Geng, Chongyang Tao, Yibin Lei, Tianyi Zhou, Michael Blumenstein, and Daxin Jiang. 2024. Retrieval-augmented retrieval: Large language models are strong zero-shot retriever. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 15933–15946.

Sriram Veturi, Saurabh Vaichal, Reshma Lal Jagadheesh, Nafis Irtiza Tripto, and Nian Yan. 2024. Rag based question-answering for contextual response prediction system. *Preprint*, arXiv:2409.03708.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. *Preprint*, arXiv:2210.03629.

A Limitations

Our model relies on pre-trained models like BM25 and FAISS, which may not fully capture domain-specific nuances in regulatory texts, potentially leading to less precise results. While rank fusion enhances retrieval accuracy, it introduces computational overhead, which can impact scalability in large-scale or real-time applications. FAISS embeddings may also struggle with ambiguous or out-of-distribution queries, limiting robustness. Furthermore, the approach is heavily dependent on the quality of the embeddings and retrieval models, necessitating periodic updates to keep pace with evolving regulatory language and datasets.

B Training and Efficiency

Our model avoids custom training by leveraging pre-trained models, ensuring efficiency and scalability. This eliminates resource-intensive training while maintaining strong performance, making it a lightweight and effective solution for regulatory QA tasks.

C Task and Data

The RIRAG shared task consists of two challenges aimed at advancing regulatory document question-answering: *Task 1: Information Retrieval* focuses on retrieving relevant passages from regulatory documents based on user queries, emphasizing efficient retrieval for effective downstream processing. *Task 2: Answer Generation* uses the passages from Task 1 to generate accurate, context-aware answers to queries. Together, these tasks address both the precision of retrieval and the complexity of answer generation, reflecting real-world QA system challenges.

The ObliQA dataset (Gokhan et al., 2024) includes 640K words of financial regulatory text from 40 UAE free zone documents, with complex legal obligations, numbered clauses, and cross-references. It pairs queries with relevant passages (single or multi-passage), annotated with DocumentID, PassageID, and text in JSON format. The dataset supports both single and cross-document retrieval tasks, with splits for training (22,295 queries), development (2,888 queries), and testing (2,786 queries), plus 446 unseen queries for final evaluation, enabling tasks of varying complexity.

We use the given ObliQA dataset which includes three subsets: the train set contains 22,295 questions, the test set has 2,786 questions, and the development set includes 2,888 questions. We use the train and development sets for evaluating various models and the final evaluation is performed on the unseen test set provided by the organizers.

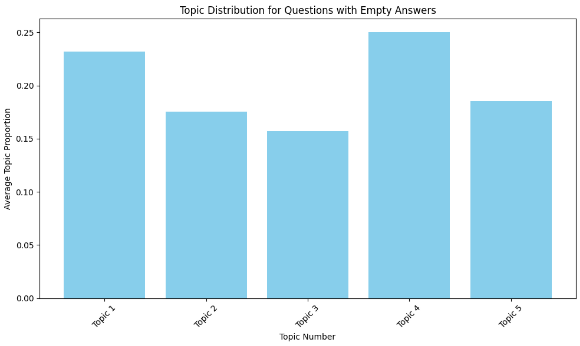

Figure 2: Topic distribution for questions with empty answers.

D Error Analysis

D.1 Topic Modeling on Questions with Empty Answers

Topic 1 consists of the following keywords: *person, authorised, specific, ADGM, assets*. Topic 2 presents: *risk, within, person, provide, compliance*. Topic 3 consists of: *compliance, person, ADGM, ensure, risk*. Topic 4 have: *could, provide, virtual, requirements, specific*. Topic 5 presents: *risk, ADGM, specific, person, compliance*.

D.2 Topic Modeling on Questions with Generated Answers

Here, Topic 1 presents: *ADGM, reporting, authorised, provide, person*. Topic 2 consists of: *financial, risk, ADGM, risks, person*. Topic 3 have: *risk, information, ADGM, management, regulator*. Topic 4 presents: *ADGM, risk, specific, investment, included*. Topic 5 consists of: *risk, authorised, constitutes, identifying, book*.

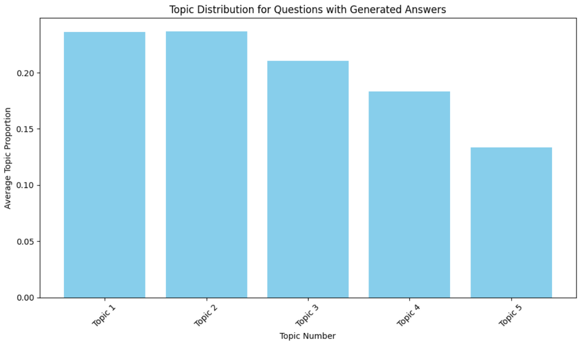

Figure 3: Topic distribution for questions with generated answers.