RAGulator: Effective RAG for Regulatory Question Answering

Islam Aushev[1] **Egor Kratkov**[1] **Evgenii Nikolaev**[1]
Andrei Glinskii[1] **Vasilii Krikunov**[1] **Alexander Panchenko**[3,2]
Vasily Konovalov[2,1] **Julia Belikova**[4,1]
[1]Moscow Institute of Physics and Technology
[2]AIRI [3]Skoltech [4]Sber AI Lab
{aushev.ia, belikova.iua, vasily.konovalov}@phystech.edu

Abstract

Regulatory Natural Language Processing (RegNLP) is a multidisciplinary domain focused on facilitating access to and comprehension of regulatory documents and requirements. This paper outlines our strategy for creating a system to address the Regulatory Information Retrieval and Answer Generation (RIRAG) challenge, which was conducted during the RegNLP 2025 Workshop. The objective of this competition is to design a system capable of efficiently extracting pertinent passages from regulatory texts (ObliQA) and subsequently generating accurate, cohesive responses to inquiries related to compliance and obligations. Our proposed method employs a lightweight BM25 pre-filtering in retrieving relevant passages. This technique efficiently shortlisting candidates for subsequent processing with Transformer-based embeddings, thereby optimizing the use of resources.

1 Introduction

The complexity, volume, and ever-changing nature of regulatory documents present unique challenges in governance, compliance, and legal frameworks across various sectors. Addressing these challenges demands specialized approaches in natural language processing (NLP) to enable effective management and utilization of regulatory content.

The Retrieval and Answer Generation (RIRAG) Shared Task as part of the RegNLP workshop focuses on building systems that can effectively navigate and extract relevant information from regulatory texts to generate precise, coherent answers for compliance and obligation-related queries. The task is divided into two main subtasks: (1) passage retrieval – given a regulatory question, participants must develop systems to identify and retrieve the most relevant passages, specifically obligations and related rules, from ADGM regulations and guidance documents; (2) answer generation – using the question and the passages retrieved in subtask 1, participants must generate a comprehensive, accurate, and coherent answer. This subtask emphasizes the ability to synthesize information from multiple sources and present it in a clear and logical manner, ensuring that the answer fully addresses the compliance and obligation requirements of the query.

This paper is structured as follows. Section 2 discusses existing work on RegNLP. Section 3 describes the ObliQA dataset, and Section 4 introduces the evaluation metrics. Section 5 describes our approach to develop RIRAG system, which first retrieves relevant passages for a given query and secondly generates an answer from these passages, and presents our evaluation of both steps. Section 6 reports the results of the applied approaches.

Our primary contributions in this work can be summarized as follows:

- We introduce a lightweight BM25 pre-filtering in retrieving relevant passages. This technique efficiently shortlisting candidates for subsequent processing with Transformer-based embeddings, thereby optimizing the use of resources.
- We also contribute a critical observation to the RegNLP community: methods that have yielded positive outcomes in broad domains may not guarantee similar success in the specialized regulatory domain. Our findings negate the assumption that the contextualization techniques, which have been effective elsewhere, can be directly applied to the regulatory domain without adaptation.

2 Related Work

The integration of Retrieval-Augmented Generation (RAG) techniques and associated technologies hold potential for enhancing RegNLP (Lewis et al., 2020). By capitalizing on advancements in

Proceedings of the 31st International Conference on Computational Linguistics, pages 114–120
January 19–24, 2025.

NLP and information retrieval systems, these methods can alleviate the difficulties posed by intricate and ever-evolving regulatory documents, thereby streamlining access to such documents and boosting compliance efficiency. RAG has significantly improved the accuracy, efficiency, and trustworthiness of LLMs by integrating external, contextually relevant and up-to-date information (Belikova et al., 2024).

Notable approaches include: Self-RAG (Asai et al., 2024) improves response quality by incorporating self-reflection mechanisms. Krayko et al. (2024) introduced an efficient QA system that combines local knowledge base search with generative context-based QA. Salnikov et al. (2023) proposed an algorithm for subgraphs extraction from a Knowledge Graph based on question entities and answer candidates. The proposed technique boosts Hits@1 scores of the pre-trained text-to-text language models by 4–6%. Shallouf et al. (2024) demonstrated how a system for argument retrieval can significantly improve the quality of a language model-based question answering system for comparative questions. All aforementioned methods highly improve the trustfulness of the QA system and minimize hallucinations (Maksimov et al., 2024).

LMs often struggle to pay enough attention to the input context and generate texts that are unfaithful or contain hallucinations. To mitigate this issue, Context-Aware Decoding (CAD) (Shi et al., 2023) was introduced, which follows a contrastive output distribution that amplifies the difference between the output probabilities when a model is used with and without context.

However, these studies do not consider regulatory documents so we are interested in testing the ability of RAG methods for solving the QA task for regulatory questions.

3 Dataset

The Obligation-Based Question Answering Dataset (ObliQA), specifically compiled for competition organizers, is based on regulatory documents provided by Abu Dhabi Global Markets (ADGM). ADGM serves as the authority overseeing financial services within the UAE's free economic zones. ObliQA has been developed as a multi-document, multi-passage Question Answering dataset, designed specifically to advance the field of Regulatory Natural Language Processing (Reg-NLP).

It comprises 27,869 questions along with their associated source passages. Each question may have from 1 to 6 relevant passages. The dataset is categorized into groups with varying distributions of relevant passages for the questions. Following this categorization, the entire dataset is split into three sections: training (comprising 22,295 questions), testing (featuring 2,786 questions), and development (consisting of 2,888 questions).

4 Evaluation

To evaluate the retrieval stage in RIRAG, we primarily use Recall@10 as the metric. This is because we depend on the retrieval module to capture as much relevant information as possible, while the task of filtering out noise is left to the answer generation module.

The answer generation subtask is evaluated by a reference-free Regulatory Passage Answer Stability Score (RePASs). RePASs designed to assess generated answers within regulatory compliance contexts. This metric evaluates answers through the lens of three pivotal criteria: (1) each sentence within an answer must find support in a corresponding sentence from the source passage(s); (2) answers are required to exclude any sentences that introduce contradictions to the information established in the source passage(s); (3) comprehensive coverage is essential; answers must encapsulate all obligations delineated in the source passages, ensuring that every critical regulatory obligation is accurately mirrored in the response.

5 Regulatory Information Retrieval and Answer Generation Task

The pipeline of the proposed approach can be found in Appendix D.

5.1 Subtask 1. Passage Retrieval

We employ two approaches to represent queries and passages: (1) sparse vector representations based on term frequencies in the query and passage, and (2) dense vector-based representations that capture semantic meaning effectively, provided by transformer-based embedders.

For the sparse vector representation, we utilized **BM25** (Robertson et al., 1994). The choice of transformer-based embedders was based on the MTEB leaderboard[1]. We experimented with two

[1] https://hf.co/spaces/mteb/leaderboard

Model	Context	F@0	F@100	F@200	F@300	F@500	F@700	F@1000
BGE-en-ICL	+	75.22	**77.82**	77.22	76.95	76.55	76.47	76.23
	-	77.39	**78.71**	78.51	78.37	78.02	77.92	77.57
NV-Embed-v2	+	74.34	**77.21**	76.91	76.36	75.87	75.72	75.72
	-	78.68	79.02	**80.45**	78.91	78.87	78.82	78.80

Table 1: Recall@10 results of the retrieval task for the transformer-based embedders. Where Context denotes enriching passages with document context, F@n represents pre-filtration with top-n passages retrieved by BM25, F@0 represents no pre-filtration. According to the results BM25 pre-filtration significantly improves the retrieval performance.

top embedders (they are comparable in the number of parameters): (1) **NV-embed-v2**[2] (Lee et al., 2024) represents the forefront in dense embedders, introducing a series of models aimed at enhancing performance; (2) **BGE-en-ICL**[3] (Xiao et al., 2023) – BAAI general embedder that supports in-context learning ability. By providing few-shot samples, it can significantly improve the model's ability to address new tasks.

Fusion To this end, we apply rank fusion to linearly fuse the passage ranking by the neural or BM25 retrievers. Reciprocal Rank Fusion (RRF) is an algorithm that evaluates the search scores from multiple, previously ranked results to produce a unified result set (Cormack et al., 2009).

Contextualization In basic RAG, embedded passages hold valuable info but lack context. To address this, we've employed Contextual Retrieval (Anthropic, 2024). By feeding both isolated text passages and their broader document context into Llama-3.1-70B (AI@Meta, 2024), we generate succinct, explanatory contexts. For our obligatory dataset, this involves presenting the passage alongside its entire originating document to an LLM, generating context, and merging this with the raw text before creating embeddings. This approach enriches each passage with pertinent background, enhancing understanding.

Reranking The re-ranker plays a key role in the RAG pipeline, improving the quality of the top-k documents. Its goal is to redistribute priorities among the found documents, selecting those that are most relevant to the given query. The reranking techniques are described in Appendix C.

5.2 Subtask 2. Answer Generation

In the process of generating answers, we employed the Llama-3.1-8B-Instruct model. Across all these experiments, a consistent *Answer Generation Prompt* was utilized to maintain uniformity (Appendix A).

There is an assumption that within a precise domain, the LLMs should heavily depend on the contextual (non-parametric) knowledge available rather than relying solely on their own (parametric) knowledge. This is because it's highly unlikely that the specific knowledge of a particular domain, like regulation, would be incorporated within the model's parameterized understanding.

Following this hypothesis, we applied Classifier-Free Guidance (CFG) (Sanchez et al., 2024). We experiment with different `guidance_scale` that decides how to divide LMs attention between context and output. In addition, we employed Context-Aware Decoding (CAD) (Shi et al., 2023). Using the same approach as with CFG but with a different formula.

The answer generation process begins once 10 relevant passages have been retrieved for each query from the passage retrieval task.

6 Results

For evaluation we used the labeled test split of ObliQA and not the hidden evaluation split that was introduced in Gokhan et al. (2024). The labeled test split contains 2,786 question-passage pairs, while the hidden evaluation has only 446 pairs.

6.1 Subtask 1. Passage Retrieval

The results of the retrieval task on Recall@10 are shown in Table 1. Our results align with previous findings. Despite its simplicity, BM25 is still a robust baseline for retrieval. The current leader on the MTEB (Muennighoff et al., 2023) leaderboard, **NV-Embed-v2**, confirms its superiority in the regulatory domain – significantly outperforming all other embedders. **BGE-en-ICL** is just slightly behind **NV-Embed-v2**.

[2] https://hf.co/nvidia/NV-Embed-v2
[3] https://hf.co/BAAI/bge-en-icl

Pipeline	F@0	F@100	F@200	F@300	F@500	F@700	F@1000
BGE-en-ICL	80.04	79.69	80.11	**80.22**	80.16	80.12	80.14
NV-Embed-v2	80.48	79.92	80.45	**80.59**	80.55	80.53	80.43
BGE-en-ICL + NV-Embed-v2	80.76	80.72	**81.10**	81.03	80.88	80.86	80.68

Table 2: Recall@10 of the combinations achieved through Reciprocal Rank Fusion (RRF) of BM25 with all variants of two dense embedders. Here, F@n denotes pre-filtration, where the top-n passages retrieved by BM25 are selected for further processing. Conversely, F@0 signifies the absence of any pre-filtration, meaning all passages are considered equally before the fusion process.

Moreover, we tested the listed embedders in a pre-filtration mode where, for semantic search, we used only the top-200 passages retrieved by BM25. This approach slightly improves all embedding-based techniques. The key factor is that BM25 filters out irrelevant passages that could erroneously be retrieved by embedders.

Fusion The outcomes of Reciprocal Rank Fusion (RRF) combining BM25 with all variants of dense embedders are delineated in Table 2. Based on these findings, employing reciprocal rank fusion remarkably enhanced the performance, signifying its effectiveness in integrating diverse retrieval systems to achieve superior results.

Contextualization Previously, contextualization was found to be an incredibly effective technique (Anthropic, 2024). However, in our experiments, it has proven counterproductive. Upon further analysis, we discovered that contextualization introduces a surplus of irrelevant information from the source documents into the passages. These unnecessary details confuse the models and significantly raise the likelihood of making incorrect retrievals. In light of these findings, we made the decision to exclude contextualization from our future experiments.

Reranking The results of the reranking are shown in Table 5. According to the results, reranking techniques do not provide significant improvements. The reranking methods with corresponding results are described in Appendix C.

6.2 Subtask 2. Answer Generation

The answer generation results are presented in Table 3. The optimal hyperparameters of the employed approaches are listed in the Table 4.

Both CFG and CAD demonstrate superior performance in RePASs, when they concentrate more effectively on the input context. However, Llama-3.1-8B, using a beam search size of 4, notably outperformed these specific adaptations of CFG and CAD.

Model	Setting	RePASs
Llama-3.1-8B	–	48.64
	BS	<u>70.09</u>
	CFG	59.22
	CAD	64.32
Target Passage	–	**95.02**

Table 3: Results of the generation task on target passages from the test split, where BS denotes beam search, CFG – Classifier-Free Guidance, CAD – Context-Aware Decoding.

It achieved a striking 70% RePASs, showcasing its proficiency in maintaining relevancy. Surprisingly, the highest RePASs of 95.0% was accomplished through a rather straightforward method: merely outputting the top-ranked passage retrieved from the preceding retrieval phase. This finding underscores the potential efficiency of simple strategies in certain contexts. At this stage, we assessed the generation techniques by employing different metrics, such as In-Accuracy or AlignScore (Zha et al., 2023). However, the reference answer generations were not available for comparison.

7 Conclusion

In this paper, we have described the system we submitted for the RIRAG challenge at the Reg-NLP workshop, specifically concentrating on developing a QA system tailored to the regulatory domain. We proposed a simple yet effective QA pipeline. Our study highlighted that lightweight BM25 pre-filtering can efficiently retrieve candidate passages for more resourceful fusion using Transformer-based embeddings. We demonstrated that techniques proven successful in general domains may not directly translate to the regulatory domain, as seen with the unsuccessful application of contextualization.

References

AI@Meta. 2024. Llama 3 model card.

Anthropic. 2024. Introducing contextual retrieval. Accessed: 2024-11-29.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Julia Belikova, Evegeniy Beliakin, and Vasily Konovalov. 2024. JellyBell at TextGraphs-17 shared task: Fusing large language models with external knowledge for enhanced question answering. In *Proceedings of TextGraphs-17: Graph-based Methods for Natural Language Processing*, pages 154–160, Bangkok, Thailand. Association for Computational Linguistics.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*.

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.

Nikita Krayko, Ivan Sidorov, Fedor Laputin, Daria Galimzianova, and Vasily Konovalov. 2024. Efficient answer retrieval system (EARS): Combining local DB search and web search for generative QA. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1584–1594, Miami, Florida, US. Association for Computational Linguistics.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. *Preprint*, arXiv:2405.17428.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Ivan Maksimov, Vasily Konovalov, and Andrei Glinskii. 2024. DeepPavlov at SemEval-2024 task 6: Detection of hallucinations and overgeneration mistakes with an ensemble of transformer-based models. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 274–278, Mexico City, Mexico. Association for Computational Linguistics.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2023. MTEB: massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 2006–2029. Association for Computational Linguistics.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3. In *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994*, volume 500-225 of *NIST Special Publication*, pages 109–126. National Institute of Standards and Technology (NIST).

Mikhail Salnikov, Hai Le, Prateek Rajput, Irina Nikishina, Pavel Braslavski, Valentin Malykh, and Alexander Panchenko. 2023. Large language models meet knowledge graphs to answer factoid questions. In *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation*, pages 635–644, Hong Kong, China. Association for Computational Linguistics.

Guillaume Sanchez, Alexander Spangher, Honglu Fan, Elad Levi, and Stella Biderman. 2024. Stay on topic with classifier-free guidance. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Maksim Savkin, Anastasia Voznyuk, Fedor Ignatov, Anna Korzanova, Dmitry Karpov, Alexander Popov, and Vasily Konovalov. 2024. DeepPavlov 1.0: Your gateway to advanced NLP models backed by transformers and transfer learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 465–474, Miami, Florida, USA. Association for Computational Linguistics.

Ahmad Shallouf, Hanna Herasimchyk, Mikhail Salnikov, Rudy Alexandro Garrido Veliz, Natia Mestvirishvili, Alexander Panchenko, Chris Biemann, and Irina Nikishina. 2024. CAM 2.0: End-to-end open domain comparative question answering system. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2657–2672, Torino, Italia. ELRA and ICCL.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Scott Wen tau Yih. 2023. Trusting your evidence: Hallucinate less with context-aware decoding. *Preprint*, arXiv:2305.14739.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources

to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. AlignScore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348, Toronto, Canada. Association for Computational Linguistics.

A Prompts and Instructions

Cotextualization Prompt

<document>
{{WHOLE_DOCUMENT}}
</document>
Here is the chunk we want to situate within the whole document
<chunk>
{{CHUNK_CONTENT}}
</chunk>
Please give a short succinct context to situate this chunk within the overall document for the purposes of improving search retrieval of the chunk. Answer only with the succinct context and nothing else.

Answer Generation Prompt

Documents: {{PASSAGE}}
Answer the question below using the given regulatory documents. Every answer sentence must be supported by a sentence in the source documents. The answer must not contain any sentences that contradict the information in the source documents. The answer must cover all the obligations present in the source documents, meaning that all critical regulatory obligations should be reflected in the answer. Don't say anything that is not supported by source documents. If the part of the given document doesn't answer the question – ignore it.
Question: {{QUESTION}}
Answer:

B Answer Generation Settings

Model	Parameter	Value
Llama-3.1-8B	`top_p`	0.95
	`temperature`	1
	`max_new_tokens`	400
	`beam_searches`	4
CFG	`guidance_scale`	1.2
CAD	`alpha`	0.2

Table 4: Answer generation models settings.

C Reranking

The reranker approaches we employed are based on the *cross-encoder* architecture. This architecture is characterized by its ability to process the query and the document concurrently. By passing these elements through the same encoder as a unified sequence, delineated by a specific separator token (`[SEP]`), it enables the model to consider the reciprocal impact of words from both texts. This design facilitates the creation of a representation that is optimally tailored for accurate classification. The training of our cross-encoder was executed using the DeepPavlov framework (Savkin et al., 2024), ensuring a robust and effective learning process.

In alignment with the ObliQA building pipeline, where the authors selectively included only those questions that exhibit a strong semantic correlation with passages. To substantiate this relationship, they employed an NLI (Natural Language Inference) approach, setting the passage as the premise and the question as the hypothesis. Inspired by their methodology, we chose to explore two NLI-based approaches for our reranking process: a naive NLI technique and the Question-Answering Natural Language Inference (QNLI).

In addition, we measure the semantic relation between queries and passages by applying `BAAI/bge-reranker-large` and `bge-reranker-large-finetuned`.

The results of the reranking are shown in Table 5. According to the results, reranking techniques do not provide any significant improvements.

Model	Top-1	Top-3	Top-5	Top-10	Recall@10
RRF(BM25, BGE-en-ICL, NV-Embed-v2)	58.33	72.51	76.20	81.01	81.10
NLI					
`nli-deberta-v3-base`	32.09	50.50	59.69	72.07	72.26
`nli-deberta-v3-large`	21.00	31.84	40.95	58.83	59.06
QNLI					
`qnli-electra-base`	25.63	41.53	49.78	63.68	63.93
`qnli-distilroberta-base`	21.86	39.45	49.57	65.51	65.77
Binary Classification					
`bge-reranker-large`	54.20	68.16	73.13	80.19	80.19
`bge-reranker-large-finetuned`	58.76	71.28	76.02	81.12	81.18

Table 5: Re-ranking metrics for different models. Top-n means the proportion of occurrence of the relevant passage in the first n passages with the highest score.

D Retrieval and Generation Pipeline

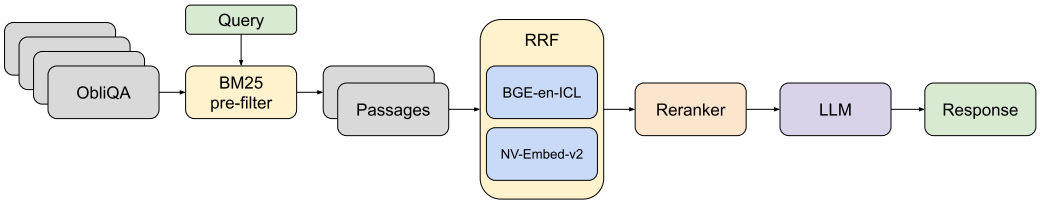

Figure 1: Retrieval and Generation Pipeline