A REGNLP Framework: Developing Retrieval-Augmented Generation for Regulatory Document Analysis

Ozan Bayer[1] **(obayer@etu.edu.tr), Elif Nehir Ulu**[2] **(enulu@havelsan.com.tr),**
Yasemin Sarkın[3] (sarkin.yasemin@student.atilim.edu.tr), **Ekrem Sütçü**[1] (esutcu@etu.edu.tr),
Defne Buse Çelik[4] (dbuse.celik@gazi.edu.tr), **Alper Karamanlıoğlu**[2] (alperk@havelsan.com.tr),
İsmail Karakaya[2] (ikarakaya@havelsan.com.tr), **Berkan Demirel**[2] (bdemirel@havelsan.com.tr),
[1]TOBB ETU, [2]HAVELSAN Inc., [3]Atılım University, [4]Gazi University
Correspondence: enulu@havelsan.com.tr

Abstract

This study presents the development of a Retrieval-Augmented Generation (RAG) framework tailored for analyzing regulatory documents from the Abu Dhabi Global Markets (ADGM)[1]. The methodology encompasses comprehensive data preprocessing, including extraction, cleaning, and compression of documents, as well as the organization of the ObliQA dataset[2]. The embedding model[3] is utilized for generating embeddings during the retrieval phase, facilitated by the `txtai` library for managing embeddings and streamlining testing. The training process incorporated innovative strategies such as duplicate recognition, dropout implementation, pooling adjustments, and label modifications to enhance retrieval performance. Hyperparameter tuning further refined the retrieval component, with improvements validated using the recall@10 metric, which measures the proportion of relevant passages among the top-10 results. The refined retrieval component effectively identifies pertinent passages within regulatory documents, expediting information access and supporting compliance efforts.

1 Introduction

Regulatory documents are comprehensive texts that outline mandatory rules and guidelines for organizational compliance. Their complexity presents significant challenges in manual analysis, often leading to inefficiencies and errors (Butler and OBrien, 2019; Padmanaban, 2024). Advances in Natural Language Processing (NLP) offer promising solutions to these challenges (Zhang and El-Gohary, 2016; Gray et al., 2023; Cejas et al., 2023). This study focuses on the development of the retrieval phase of a Retrieval-Augmented Generation (RAG) framework, aiming to accurately identify related information within Abu Dhabi Global Markets (ADGM)'s regulatory documents. By enhancing retrieval accuracy, the framework seeks to facilitate rapid access to relevant information, thereby supporting effective compliance and decision-making processes.

The methodology involves comprehensive data preprocessing, including extraction, cleaning, and compression of documents, as well as the organization of the ObliQA dataset. The embedding model is selected for embedding generation due to its efficiency in producing high-quality text representations. To streamline testing procedures, the `txtai` library is utilized, serving as an all-in-one embedding database that supports semantic search and language model workflows. The training process incorporated innovative strategies such as duplicate recognition, dropout implementation, pooling adjustments, and label modifications to enhance model performance. Hyperparameter tuning further optimized the retrieval component, and retrieved passages are validated by the recall@10 metric.

The refined retrieval framework effectively identifies relevant passages within regulatory documents, accelerating information access and supporting compliance efforts. This study underscores the transformative potential of integrating NLP technologies into regulatory processes, laying a solid foundation for future research aimed at developing comprehensive RAG systems.

2 Related Work

The application of RAG methods to regulatory workflows remains an underexplored area in the literature. Oyewole (2024) highlights the potential of RAG to improve efficiency in distinct domains by combining information retrieval and generation. However, the study notes that the implications of RAG for regulatory documents require further in-

[1]https://www.adgm.com/
[2]https://github.com/RegNLP/ObliQADataset
[3]intfloat/multilingual-e5-large

Proceedings of the 31st International Conference on Computational Linguistics, pages 97–101
January 19–24, 2025. ©COLING 2025

vestigation.

The integration of NLP into regulatory processes has been explored across various sectors, including the construction industry, financial, and healthcare sectors.

In the construction industry, Zhang and El-Gohary (2016) utilize semantic-based information extraction to automate compliance checks within construction regulations, reducing manual effort and expediting processes. In the financial sector, Oyewole (2024) develops NLP tools to analyze financial regulatory documents, enhancing both accuracy and operational efficiency. In the healthcare sector, Wu et al. (2021) employ BERT-based models to classify potential risks in drug labeling texts, providing rapid analyses for regulatory agencies. Subsequently, Wu (2023) introduces RxBERT, improving information extraction from drug labeling documents.

3 Dataset

The dataset comprises 40 regulatory documents provided by ADGM, each ranging from approximately 30 to 100 pages. These documents are segmented into passages, with each passage stored as a JSON file containing "ID," "DocumentID," and "PassageID." The passages average 60 words, with lengths varying from 1 to 24,312 words. The test dataset includes 2,786 questions, each accompanied by "QuestionID," "Question," and the corresponding passages expected to be retrieved.

4 Methodology

The methodology encompasses data preprocessing, model selection and training procedures.

4.1 Data Preprocessing

The dataset comprises regulatory documents from ADGM, provided in JSON format. Each document includes fields such as "ID," "DocumentID," "PassageID," and the corresponding text passage. The preprocessing steps involved:

1. **Data Extraction:** Parsing JSON files to extract relevant fields and converting them into a more readable format for analysis.

2. **Data Cleaning:** Identifying and removing entries with empty strings or missing values to ensure data quality.

3. **Data Compression:** Storing processed documents in compressed CSV files to optimize storage and processing efficiency.

4. **ObliQA Dataset Handling:** Extracted questions and their associated relevant passages from the ObliQA training and test datasets and organizing them into lists for subsequent processing.

4.2 Model Selection and Embedding

For extracting embedding vectors from the textual data, the *intfloat/multilingual-e5-large* model (Wang et al., 2024) is selected due to its efficiency in generating multilingual embeddings. The model comprises 24 layers with an embedding size of 1,024. To facilitate embedding and streamline testing procedures, the `txtai` library is utilized. This library serves as an all-in-one embeddings database, supporting semantic search and language model workflows (NeuML, 2023). Using this library, vectorizing, indexing, and searching capabilities can be achieved much more easily.

4.3 Training Procedure

The training process aims to fine-tune the model for effective retrieval of relevant passages in response to specific queries. The steps involved:

1. **Batch Preparation:** Organizing the dataset into batches, each containing pairs of questions and their corresponding passages.

2. **Label Matrix Construction:** Creating a label matrix analogous to an identity matrix, indicating positive (1) and negative (0) embeddings. As shown in Equation (1), the label matrix $\mathbf{L}$ is constructed as follows:

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

3. **Embedding Generation:** Utilizing the model to generate embeddings for each question and passage pair.

4. **Similarity Calculation:** Computing cosine similarity between embeddings to populate a similarity matrix, reflecting the degree of similarity between questions and passages.

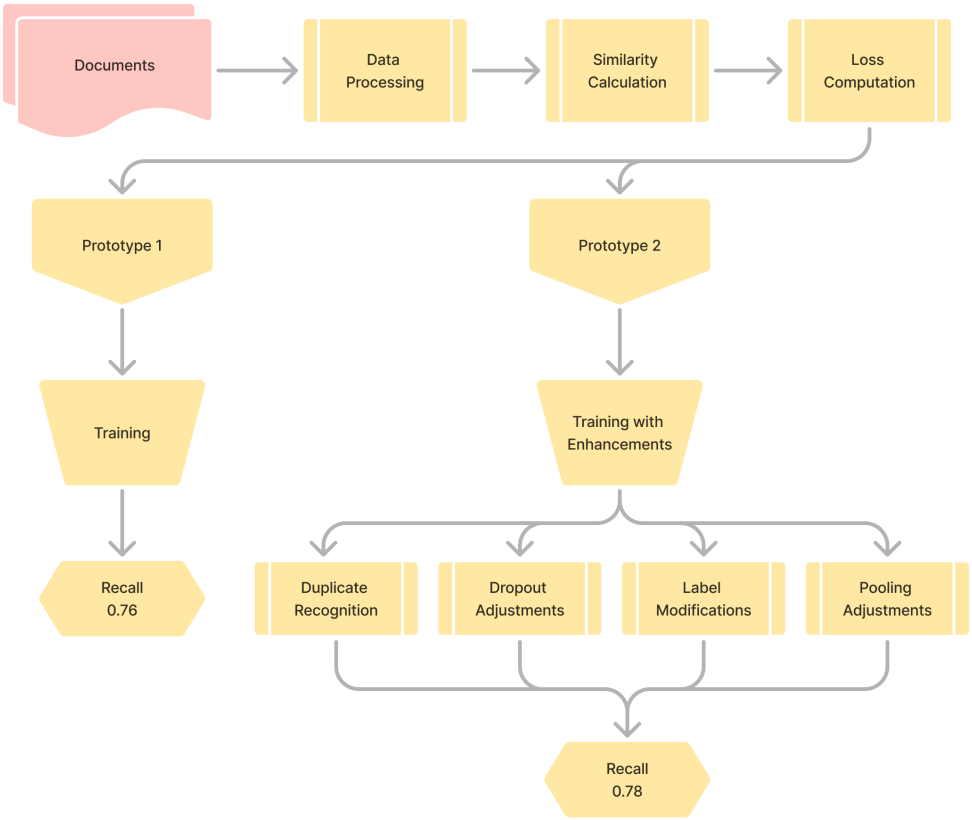

Figure 1: Proposed framework.

5. **Loss Computation:** Applying Mean Squared Error (MSE) loss between the label and similarity matrices to quantify the model's performance.

6. **Parameter Optimization:** Adjusting model parameters based on the loss function to enhance retrieval accuracy.

4.4 Prototype Development and Challenges

As shown in Figure 1, two prototypes are developed during the training phase:

- **Prototype 1:** Trained over three epochs with a learning rate of 10^{-5}. This prototype exhibited issues such as sudden increases in loss and a tendency to predict similar probabilities for different passages.

- **Prototype 2:** Implemented several enhancements, including:

  – **Duplicate Recognition:** Modifying the training model to compare question embeddings with themselves, allowing the identification of duplicate questions as positive embeddings.

  – **Dropout Addition:** Introducing a dropout rate to mitigate overfitting.

  – **Pooling Adjustment:** Applying average pooling to remove padded values and compute the mean of token embeddings, ensuring comprehensive representation.

  – **Label Adjustment:** To enhance flexibility in assessing similarity, the labels for negative embeddings are adjusted from 0 to 0.5. This adjustment allows the model to better capture partial relationships between embeddings.

$$\mathbf{L} = \begin{bmatrix} 1 & 0.5 & 0.5 & 0.5 \\ 0.5 & 1 & 0.5 & 0.5 \\ 0.5 & 0.5 & 1 & 0.5 \\ 0.5 & 0.5 & 0.5 & 1 \end{bmatrix} \quad (2)$$

As shown in Equation (2), the matrix $\mathbf{L}$ illustrates a 4x4 example. Diagonal elements (1) represent positive embeddings, indicating matching question-passage pairs, while off-diagonal elements (0.5) represent negative embeddings, reflecting partial similarity. This modification in Prototype 2 helps the model better distinguish nuanced relationships, enhancing retrieval performance.

Hyperparameter tuning is conducted to optimize the model performance. A batch size of 8 is chosen to balance memory usage and training stability. The learning rate is set to 2×10^{-7}, which facilitated gradual and stable updates to the model parameters. To accommodate the tokenization of regulatory documents, a token length of 256 is used, ensuring adequate representation of text passages while maintaining computational efficiency. Training is conducted over 3 epochs, balancing sufficient learning iterations with computational constraints.

5 Results and Discussion

Recall@10 measures the proportion of relevant passages among the top 10 returned results. The models are evaluated using the recall@10 metric. Prototype 1 achieved a recall@10 of 0.76, while Prototype 2 improved to 0.78, indicating enhanced retrieval effectiveness.

Prototype	Recall@10
Prototype 1	0.76
Prototype 2	0.78

Table 1: Recall@10 Scores for Prototypes

These results mean that duplicate detection, the addition of dropout, pooling adjustments, and adjustments in the labels are responsible for Prototype 2's higher performance. The higher recall@10 score shows that the model is better at correctly selecting relevant passages within the regulatory documents.

However, the incremental improvement from one prototype to another indicate that further improvements are needed to achieve even more significant retrieval performances. Future work should proceed in the direction of exploiting further training techniques, refining hyperparameters, and using more complex models to further improve the model's performance in processing complex regulatory texts.

6 Conclusion

The integration of NLP into regulatory processes has huge potential to facilitate compliance efficiency in many industries. This paper contributes to this dynamic area by developing a RAG model focused on the analytical aspects of regulatory documents obtained from the ADGM. The focus of the paper on the retrieval component of the RAG model enables the study to address certain challenges related to the extraction of relevant information in long and complex regulatory texts.

The methodology includes detailed data preprocessing that enables document extraction and cleaning to ensure the quality and relevance of the dataset. The choice of the model to generate embeddings, combined with the work using the `txtai` library, allowed fast embedding and smooth testing. Training included state-of-the-art methods, such as duplicate detection, dropout, and tuning pooling for better performance. This model is further optimized by applying techniques for hyperparameter tuning; the retrieval accuracy is improved, as estimated from the recall@10 metric.

These results confirm that the refined retrieval model efficiently retrieves relevant regulatory passages to speed up access and compliance to information. This is particularly important, given the complexity and volume of regulatory texts, usually beyond manual human analysis.

In conclusion, this work points out the transformational role that NLP technologies, in particular RAG frameworks, could play if embedded in regulatory processes. Realized progress during the retrieval phase provides a firm base for subsequent research to build up general RAG systems. Further work will have to be addressed for the development of more robust generation techniques and fine-tuned embedding models, which can allow improving the overall compliance workflows. Such systems have the potential to revolutionize regulatory compliance by providing accurate and contextually relevant information, leading to an agile and responsive regulatory environment.

References

Tom Butler and Leona OBrien. 2019. Artificial intelligence for regulatory compliance: Are we there yet? *Journal of Financial Compliance*, 3(1):44–59.

Orlando Amaral Cejas, Muhammad Ilyas Azeem, Sallam Abualhaija, and Lionel C. Briand. 2023. Nlp-based automated compliance checking of data processing agreements against gdpr. *IEEE Transactions on Software Engineering*, 49(9):4282–4303.

Magnus Gray, Joshua Xu, Weida Tong, and Leihong Wu. 2023. Classifying free texts into predefined sections using ai in regulatory documents: A case study with drug labeling documents. *Chemical Research in Toxicology*, 36(8):1290–1299.

NeuML. 2023. txtai: All-in-one embeddings database for semantic search, llm orchestration and language model workflows.

Adebayo Oyewole. 2024. Automating financial reporting with natural language processing: A review and case analysis. *World Journal of Advanced Research and Reviews*, 21(3):575–589.

Harish Padmanaban. 2024. Navigating the complexity of regulations: Harnessing ai/ml for precise reporting. *Journal of Artificial Intelligence General science (JAIGS) ISSN:3006-4023*, 3(1):49–61.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *Preprint*, arXiv:2402.05672.

Ling Wu. 2023. Rxbert: Enhancing drug labeling text mining and analysis with ai language modeling. *Experimental Biology and Medicine*, 248(21):1937–1943.

Yifan Wu, Zhi Liu, Ling Wu, and Min Chen. 2021. Bert-based natural language processing of drug labeling documents: A case study for classifying drug-induced liver injury risk. *Frontiers in Artificial Intelligence*, 4:729834.

Jie Zhang and Nora M. El-Gohary. 2016. Semantic nlp-based information extraction from construction regulatory documents for automated compliance checking. *Journal of Computing in Civil Engineering*, 30(2):04015014.