COLING 2025

RegNLP-2025
The First Workshop on Regulatory NLP

Workshop Proceedings

Editors
Tuba Gokhan, Kexin Wang, Iryna Gurevych, Ted Briscoe

January 20, 2025
Abu Dhabi, UAE

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 979-8-89176-217-6

Preface

Regulatory compliance is a fundamental aspect of governance across industries, yet the complexity and volume of regulatory texts often make adherence a challenging task. The field of Natural Language Processing (NLP) has started to transform this landscape by enabling automated solutions for information retrieval, question answering, and document cross-referencing. Despite this progress, significant challenges remain in adapting general NLP methods to the nuanced and specialized needs of regulatory domains.

The RegNLP-2025 workshop was organized to address these challenges, establish Regulatory NLP as a specialized area, and foster collaboration among researchers, practitioners, and policymakers. This inaugural event provided a platform for exploring advanced methods, tools, and datasets tailored to the regulatory context, aiming to improve the efficiency and accuracy of compliance tasks.

This workshop aimed to bring together NLP researchers, domain experts, and industry stakeholders to discuss developments and challenges in Regulatory NLP. It aimed to introduce new methods for regulatory information retrieval and answer generation while encouraging the exchange of knowledge on the development and evaluation of domain-specific tools and models. The workshop also included the Regulatory Information Retrieval and Answer Generation (RIRAG) shared task, which invited participants to explore innovative approaches for retrieving relevant regulatory passages and generating coherent, accurate answers to compliance-related queries.

We are grateful to the program committee for their detailed reviews, our keynote speakers for their insightful contributions, and the authors and attendees for their participation. Special thanks to the shared task participants for advancing research in regulatory information retrieval and answer generation.

We hope this workshop fosters new collaborations and ideas in the Regulatory NLP domain, paving the way for further advancements in the future.

Tuba Gokhan, Kexin Wang, Iryna Gurevych, Ted Briscoe

January 2025

Organizing Committee

- **Tuba Gokhan**, Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI)
- **Kexin Wang**, UKP Lab, Technical University of Darmstadt
- **Iryna Gurevych**, UKP Lab, Technical University of Darmstadt & MBZUAI
- **Ted Briscoe**, Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI)

Program Committee

- **Sallam Abualhaija**, University of Luxembourg
- **Chetan Arora**, Monash University
- **Thales Bertaglia**, Maastricht University
- **Travis D. Breaux**, Carnegie Mellon University
- **Silvana Castano**, University of Milan
- **Luigi Di Caro**, University of Turin
- **Ashish Chouhan**, Heidelberg University
- **Chandra Kiran Reddy Evuru**, University of Maryland
- **Alfio Ferrara**, University of Milan
- **Tunga Gungor**, Bogazici University
- **Lena Held**, UKP Lab, Technical University of Darmstadt
- **Timour Igamberdiev**, UKP Lab, Technical University of Darmstadt
- **Daniel Martin Katz**, Chicago-Kent College of Law - Illinois Institute of Technology
- **Manolis Koubarakis**, National and Kapodistrian University of Athens
- **Prodromos Malakasiotis**, Athens University of Economics and Business
- **Luisa Mich**, University of Trento
- **Paulo Quaresma**, University of Evora
- **Carlo Sansone**, University of Naples Federico II
- **Dimitrios Tsarapatsanis**, University of York
- **Peter Vickers**, University of Sheffield
- **Nicola Zeni**, University of Trento

Table of Contents

Conference Program

9:00–10:15 **Session 1**

9:00–9:05 ***Opening Remarks***

9:05–9:35 ***Invited Speaker***

9:35–9:55 *Shared Task RIRAG-2025: Regulatory Information Retrieval and Answer Generation*
Tuba Gokhan, Kexin Wang, Iryna Gurevych and Ted Briscoe

9:55–10:15 *Challenges in Technical Regulatory Text Variation Detection*
Shriya Vaagdevi Chikati, Samuel Larkin, David Minicola and Chi-kiu Lo

10:15–11:00 **Coffee Break**

11:00–12:20 **Session 2**

11:00–11:30 ***Invited Speaker***

11:30–11:55 *Bilingual BSARD: Extending Statutory Article Retrieval to Dutch*
Ehsan Lotfi, Nikolay Banar, Nerses Yuzbashyan and Walter Daelemans

11:55–12:20 *Unifying Large Language Models and Knowledge Graphs for efficient Regulatory Information Retrieval and Answer Generation*
Kishore Vanapalli, Aravind Kilaru, Omair Shafiq and Shahzad Khan

12:20–14:00 **Lunch Break**

14:00–15:20 **Session 3**

14:00–14:20 *A Hybrid Approach to Information Retrieval and Answer Generation for Regulatory Texts*
Jhon Stewar Rayo Mosquera, Carlos Raul De La Rosa Peredo and Mario Garrido Cordoba

14:20–14:40 *1-800-SHARED-TASKS at RegNLP: Lexical Reranking of Semantic Retrieval (LeSeR) for Regulatory Question Answering*
Jebish Purbey, Drishti Sharma, Siddhant Gupta, Khawaja Murad, Siddartha Pullakhandam and Ram Mohan Rao Kadiyala

14:40–15:00 *MST-R: Multi-Stage Tuning for Retrieval Systems and Metric Evaluation*
Yash Malviya, Karan Dhingra and Maneesh Singh

15:00–15:20 *AUEB-Archimedes at RIRAG-2025: Is Obligation concatenation really all you need?*
Ioannis Chasandras, Odysseas S. Chlapanis and Ion Androutsopoulos

15:20–16:00 **Coffee Break**

16:00–17:30 **Poster Session**

16:00–17:30 *Structured Tender Entities Extraction from Complex Tables with Few-short Learning*
Asim Abbas, Mark Lee, Niloofer Shanavas, Venelin Kovatchev and Mubashir Ali

16:00–17:30 *A Two-Stage LLM System for Enhanced Regulatory Information Retrieval and Answer Generation*
Fengzhao Sun, Jun Yu, Jiaming Hou, yutong lin and Tianyu Liu

16:00–17:30 *NUST Nova at RIRAG 2025: A Hybrid Framework for Regulatory Information Retrieval and Question Answering*
Mariam Babar Khan, Huma Ameer, Seemab Latif and Mehwish Fatima

16:00–17:30 *NUST Alpha at RIRAG 2025: Fusion RAG for Bridging Lexical and Semantic Retrieval and Question Answering*
Muhammad Rouhan Faisal, Muhammad Abdullah, Faizyaab Ali Shah, Shalina Riaz, Huma Ameer, Seemab Latif and Mehwish Fatima

16:00–17:30 *NUST Omega at RIRAG 2025: Investigating Context-aware Retrieval and Answer Generations-Lessons and Challenges*
Huma Ameer, Muhammad Hannan Akram, Seemab Latif and Mehwish Fatima

16:00–17:30 *Enhancing Regulatory Compliance Through Automated Retrieval, Reranking, and Answer Generation*
Kübranur Umar, Hakan Doğan, Onur Özcan, İsmail Karakaya, Alper Karamanlıoğlu and Berkan Demirel

16:00–17:30 *A REGNLP Framework: Developing Retrieval-Augmented Generation for Regulatory Document Analysis*
Ozan Bayer, Elif Nehir ULU, Yasemin Sarkın, Ekrem Sütçü, Defne Buse Çelik, Alper Karamanlıoğlu, İsmail Karakaya and Berkan Demirel

16:00–17:30 *Regulatory Question-Answering using Generative AI*
Devin Quinn, Sumit P. Pai, Iman Yousfi, Nirmala Pudota and Sanmitra Bhattacharya

16:00–17:30 *RIRAG: A Bi-Directional Retrieval-Enhanced Framework for Financial Legal QA in ObliQA Shared Task*
Xinyan Zhang, Xiaobing Feng, Xiujuan Xu, zhiliang zheng and Kai Wu

16:00–17:30 *RAGulator: Effective RAG for Regulatory Question Answering*
Islam Aushev, Egor Kratkov, Evgenii Nikoalev, Andrei Vladimirovich Glinskii, Vasilii Krikunov, Alexander Panchenko, Vasily Konovalov and Julia Belikova

Shared Task RIRAG-2025: Regulatory Information Retrieval and Answer Generation

Tuba Gokhan[1], **Kexin Wang**[2], **Iryna Gurevych**[1,2], **Ted Briscoe**[1]
[1]Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI)
[2]Ubiquitous Knowledge Processing Lab (UKP Lab)
Department of Computer Science and Hessian Center for AI (hessian.AI)
Technical University of Darmstadt

Abstract

This paper provides an overview of the Shared Task RIRAG-2025, which focused on advancing the field of Regulatory Information Retrieval and Answer Generation (RIRAG). The task was designed to evaluate methods for answering regulatory questions using the ObliQA dataset. This paper summarizes the shared task, participants' methods, and the results achieved by various teams.

1 Introduction

Regulatory compliance is a critical but highly complex domain, requiring organizations to interpret and adhere to a wide range of rules, standards, and obligations. These tasks are traditionally labor-intensive and involve meticulous analysis of all regulatory documents to ensure compliance. The growing volume and complexity of regulations has made manual processes increasingly unsustainable. Addressing these challenges necessitates innovative solutions to automate regulatory compliance tasks.

The **Regulatory Information Retrieval and Answer Generation (RIRAG)** focuses on automating two core processes: retrieving relevant regulatory information and generating concise, accurate answers to compliance-related questions. By combining information retrieval and answer generation, RIRAG provides a framework to streamline compliance workflows and enhance organizational efficiency.

To foster collaboration and innovation in this emerging field, we organized the **RIRAG-2025 shared task**. This shared task aims to benchmark and advance methodologies for regulatory information retrieval and answer generation, bringing together academic and industrial researchers to address real-world compliance challenges.

2 RIRAG-2025

2.1 Task Description

The Regulatory Information Retrieval task seeks to automate the extraction and synthesis of information from complex regulatory documents. This involves addressing multi-passage and multi-document challenges inherent to regulatory compliance. The task is divided into two subtasks:

Subtask 1: Information Retrieval: The objective is to retrieve the most relevant passages from a regulatory corpus for a given compliance-related question. These passages form the foundation for generating accurate answers.

Subtask 2: Answer Generation: This subtask focuses on generating a comprehensive based on the passages retrieved in Subtask 1. The generated answers must integrate all relevant obligations while avoiding contradictions or omissions.

2.2 Dataset: ObliQA

The shared task leverages the ObliQA dataset [1], a regulatory compliance-focused dataset derived from Abu Dhabi Global Market (ADGM) regulations. ObliQA comprises 27,869 questions, each annotated with corresponding passages, making it a robust resource for developing and benchmarking RIRAG systems. The dataset poses unique challenges, including:

Single-Passage Questions: Questions that require retrieving and analyzing a single passage.

Multi-Passage Questions: Questions necessitating the integration of multiple passages for a complete answer.

2.3 Baseline System

The baseline system (Gokhan et al., 2024) serves as a foundational framework for the participants, providing a clear reference for addressing the RIRAG

[1]https://github.com/RegNLP/ObliQADataset

Proceedings of the 31st International Conference on Computational Linguistics, pages 1–4
January 19–24, 2025.

task. For passage retrieval, the system combines BM25, dense retrieval models (e.g. DRAGON + and ColBERTv2), and rank fusion techniques to retrieve relevant passages. The answer generation component uses GPT-4-turbo-1106 with prompt engineering to synthesize obligation-focused answers from the retrieved passages.

2.4 Evaluation

To evaluate system performance, different metrics are applied to the two subtasks. For Subtask 1 (Information Retrieval), Recall at 10 (R@10) and Mean Average Precision at 10 (M@10) are used to assess the system's ability to retrieve relevant passages effectively. For Subtask 2 (Answer Generation), the Regulatory Passage Answer Stability Score (RePASs)[2] measures the quality of generated answers based on their entailment with source passages, avoidance of contradictions, and coverage of obligations.

3 Overview of Shared Task

The task was organized in time for COLING 2025 as part of the RegNLP 2025 workshop. A total of 19 teams participated, with 16 of them submitting both their system results and papers describing their approach.

During the development stage, the teams worked with the publicly available ObliQA dataset, which served as the primary resource for system training and fine-tuning. To support additional methodological exploration, the entire set of 40 hierarchically structured regulatory documents, from which the ObliQA dataset was derived, was also made available to participants.

In the evaluation stage, submissions were tested on a hidden subset of the ObliQA dataset consisting of 446 unseen questions. These questions were provided without access to their associated ground truth passages.

4 Overview of Teams' Methodologies

The participating teams in the RIRAG-2025 shared task employed diverse methodologies to address the challenges posed by the two subtasks. This section provides an overview of the approaches used by the teams, categorized by subtask.

[2] https://github.com/RegNLP/RePASs

4.1 Subtask 1: Information Retrieval

The participating teams employed a diverse range of methods for the information retrieval task, combining sparse retrieval, dense retrieval, hybrid systems, and re-ranking strategies to optimize passage retrieval for regulatory queries.

BM25 was a foundational component in many teams' systems, often augmented with additional techniques to enhance performance. Teams utilizing BM25 included USTC-IAT-United, NUST Nova, NUST Alpha, JurisCore, Ocean's Eleven, NLP-MindMappers, NLP-MJR, TEAM: 1-800, Indic aiDias, and AUEB. Hybrid systems were frequently implemented to balance lexical precision with semantic understanding. For example, USTC-IAT-United combined BM25, DRAGON+, ColBERTv2, and a fine-tuned LLaMA-2-7B model, employing a hybrid expert mechanism with dynamic weight assignment. Ocean's Eleven utilized BM25, NV-Embed-v2, and BGE-en-ICL embeddings, leveraging reciprocal rank fusion and NLI-based re-ranking to enhance retrieval relevance. Havelsan integrated bge-m3, e5-large-v2, and Jina embeddings, combined with context-aware chunking, to create a robust hybrid retrieval system.

Many teams further refined retrieval results using re-ranking models and dynamic filtering. For instance, AICOE employed text-embedding-ada-002 embeddings alongside RankGPT for sliding-window re-ranking, while Indic aiDias implemented a multi-stage tuning process with reciprocal rank fusion and context-based filtering. NLP-Alpacas applied msmarco-roberta-base-v2 and BAAI/bge-base-en-v1.5, using triplet-based fine-tuning and FAISS indexing for improved passage ranking.

Table 1 provides an overview of the teams and their respective methods.

4.2 Subtask 2: Answer Generation

The participating teams adopted various methods for the answer generation task, focusing on large language models (LLMs), prompt engineering, and post-processing strategies to produce accurate regulatory-aligned responses. Many teams employed state-of-the-art generative models to synthesize answers from retrieved passages. For instance, NLP-MindMappers and NUST Omega utilized Few-Shot prompting and Chain-of-Thought (CoT) techniques with GPT models to generate structured and comprehensive answers. Mean-

Table 1: Overview of Teams' Methodologies for Subtask 1: Information Retrieval

Paper ID	Team Name	Retrieval Methods	Key Features
11	USTC-IAT-United	LLaMA-2-7B fine-tuned + BM25 + DRAGON+ + ColBERTv2	Hybrid expert mechanism, Dynamic weight assignment
12	NUST Nova	LegalBERT + BM25 + FAISS + Neo4j Graph-Based Retrieval	Graph-based retrieval, Score fusion, Re-ranking
13	NUST Alpha	BM25 + FAISS	Rank fusion, GPT-based filtering , Re-ranking
14	NUST Omega	LegalBERT + Gemini + OpenAI embeddings + FAISS	Metadata-driven query matching, Topic modeling
15	Havelsan	bge-m3 + e5-large-v2 + Jina embeddings + hybrid search	Hybrid retrieval, Context-aware chunking, Re-ranking
16	Obayer	intfloat/multilingual-e5-large + txtai	
17	AICOE	text-embedding-ada-002 + RankGPT	Two-step retrieval, Sliding-window re-ranking
18	JurisCore	BM25 + Dense Retrieval + BDD-FinLegal	Cross-encoder re-ranking, Adaptive dynamic weighting
19	Ocean's Eleven	BM25 + NV-Embed-v2 + BGE-en-ICL	Reciprocal rank fusion, NLI-based re-ranking
20	NLP-MindMappers	BM25 + all-MiniLM-L6-v2 + FAISS	Bi-encoder retrieval, BM25 re-ranking, Multiple negatives ranking loss
21	NLP-Alpacas	msmarco-roberta-base-v2 + BAAI/bge-base-en-v1.5 + FAISS	Multiple negatives ranking loss, Triplet-based fine-tuning, FAISS-based indexing
22	NLP-MJR	BM25 + BAAI/bge-small-en-v1.5	Weighted score fusion, Semantic matching, Hybrid retrieval
23	TEAM: 1-800	BM25 + BGE-small-en-v1.5 + MPNet V2	Lexical-semantic score fusion, LeSeR reranking, MNSR fine-tuning
24	Indic aiDias	BM25 + BGE-EN-ICL + E5-FT + Q2Q	Reciprocal rank fusion, Context-based filtering, Multi-stage tuning
25	AUEB	BM25 + Voyage-Law-2 + Voyage-Finance-2 + Voyage-Rerank-2	Triple rank Fusion, Re-ranking
26	NLP-LingoLlamas	MiniLM-L6-v2 + stella en 400M v5 + Gemini-1.5-pro-002	Fine-tuning with negatives, Inverted re-ranking retrieval

Table 2: Overview of Teams' Methodologies for Subtask 2: Answer Generation

Paper ID	Team Name	Generative Models	Key Features
11	USTC-IAT-United	Qwen2-72B	Scoring-based passage filtering, Prompt
12	NUST Nova	Llama3-70b	Prompt
13	NUST Alpha	GPT-3.5	Prompt
14	NUST Omega	GPT *	Few-Shot, CoT, Prompt
15	Havelsan	LLaMA-3.1-8B-Instruct	Prompt
16	Obayer	—	
17	AICOE	GPT-4o	Prompt
18	JurisCore	—	
19	Ocean's Eleven	LLaMa-3.1-8B-Instruct, CFG, CAD	Prompt
20	NLP-MindMappers	Gemma 2B , GPT-4o	Few-Shot, CoT
21	NLP-Alpacas	T5-base, GPT-4o	Prompt
22	NLP-MJR	GPT 3.5 Turbo, GPT-4o Mini, Llama 3.1	Prompt
23	TEAM: 1-800	Qwen2.5 7B, Gemma-2 9B, Mistral 7B, Nemo 12B	Prompt
24	Indic aiDias	LLaMA-3.1-8B-Instruct, Single line, Identity function	
25	AUEB	GPT-4o Mini	Scoring and Obligation-based passage filtering, Post-Processing
26	NLP-LingoLlamas	Gemini-1.5-pro-002	Prompt

while, AUEB and USTC-IAT-United implemented passage filtering mechanisms to ensure the relevance and alignment of generated responses with regulatory obligations. Table 2 summarizes the models and key features utilized by each team.

5 Teams' Evaluation Results

The evaluation of team submissions was conducted separately for both subtasks.

The evaluation was based on a hidden subset of the ObliQA dataset consisting of 446 unseen questions. Table 3 provides a detailed breakdown of the scores for all teams and their submissions. Some teams submitted multiple versions of their systems, showcasing iterative improvements and different configurations.

Subtask 1 (Information Retrieval): The highest R@10 and M@10 scores were achieved by **Indic aiDias** with their first submission, scoring 0.787 and 0.663, respectively. Teams **NLP-MJR** (R@10: 0.731, M@10: 0.602) and **TEAM: 1-800** (R@10: 0.705, M@10: 0.562) also performed strongly in the retrieval subtask.

Subtask 2 (Answer Generation): The best RePASs score (0.973) was achieved by **Indic aiDias** with their first submission, closely followed by **Ocean's Eleven** (RePASs: 0.971) across two submissions. These teams demonstrated high entailment, contradiction avoidance, and obligation coverage in their generated answers. Teams **AUEB** and **NLP-MJR** also exhibited strong performance, with RePASs scores of 0.947 and 0.558, respectively.

6 Lessons from RIRAG-2025

The RIRAG-2025 shared task attracted a substantial number of participating teams from both academia and industry. This strong participation underscores the rapid growth and increasing interest in the RegNLP field.

A significant observation during the task was the limited integration of the hierarchical regulatory documents provided into the participants' approaches. Although the teams primarily used the ObliQA dataset, the rich interconnected structure of the entire set of regulatory documents was underutilized. Regulatory rules often refer to or build on one another, and understanding these relationships is crucial for generating accurate and comprehensive answers. Future shared tasks can address this perhaps by providing annotated examples of rule connections and offering detailed guidelines to help participants incorporate these relationships into their system designs.

Table 3: Evaluation Scores of Team Submissions for Subtasks 1 and 2 in the RIRAG-2025 Shared Task, based on a hidden dataset containing 446 questions.

Paper ID	**Team Name**	**R@10**	**M@10**	**Es**	**Cs**	**OCs**	**RePASs**
Baseline	BM25(passage-only)+GPT-4	0.761	0.624	0.310	0.120	0.176	0.455
Baseline	BM25(rank fusion)+GPT-4	0.764	0.625	0.312	0.125	0.152	0.446
11	USTC-IAT-United *	0.720	0.593	0.777	0.234	0.258	0.600
12	NUST - Group 3 - Team NOVA	0.393	0.227	0.358	0.307	0.109	0.387
13	NUST - Group 1- Team Alpha	0.672	0.521	0.505	0.109	0.098	0.498
14	NUST - Group 2 - Team Omega	0.585	0.097	0.489	0.239	0.167	0.473
15	Havelsan	0.677	0.541	0.330	0.278	0.161	0.404
16	Obayer*	0.780	-	-	-	-	-
17	AICOE	0.633	0.515	**0.827**	0.254	0.230	0.601
18	JurisCore - Submisson 1	0.314	0.093	0.208	0.577	0.005	0.212
	JurisCore - Submisson 2	0.650	0.503	0.395	0.378	0.109	0.375
	JurisCore - Submisson 3	0.650	0.503	0.177	0.716	0.028	0.163
19	Ocean's Eleven - Submission 1	0.686	0.548	**0.986**	0.065	**0.991**	**0.971**
	Ocean's Eleven - Submission 2	**0.694**	0.558	**0.986**	0.062	0.989	**0.971**
	Ocean's Eleven - Submission 3	0.693	0.554	**0.986**	0.149	**0.998**	0.945
20	NLP-MindMappers †	0.662	0.534	0.487	0.174	0.136	0.483
21	NLP-Alpacas * †	0.809	0.625	0.416	**0.046**	0.063	0.477
22	NLP-MJR	**0.731**	**0.602**	0.525	0.156	0.305	0.558
23	TEAM: 1-800	0.705	**0.562**	0.573	0.348	0.090	0.438
24	Indic aiDias - Submission 1	**0.787**	**0.663**	**0.987**	0.062	**0.993**	**0.973**
	Indic aiDias - Submission 2	**0.787**	**0.663**	0.092	**0.037**	0.444	0.316
	Indic aiDias - Submission 3	**0.787**	**0.663**	**0.987**	0.129	0.644	0.834
25	AUEB NLP Group - Submission 1	**0.694**	0.594	0.446	**0.031**	0.502	0.639
	AUEB NLP Group - Submission 2	**0.694**	0.594	0.375	0.110	0.423	0.562
	AUEB NLP Group - Submission 3	**0.694**	0.594	**0.986**	0.096	0.951	**0.947**
26	NLP-LingoLlamas †	0.611	0.499	0.422	0.218	0.048	0.418

Bold values represent the highest performance for each metric.
Teams marked with * could not be evaluated due to incomplete or invalid submissions. Results for these teams are extracted from the original team papers and correspond to evaluations on the ObliQA test set. All other results are based on the hidden dataset of 446 questions.
Teams marked with †did not finalize their camera-ready version for submission.

In the answer generation subtask, we employed RePASs, a metric specifically designed for RIRAG. However, we observed two critical areas for improvement. Firstly, RePASs is currently limited in its ability to evaluate verbatim reproduction of retrieved passages, which can affect the depth and originality of generated answers. Secondly, it lacks a mechanism to evaluate the fluency and cohesion of generated answers. To address these shortcomings, future iterations could enhance RePASs by incorporating penalties for excessive verbatim text and integrating components that assess linguistic quality. Specifically, we will explore the inclusion of semantic similarity thresholds to ensure that generated answers synthesize information rather than directly copying it. Additionally, we intend to incorporate LLM-based evaluations to measure fluency and cohesion, providing qualitative assessments of the generated text.

7 Conclusion

The RIRAG-2025 shared task showcased innovative approaches to tackling the challenges of regulatory information retrieval and answer generation. By leveraging the ObliQA dataset and a robust evaluation framework, participants were able to explore diverse methodologies, from hybrid retrieval systems combining sparse and dense models to advanced generative techniques supported by prompt engineering and post-processing.

While the task brought to light many promising methodologies, it also revealed areas for future exploration. The shared task has set a benchmark for further research in this domain, fostering collaboration between academic and industrial researchers and driving advancements in the automation of regulatory compliance tasks.

Acknowledgments

We would like to express our gratitude to the organizers of COLING 2025 for providing a platform to host the RIRAG-2025 shared task as part of the RegNLP workshop. We thank ADGM for their support in developing the dataset and for expert annotation. We also extend our thanks to all participating teams for their contributions to advancing the field of regulatory information retrieval and answer generation.

References

Tuba Gokhan, Kexin Wang, Iryna Gurevych, and Ted Briscoe. 2024. RIRAG: Regulatory Information Retrieval and Answer Generation. *Preprint*, arXiv:2409.05677.

Challenges in Technical Regulatory Text Variation Detection

Shriya Vaagdevi Chikati [a,c], **Samuel Larkin** [a], **David Minicola**[b], **Chi-kiu Lo 羅致翹**[a*]
[a]Digital Technologies Research Centre [b]Construction Research Centre
National Research Council Canada
[c]University of Waterloo
chikatishriya@gmail.com, {Samuel.Larkin, David.Minicola, Chikiu.Lo}@nrc-cnrc.gc.ca
*corresponding author

Abstract

We present a preliminary study on the feasibility of using current natural language processing techniques to detect variations between the construction codes of different jurisdictions. We formulate the task as a sentence alignment problem and evaluate various sentence representation models for their performance in this task. Our results show that task-specific trained embeddings perform marginally better than other models, but the overall accuracy remains a challenge. We also show that domain-specific fine-tuning hurts the task performance. The results highlight the challenges of developing NLP applications for technical regulatory texts.

1 Introduction

In Canada, the regulation of building construction is the jurisdiction of the provinces and territories (P/T). National Research Council of Canada is responsible for publishing the National Model Construction Codes[1] (NMCC), a collection of model construction codes, to help promote consistency among the P/T construction codes. The model construction codes set out the technical requirements for the design and construction of new buildings, as well as the change of use and demolition of existing buildings. The NMCC are adopted individually by each P/T legislatures while modifications are made to suit local needs. As a result, there are various kinds of variations between the NMCC and each of the P/T construction codes. Variations in construction codes create barriers to the free movement of construction goods, services, and investments within Canada. To support inter-P/T trades and the mobility of talents, harmonizing the construction codes across Canada is a key priority in the development and maintenance of the codes and tracking code variations is the first step towards code harmonization. Due to the high volume and high technicality of the construction codes, tracking variations in them is a difficult and expensive task that requires labor-intensive studies done by codes advisors. Some of the challenges in technical domain knowledge include identifying similarity/dissimilarity in definitions of technical terms or material specifications when they are applied in context, e.g. "sawdust" could be equivalent to "combustible pallets" in the fire codes. This study explores the potential of leveraging recent development of natural language processing (NLP) techniques to detect variations between the NMCC and the adopted codes in each P/T.

The NMCC consists of the National Building Code (NBC), the National Fire Code (NFC), the National Plumbing Code (NPC), and the National Energy Code for Buildings(NECB). Each P/T has the autonomy in deciding how the NMCC are adapted, or referenced. For example, for the 2015 edition of NBC, NFC, and NPC, New Brunswick, Nova Scotia, Manitoba and Saskatchewan adopted them with some modifications and additions; while Ontario developed their codes based on the national models, but with significant variations in content and scope and made major reorganization of the order of the provisions. The technical provisions of the construction codes (both the NMCC and the P/T construction codes) are structured, segmented and numbered and the detailed provisions are found at the sentence level. Hence, the majority of variations tracked at the sentence level. The variations in the construction codes are classified by National and P/T codes advisors into the following categories:

1. **P/T Only (Addition)** - The P/T has decided to add a new technical provision (i.e. a sentence) to their local construction codes.

2. **National Only (Deletion)** - The P/T has decided not to include a technical provision of

[1] https://nrc.canada.ca/en/certifications-evaluations-standards/codes-canada/codes-canada-publications

Proceedings of the 31st International Conference on Computational Linguistics, pages 5–9
January 19–24, 2025. ©COLING 2025

the NMCC in their local construction codes.

3. **Common Sentence** - The technical provision appear in both the the NMCC and the corresponding local construction codes. This category is further divided into two subcategories: a) adopted without changes or with editorial changes and b) modified with technical variations. Table 2 shows some examples of sentences with editorial changes vs. technical variations.

The sentences in the second example of editorial changes in table 2 of Appendix A are rewritten with a different focus, although remain technically the same. On the other hand, for the first example of technical variations, the sentences are nearly identical, except for the specified distance. This presents a huge challenge in classifying the sentences into the two subcategories because comprehensive technical knowledge is required to identify the technical variations. Therefore, in this preliminary study, we focus on the three main categories (i.e. P/T Only, National Only and Common Sentence) and we then formulate the task of variations classification as a sentence alignment task. "Common Sentence" in nation codes and P/T codes should be aligned while "P/T Only" and "National Only" should remain unaligned. The challenge here, though, is that the sentences in the two documents are not in the same order and we show that off-the-shelf sentence alignment tools perform poorly in this task.

In this paper, we study the potential and challenges of detecting variations between the NMCC and the construction codes in each P/T. We evaluate the performance of different sentence representation models on the sentence alignment task. Our results show that developing NLP applications for technical regulatory texts requires more investments and efforts.

2 Sentence Alignment Methods

Sentence alignment methods have two major components: 1) the sentence representation model for reflecting how similar sentences are to each other; and 2) alignment extraction approach for deciding the sentence alignment given the sentence similarity scores.

2.1 Sentence Representation

Representing sentences in a vector space enables numeric computation of their relations, such as similarity, which then enables practical tasks, like text classification and information retrieval. Similarity of sentences in vector space is usually computed using cosine similarity that measures the angular distance of their vectors. There are two generations of sentence representation models, distributional and neural.

2.1.1 Sparse distributional Vector Model

The most commonly used distributional text representation is the bag-of-words model (BoW). The BoW model considers a text as an unordered collection of words and the frequency of occurrence of each word in the text is the value of the dimension representing that word. Variants of the BoW model include 1-hot encoding that only considers the presence of each word rather than their frequency; and term frequency-inverse document frequency (tf-idf) weighted BoW that weights the frequency of each word by their tf-idf. Both variants intend to down-weight the influence of function words (more frequently occurring) and up-weight the influence of content words (less frequently occurring) in representing the text. Sparse distributional vector models suffer from the curse of dimensionality as the dimensions of the vector space grows with the size of the vocabulary of the sentences it represents.

2.1.2 Neural Sentence Embeddings

In contrast with distributional vector model, neural sentence embeddings are learned representation models. LASER (Artetxe and Schwenk, 2019) and LaBSE (Feng et al., 2022) are the most commonly used massively multilingual sentence embeddings models. While LASER and LaBSE are pretrained for general purpose sentence representation, some of the sentence embedding models are further trained for specific tasks to obtain better performance. We choose to experiment with a few sentence embedding models that perform well on the English semantic textual similarity task because our sentence alignment task relies on the sentence similarity scores. The models are bilingual-embedding-large (Javaness, 2024), multilingual-e5-large-instruct (Wang et al., 2024), mxbai-embed-large-v1 (Shakir et al., 2024) and GIST-embedding-v0 (Solatorio, 2024).

2.2 Alignment Extraction

State-of-the-art parallel text sentence alignment algorithms, such as Vecalign (Thompson and Koehn, 2019) and SentAlign (Steingrimsson et al., 2023), take advantage of parallel text where sentences in the two documents are roughly in monotonic order with local sentence reordering and use approximation or divide and conquer approaches to reduce the time and space complexity of the alignment extraction component in the algorithm. However, as we mentioned before, the P/T construction codes have major reorganization of the order of the provisions when comparing to the NMCC such that the assumption of monotonic sentence order does not hold for the task at hand.

The alignment extraction component is naturally a maximum weight bipartite matching problem. We consider all the sentences in one document as one partition of vertices in a graph, all the sentences in the other document as the other partition of vertices of the graph and the sentence similarity scores between all the sentence pairs as the weights of the edges. The matching produced by the maximum weight matching algorithm, that maximizes the sum of the weights of the edges included in the matching, is then the optimal sentence alignment. Thus, we propose to use the Scipy library[2] implementation of the LAPJVsp algorithm (Jonker and Volgenant, 1987) for the maximum weight matching problem in our experiment.

3 Experiments

The NMCC are updated and published once every five years and each of the P/T adopts the updated version in subsequent years. National and P/T codes advisors then work on analyzing the variations. The latest edition of NMCC with all variations tracked and classified was published in 2015 (Canadian Commission On Building And Fire Codes, 2015a,b,c; Canadian Commission On Building And Fire Codes and Natural Resources Canada, 2015). In our experiments, we use the English version of the 2015 edition of the NMCC and the corresponding P/T construction codes. We create the training and testing sets for our experiments by doing a random 80/20 split.

3.1 Evaluation Metric

To evaluate the performance of the construction code variation detection, we use the alignment error rate (AER) of the predicted alignment (Och and Ney, 2000). In our task, we only have the set of "sure" gold standard alignment (i.e. the Common Sentence labeled by domain experts) but not the set of possible alignment. The AER is, therefore, simplified as $AER = 1 - \left(\frac{2 \times |P \cap G|}{|P| + |G|}\right)$ where P is the set of predicted alignment and G is the set of gold standard alignment. The lower AER means that the alignment model is better.

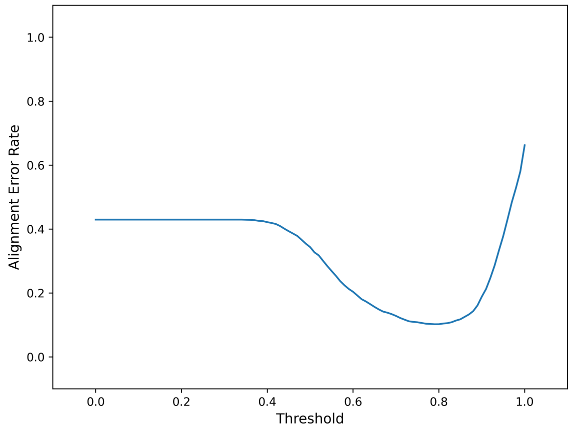

Figure 1: Training AER of the step-wise search for the optimal similarity score threshold for determining aligned sentence pairs.

3.2 Solution to Over-Alignment

The problem of using maximum weight matching to extract aligned sentence pairs is that it always returns a full matching such that every sentence in the smaller set is matched with a sentence in the other set. That means the extracted alignment would be over-aligned, in the sense that a sentence pair would be returned as aligned even with low similarity score. To reclassify the over-aligned sentence pairs into "P/T Only" and "National Only", we do a step-wise search on the training data to determine a threshold of similarity score for sentence pairs to remain aligned. As we increase the similarity score threshold for sentence pairs to remain aligned by 0.01 at each step, the AER drops until the threshold is too high and starts separating sentence pairs that ought to stay aligned to each other. This is being done for each individual model, including the domain fine-tuned model. Figure 1 shows a typical plot of the training AER against the similarity score threshold in the step-wise search. We then apply the optimal threshold that results in the lowest training AER to the test set.

3.3 Domain Specific Fine-tuning

Since the technical provisions in the construction codes are highly domain specific, we also experiment with fine-tuning the sentence representation model using contrastive training with the in-domain

[2] https://docs.scipy.org/doc/scipy/reference/generated/scipy.sparse.csgraph.min_weight_full_bipartite_matching.html

training data. We choose to fine-tune the GIST-Embedding-v0 model because it achieves the lowest training AER. Contrastive training leverages positive and negative sentence pairs to teach the model how to differentiate between similar and dissimilar sentences. This process helps the model to push dissimilar sentences further apart and pull similar sentences closer together in the embedding space. In our experiment, the positive pairs are the "Common Sentence" and we create the negative pairs by pairing the "National Only" sentences with the "P/T Only" sentences. Thus, the size of the fine-tuning data set is the same as the training data set, i.e. 80% of the complete NMCC, around four thousand sentence pairs. We used the default learning rate in fine-tuning.

4 Results

Table 1 shows the training and testing AER of the experiments on using different sentence representation models for detecting construction code variations. We see that the parallel text alignment baseline, Vecalign, is clearly not suitable for our task. Vecalign uses LASER as the underlying sentence representation model and assumes the sentences in the two input documents are in monotonic order. When we compare the performance of Vecalign with that of our experiment on the LASER model, we conclude that our proposal of using maximum weight matching algorithm for alignment extraction is more suitable for the task.

Model	Train	Test
parallel text alignment baseline		
Vecalign	0.4564	0.4568
distributional vector-based		
Bag-of-Words (BoW)	0.1402	0.1554
1-hot	0.1296	0.1426
tf-idf weighted BoW	0.1233	0.1372
pretrained sentence embeddings		
LASER	0.1635	0.1783
LaBSE	0.1352	0.1471
task specific trained embeddings		
bilingual-embedding-large	0.1183	**0.1306**
multilingual-e5-large-instruct	0.1210	0.1403
mxbai-embed-large-v1	0.1194	0.1366
GIST-embedding-v0	**0.1165**	0.1339
construction codes fine-tuned		
GIST-embedding-v0	0.1370	0.1522

Table 1: Training and testing AER of the experiments on using different sentence representation models for detecting construction code variations.

The tf-idf weighted bag-of-words model performs better than both of the pretrained sentence embedding models (LASER and LaBSE) and only marginally worse than the task specific trained embedding models.

The domain specific fine-tuning model performs significantly worse than the base model before fine-tuning. This is perhaps due to limited amount of in-domain data used in fine-tuning the model.

Overall, the AER for all our experiments are high. With an AER higher than 13% by all models before attempting to classify editorial and technical variation, we demonstrated that NLP research and development on technical regulatory texts remains an open question and great challenge.

5 Conclusion

In this study, we explored the potential of using current natural language processing techniques to detect variations between the National Model Construction Codes and the construction codes of different Canadian P/T. We evaluated various sentence representation models on this task. The overall bad performance across all models suggests that current NLP technologies are not yet fully equipped to handle the complexity of technical regulatory text. This highlights the need for further research in developing NLP models that could acquire the necessary technical knowledge from technical regulatory text and improving the accuracy and reliability of NLP tools in technical regulatory application.

Limitations and Ethical Considerations

Our work on automatic construction codes variation detection is intended to assist National and P/T codes advisors in tracking and analyzing variations, with the goal of harmonizing codes across Canada. The output of automatic construction codes variation detection *will NOT and should NOT* be used directly by any code users or code enforcement bodies before verification by technical experts in Canadian construction codes. As the accuracy of the current experiments are low, the risk of resulting in misleading information is high if the model output is directly used by code users or enforcement bodies. Misusing the model output could lead to financial loss, noncompliance or wrongful enforcement of construction codes. We think that it is of utmost importance to restrict the use of the model output to technical experts specialized in construction code variation identification.

Code	Sentence
Common Sentence (editorial)	
NBC	Where a fire safety plan is required, it shall conform to Section 2.8. of Division B of the NFC.
British Columbia BC	Fire safety plans shall conform to the British Columbia Fire Code.
NBC	Visual signal devices required by Sentence 3.2.4.19.(1) shall continue to emit a visible signal while voice instructions are being transmitted.
Ontario BC	The voice communication system referred to in Sentence (1) shall be designed so that visual signal devices are not interrupted while voice instructions are being transmitted.
Common Sentence (technical)	
NBC	The developed length of a building sewer between the building and the first manhole to which the building sewer connects shall not exceed 75 m.
Ontario BC	The developed length of a building sewer between the building and the first manhole to which the building sewer connects shall not exceed 30 m.
NFC	The removal, abandonment in place, disposal or temporary taking out of service of an underground piping system shall be in conformance with good engineering practice. (See Note A-4.3.16.1.(1).)
Alberta FC	Corrosion protection systems shall be maintained in operating condition when a storage tank is temporarily out of service and during seasonal shutdowns.

Table 2: Examples of sentences in the NMCC modified with editorial changes vs. technical variations in the P/T construction codes.

References

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Canadian Commission On Building And Fire Codes. 2015a. National building code of canada: 2015.

Canadian Commission On Building And Fire Codes. 2015b. National fire code of canada: 2015.

Canadian Commission On Building And Fire Codes. 2015c. National plumbing code of canada: 2015.

Canadian Commission On Building And Fire Codes and Natural Resources Canada. 2015. National energy code of canada for buildings: 2015.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

La Javaness. 2024. Bilingual-embedding-large.

R. Jonker and A. Volgenant. 1987. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340.

Franz Josef Och and Hermann Ney. 2000. A comparison of alignment models for statistical machine translation. In *COLING 2000 Volume 2: The 18th International Conference on Computational Linguistics*.

Aamir Shakir, Darius Koenig, Julius Lipp, and Sean Lee. 2024. Boost your search with the crispy mixedbread rerank models.

Aivin V. Solatorio. 2024. Gistembed: Guided in-sample selection of training negatives for text embedding fine-tuning. *Preprint*, arXiv:2402.16829.

Steinthor Steingrimsson, Hrafn Loftsson, and Andy Way. 2023. SentAlign: Accurate and scalable sentence alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 256–263, Singapore. Association for Computational Linguistics.

Brian Thompson and Philipp Koehn. 2019. Vecalign: Improved sentence alignment in linear time and space. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1342–1348, Hong Kong, China. Association for Computational Linguistics.

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.

A Examples of Common Sentence

Table 2 shows the example of aligned sentences in NMCC and P/T construction codes with different variation types (editorial vs. technical).

Bilingual BSARD: Extending Statutory Article Retrieval to Dutch

Ehsan Lotfi* **Nikolay Banar*** **Nerses Yuzbashyan** **Walter Daelemans**

CLiPS, University of Antwerp, Belgium
{ehsan.lotfi, nicolae.banari, nerses.yuzbashyan, walter.daelemans}
@uantwerpen.be

Abstract

Statutory article retrieval plays a crucial role in making legal information more accessible to both laypeople and legal professionals. Multilingual countries like Belgium present unique challenges for retrieval models due to the need for handling legal issues in multiple languages. Building on the Belgian Statutory Article Retrieval Dataset (BSARD, Louis and Spanakis (2022)) in French, we introduce the bilingual version of this dataset, bBSARD. The dataset contains parallel Belgian statutory articles in both French and Dutch, along with legal questions from BSARD and their Dutch translation. Using bBSARD, we conduct extensive benchmarking of retrieval models available for Dutch and French. Our benchmarking setup includes lexical models, zero-shot dense models, and fine-tuned small foundation models. Our experiments show that BM25 remains a competitive baseline compared to many zero-shot dense models in both languages. We also observe that while proprietary models outperform open alternatives in the zero-shot setting, they can be matched or surpassed by fine-tuning small language-specific models. Our dataset and evaluation code are publicly available.

1 Introduction

Open access to legal information is considered a fundamental right according to the Charter of Fundamental Rights in the European Union (European Union, 2012). Effective retrieval models are an essential component to ensuring this right, as they allow laypeople and legal professionals to efficiently search through vast amounts of legal information. In countries like Belgium, where laws are available in multiple languages (e.g. French and Dutch), the need for high-performance legal retrieval models becomes even more crucial, as they require equal accessibility to relevant legal material regardless of the language in use.

The retrieval task (Thakur et al., 2021) has experienced a significant boost due to the recent advances in textual embeddings, which rely on extensively pre-trained large language models (LLMs; Zhao et al., 2024). These models can encode text into vector representations which perform very well across a broad range of tasks (Muennighoff et al., 2023), including classification (Maas et al., 2011; Saravia et al., 2018; O'Neill et al., 2021) and clustering (Aggarwal and Zhai, 2012; Geigle et al., 2021). Open-source models like E5 (Wang et al., 2022, 2023, 2024) and BGE-M3 (Chen et al., 2024), along with private models from VoyageAI (2024) and OpenAI (2024) have shown remarkable results in zero-shot retrieval, across multiple languages, and different domains, including various legal benchmarks (Muennighoff et al., 2023). These developments offer great opportunities to improve accessibility in multilingual legal jurisdictions.

Belgium invests significant resources to consolidate[1] its laws in both French and Dutch, which is done by the manual labor of qualified legal professionals. This results in a highly valuable resource for research in multilingual legal retrieval models. Building on this resource, and the Belgian Statutory Article Retrieval Dataset (BSARD; Louis and Spanakis, 2022) in French, we introduce the Bilingual Belgian Statutory Article Retrieval Dataset (bBSARD), which we curated by scraping parallel Dutch and French articles, and translating the BSARD questions into Dutch. Using bBSARD, we conducted extensive benchmarking of retrieval models available for Dutch and French, both in zero-shot and fine-tuned scenarios.

In addition to a parallel bilingual legal corpus, bBSARD offers a much-needed retrieval benchmark for the Dutch language, allowing for more accurate and reliable evaluation of Dutch retrieval

*Indicates equal contribution

[1] https://www.ejustice.just.fgov.be/cgi_loi/contenu.pl?language=nl&view_numac=2019050815nl

Proceedings of the 31st International Conference on Computational Linguistics, pages 10–21
January 19–24, 2025. ©COLING 2025

models. bBSARD dataset and evaluation code are available on the HuggingFace hub[2] (under the `cc-by-nc-sa-4.0` license), and our GitHub repository[3] (under the MIT license), respectively.

2 Related Work

In the last few years, the field of legal NLP has gained increased interest, leading to the development of a growing number of datasets for research. In this section, we focus specifically on datasets that address the task of legal retrieval grounded in legal provisions, including documents, statutory articles, and cases.

CAIL2018 (Xiao et al., 2018; Zhong et al., 2018) is a dataset designed for legal judgment prediction in Chinese, released as part of the Chinese AI and Law Challenge[4]. It contains over 2.68 million Chinese criminal cases, linked to 183 law articles and 202 charges. One of the subtasks from this challenge involved predicting relevant law articles based on the factual descriptions of specific cases. Following this, the CAIL2019-SCM dataset (Xiao et al., 2019) focuses on similar case matching with 8,964 case triplets (in which two cases are similar) sourced from the Supreme People's Court of China.

Zhong et al. (2020) released JEC-QA, a question answering dataset based on the Chinese bar exam. The dataset contains 26,365 multiple-choice questions, along with 3,382 Chinese legal provisions.

The AILA competitions (Bhattacharya et al., 2019, 2021) introduced datasets for precedent and statute retrieval from Indian law, with content in English. For each year, around 50 queries were linked to relevant documents in retrieval corpora containing 197 statutes and around 3,000 prior cases.

Similarly, COLIEE (Rabelo et al., 2021, 2022; Kim et al., 2022; Goebel et al., 2024) competitions include the task of statute article retrieval from provided datasets. For each year, the datasets contain around 100 test questions from the Japanese legal bar exams, labeled with relevant articles from the Japanese Civil Code, translated into English. The provided training sets include up to 1000 question-article pairs.

Chen et al. (2023) introduced EQUALS, a dataset containing 6,914 question-article-answer triplets, with a corresponding retrieval corpus of 3,081 Chinese law articles. The question-answer pairs were collected from a free Chinese legal advice forum, then revised and further annotated by senior law students. Similarly, STARD (Su et al., 2024) introduced 1,543 queries from the general public, with a retrieval corpus of 55,348 Chinese statutory articles.

GerLayQA (Büttner and Habernal, 2024) consists of around 21,000 legal questions from laymen paired with answers from legal professionals and grounded in paragraphs from German law books.

Most related to our work is BSARD (Louis and Spanakis, 2022); a statutory article retrieval dataset which contains over 1,100 legal questions from Belgian citizens and around 22,600 Belgian law articles as the retrieval corpus. LLeQA (Louis et al., 2024b) complements BSARD with answers from legal experts, along with an additional 760 legal questions and 5,308 statutory articles. While LLeQA is a more extensive resource than BSARD, the latter is available under less restrictive terms[5] and does not require a separate user agreement[6].

The resources presented above support the training, evaluation, and benchmarking of retrieval models across different legal domains and languages, highlighting the need for tailored approaches in each jurisdiction. Contributing to the growing field of legal NLP, we introduce bBSARD, a bilingual dataset built on BSARD which offers parallel Belgian law articles in both French and Dutch, along with legal questions translated from French to Dutch. In addition to providing a reliable benchmark for the retrieval task in Dutch, bBSARD aims to address challenges of legal retrieval in multilingual jurisdictions.

3 Dataset

As mentioned, we base our work on the BSARD dataset (Louis and Spanakis, 2022), extending it to the Dutch language by adding the corresponding articles and questions. We discuss the procedure in the following sections.

3.1 Legislation in Dutch

To get the BSARD legislation and articles in Dutch, we leverage **Justel**[7], the multilingual database

[2] https://huggingface.co/datasets/clips/bBSARD
[3] https://github.com/nerses28/bBSARD
[4] http://cail.cipsc.org.cn

[5] https://huggingface.co/datasets/maastrichtlawtech/bsard
[6] https://huggingface.co/datasets/maastrichtlawtech/lleqa
[7] https://www.ejustice.just.fgov.be/cgi_loi/welcome.pl?language=nl

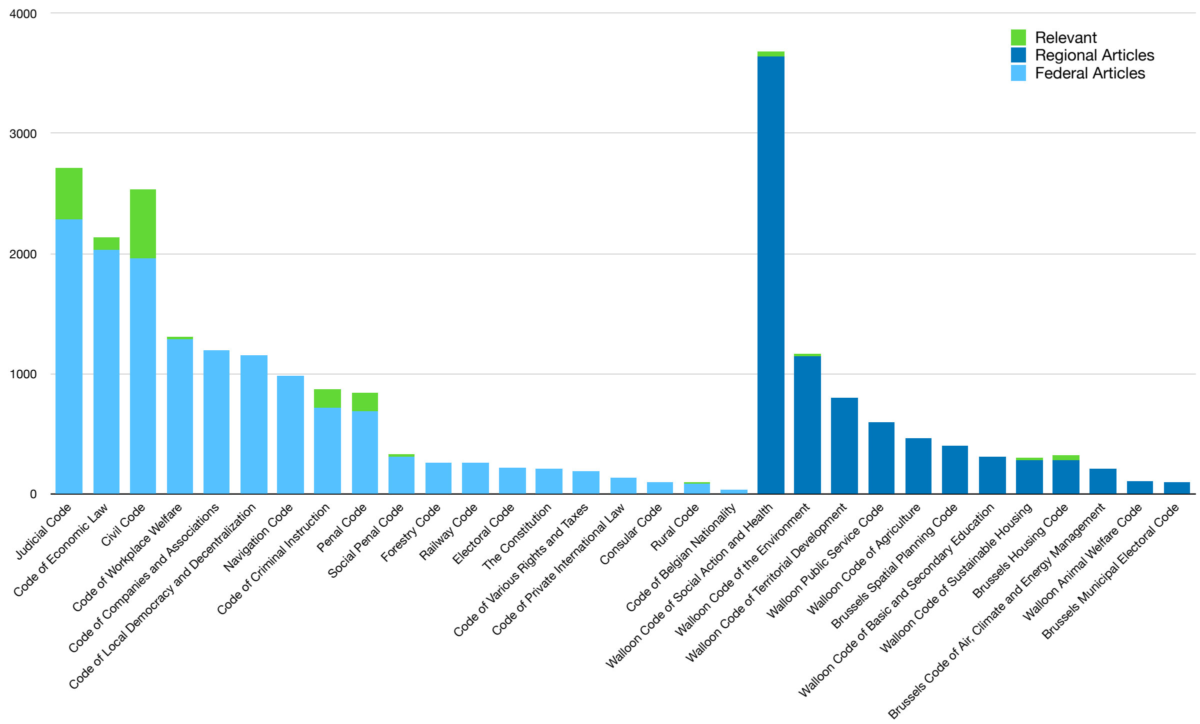

Figure 1: Distribution of different codes in the bBSARD article corpus. 'Relevant' articles (green) are the ones cited in the question set. Light and dark blue columns correspond to the Federal and Regional codes, respectively.

maintained by the Belgian Federal Government that provides online access to most Belgian legislation in French, Dutch and (often) German. Since there are no public APIs, we scrape the appropriate French and Dutch pages (52 pages for each language), according to the BSARD corpus. Considering the continuous changes and updates, and the fact that BSARD was curated in May 2021 (Louis and Spanakis, 2022), we make sure that the Dutch and French articles come from the same legislative version, by manually controlling their enforcement dates.

In the end, we manage to retrieve and align 22,417 out of 22,633 articles (99%) in both languages (see Appendix A for the alignment process.). The missing 216 articles mostly belong to the Walloon Code of Environment-Decrees (126 articles absent from the Dutch page), and the Military Penal Code (66 articles absent from the database). Fortunately, these missing articles contribute only marginally to the relevant subset (only 1 missing article is cited in a multi-referenced question). Table 3 in Appendix A summarizes the differences between the original and bilingual datasets.

Figure 1 shows how different codes contribute to the complete and relevant set of articles. The majority of relevant articles (i.e. annotated as necessary to answer questions, colored light green in the chart) come from four Federal codes: Judicial, Civil, Penal, and Criminal Instruction.

3.2 Questions in Dutch

BSARD contains 1,108 questions (split as 886/222 for the train/test sets), each labeled by experts with the IDs of the corresponding relevant law articles from the corpus. These questions have been curated in partnership with Droits Quotidiens[8], from emails sent by Belgian citizens to this organization, asking for advice on legal issues. They cover a wide range of topics, with around 85% of them being either about family, housing, money, or justice, while the remaining 15% concern either social security, foreigners, or work (Louis and Spanakis, 2022).

To produce these questions in Dutch, we opt for automatic translation followed by human inspection. We first prompt OpenAI's GPT-4o with the original French question, as well as the relevant articles (to provide context), and ask for the Dutch translation (The full prompt can be found in Appendix B). To increase translation fidelity, we set the temperature to 0 (Peng et al., 2023). We then asked a native speaker to examine a random sample of 100 translated questions, and annotate them for potential issues. The results showed (legally)

[8] https://droitsquotidiens.be/

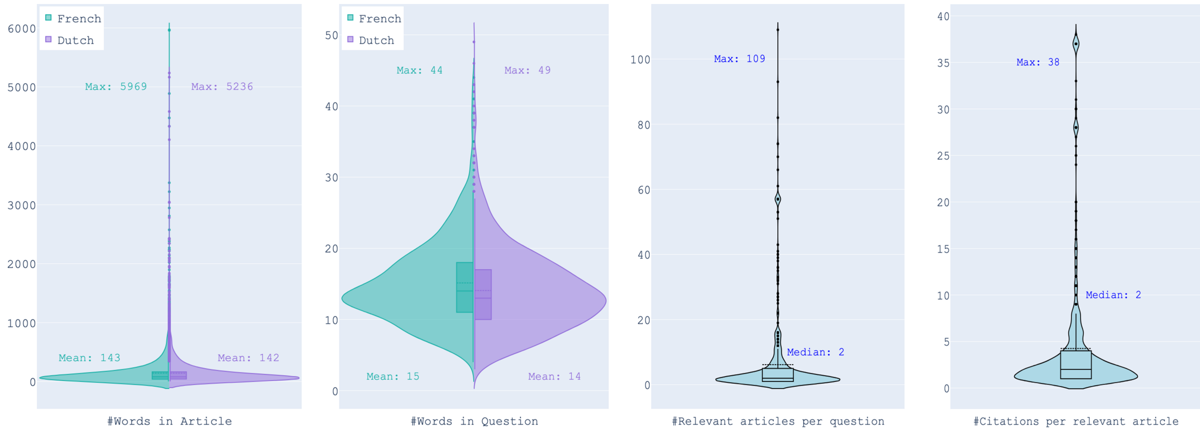

Figure 2: Basic statistics of bBSARD. From the left: Number of words in the articles (French and Dutch), Number of words in the questions (French and Dutch), number of relevant articles per question, and number of citations per relevant article.

inaccurate choice of words in 2%, and minor semantic/grammatical/lexical issues (e.g. translation being too literal) in 6% of the studied samples.

Figure 2 shows basic statistical features of the bBSARD dataset. The French and Dutch articles have an average length of 143 and 142 words, respectively, while for the questions these numbers stand at 15 and 14 words. Regarding the question-article mapping, 1,611 distinct articles (out of 22,417) contribute to the relevant subset, and 75% of questions have fewer than five references, with a median value of two.

4 Experimental Setup

This section describes our experimental setup used to benchmark the retrieval performance of a selection of models on bBSARD. We mostly reuse the codebase from BSARD (Louis and Spanakis, 2022), making modifications where necessary to accommodate the retrieval models to the specific requirements of our experiments. Below we describe the models, data processing steps, and evaluation metrics used in our experiments.

4.1 Models

We select a diverse range of models in three different categories/settings: lexical, zero-shot, and fine-tuned.

4.1.1 Lexical models

Lexical approaches for retrieval rely on keyword matching and utilize various word (or token) weighting schemes and algorithms to determine the relevance of documents for a given query. The most popular algorithms are TF-IDF (Term Frequency-Inverse Document Frequency; Sparck Jones, 1972; Salton and Yang, 1973) and BM25 (Best Match 25; Robertson et al., 1994). Despite the lexical gap issues, where the vocabulary used in queries can differ from that in relevant documents, BM25 remains a robust baseline for many retrieval tasks. Remarkably BM25 was outperformed only recently by E5 (Wang et al., 2022) on the BEIR retrieval benchmark (Thakur et al., 2021) in a setup that does not utilize any labeled data. In our experiments, we evaluate both TF-IDF and BM25.

4.1.2 Zero-shot models

Recently, LLMs achieved impressive results on various retrieval tasks (Zhao et al., 2024). For the zero-shot setting, we select the following multilingual retrieval models, from both open and proprietary categories: mContriever[9] (Izacard et al., 2022), LaBSE (Feng et al., 2022), mE5 (Wang et al., 2024), E5$_{mistral-7b}$ (Wang et al., 2023), BGE-M3 (Chen et al., 2024), DPR-XM (Louis et al., 2024a), BGE-Multilingual-Gemma2 (Li et al., 2024), jina-embeddings-v3 (Sturua et al., 2024), mGTE (Zhang et al., 2024), voyage-3 (VoyageAI, 2024), text-embedding-3-large (OpenAI, 2024). For models with a maximum input length of 512 tokens (except LaBSE), we divide the text into overlapping chunks of 200 tokens with an overlap of 20 tokens between neighboring chunks to mitigate the input length limitations. We do not impose any limits on the input length for other models, allowing them to handle truncation if necessary.

[9]https://huggingface.co/facebook/mcontriever-msmarco

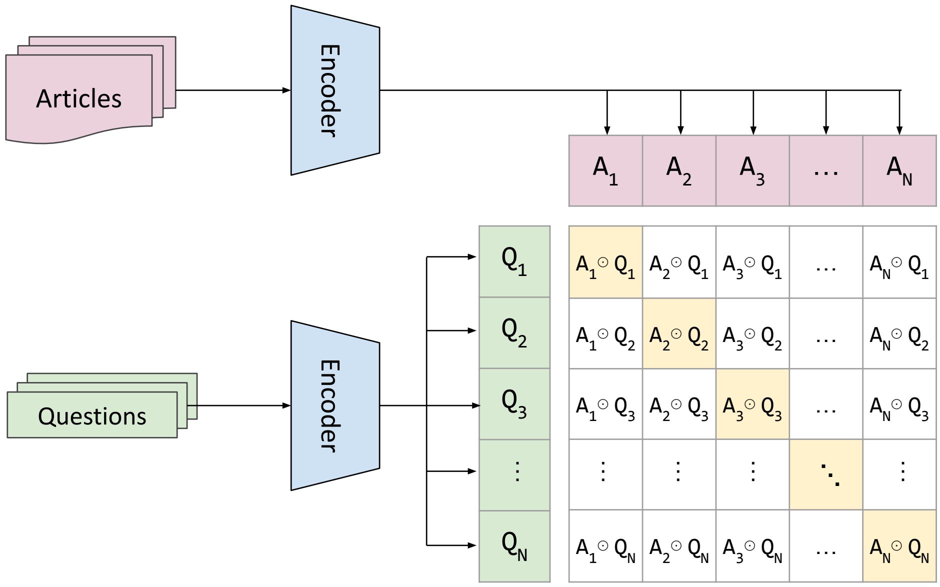

Figure 3: Standard Siamese Bi-Encoder setting with in-batch negatives, which we use for fine-tuning. Articles and Questions are encoded separately with the same model into vectors. For each question Q_i, the relevant article A_i is the positive sample, while all other articles in the batch are used as negatives. $\odot$ represents the cosine similarity operator.

In addition, we experiment with context-independent word embeddings, using word2vec (Mikolov et al., 2013b,a) for Dutch (Tulkens et al., 2016) and French (Fauconnier, 2015), as well as fastText (Bojanowski et al., 2017) for both Dutch and French (Grave et al., 2018). To construct embeddings of text chunks from these models, we apply mean-pooling to the word embeddings, with the exception of LaBSE, which uses the [CLS] token representation. In all cases, cosine similarity is employed to score similarity between the embeddings.

The evaluation is conducted on a single GPU with 48GB of RAM for E5$_{mistral-7b}$ and BGE-Multilingual-Gemma2. For other models, we use a single GPU with 8GB of RAM. Each experiment takes between five minutes for smaller models and up to 30 minutes for larger models.

4.1.3 Fine-tuned models

Foundation models can achieve competitive results compared to zero-shot retrieval models when fine-tuned on domain-specific data. In our evaluations, we select RobBERT-2023 (Delobelle and Remy, 2024) and Tik-to-Tok (Remy et al., 2023) for Dutch, and CamemBERT (Martin et al., 2020) and Flaubert (Le et al., 2020) for French. We also include XLM-Roberta to examine the potential advantage of language-specific models over the generic multilingual ones.

We primarily follow the experimental setup of BSARD and fine-tune the models in a Siamese setting (Reimers and Gurevych, 2019), which encodes the query and document via the same model (Figure 3). We optimize the contrastive loss with a temperature of 0.05 and in-batch negatives (Henderson et al., 2017; Karpukhin et al., 2020) with a batch size of 22. The optimization is performed using AdamW (Loshchilov and Hutter, 2017) with a learning rate of 2e-5, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay of 0.01. The learning rate undergoes a warm-up over the first 500 steps, followed by linear decay. Following Louis and Spanakis (2022), training is performed for 100 epochs, which takes 4.5-5.5 hours (depending on model size) on a single GPU with 24GB of RAM. Finally, we employ cosine similarity to score the embeddings.

4.2 Metrics

To assess the performance of our models, we employ standard retrieval metrics: macro-averaged recall@k (R@k), mean average precision@k (MAP@k), mean reciprocal rank@k (MRR@k), and normalized discounted cumulative gain@k (nDCG@k).

5 Results and Discussion

In this section, we present the performance of various retrieval models evaluated on the Dutch and French subsets of the bBSARD dataset (see Tables

T	Model	Size	R@100	R@200	R@500	MAP@100	MRR@100	nDCG@10	nDCG@100
	TF-IDF	-	39.21	46.38	52.76	8.53	14.25	12.38	16.74
	BM25	-	40.19	47.95	54.57	16.07	22.63	20.07	23.57
	word2vec	-	41.06	51.02	58.94	8.28	15.27	11.66	17.05
	fastText	-	31.47	38.26	49.08	7.27	12.67	8.89	13.86
	mE5$_{small}$	118M	45.43	52.25	61.10	13.42	21.79	17.67	22.79
	mContriever	178M	47.92	58.38	68.32	11.38	20.15	14.83	21.82
	DPR-XM	277M	40.44	46.12	53.16	13.57	21.78	16.40	21.79
	mE5$_{base}$	278M	50.14	57.68	65.30	16.47	25.64	20.81	26.49
	mGTE	305M	52.78	61.97	73.09	15.86	24.92	20.08	26.80
	LaBSE	471M	20.51	28.42	42.18	2.34	6.60	3.50	7.18
	mE5$_{large}$	560M	58.35	65.83	70.83	21.88	34.28	28.47	33.51
	mE5$_{large-instruct}$	560M	59.48	66.80	75.21	18.66	29.93	24.84	31.33
	BGE-M3	568M	61.12	67.20	77.56	18.31	30.40	24.04	31.21
	jina-embeddings-v3	572M	60.70	67.92	77.37	18.59	31.21	24.70	31.58
	E5$_{mistral-7b}$	7B	68.35	73.91	82.82	30.24	43.26	37.70	43.02
	BGE-Mult.-Gemma2	9B	69.94	76.23	81.28	25.07	37.66	30.95	39.11
	voyage-3	-	73.08	79.37	85.67	32.81	46.38	40.06	46.21
	embedding-3-large	-	**75.70**	**80.22**	**88.24**	29.73	42.99	36.83	44.40
✓	Tik-to-Tok$_{base}$	116M	73.90	79.02	83.29	39.24	45.69	42.75	49.90
✓	RobBERT-2023$_{base}$	125M	75.08	79.33	83.40	**40.51**	**47.68**	**44.76**	**51.36**
✓	XLM-Roberta$_{base}$	279M	62.06	68.26	75.40	26.61	32.00	30.65	37.10

Table 1: Retrieval performance of different models on the Dutch subset of bBSARD (test set). Evaluations are zero-shot for the dense models, except for the last 3 models (check-marked) which are fine-tuned.

T	Model	Size	R@100	R@200	R@500	MAP@100	MRR@100	nDCG@10	nDCG@100
	TF-IDF	-	41.69	51.05	60.22	8.74	12.85	11.34	17.45
	BM25	-	51.81	56.95	65.51	17.02	26.02	21.54	27.52
	word2vec	-	49.93	62.29	71.11	13.45	21.45	17.32	23.62
	fastText	-	24.84	32.36	43.88	5.05	10.03	7.40	10.65
	mE5$_{small}$	118M	46.26	51.74	59.25	13.67	23.48	18.49	23.03
	mContriever	178M	46.01	56.62	68.42	12.94	21.56	17.59	22.94
	DPR-XM	277M	40.91	47.34	55.13	10.83	19.31	14.31	19.74
	mE5$_{base}$	278M	47.62	56.60	63.60	16.76	26.25	21.90	26.28
	mGTE	305M	57.54	66.57	77.02	19.40	30.14	24.13	31.02
	LaBSE	471M	21.62	32.86	46.66	2.74	7.00	4.17	7.67
	mE5$_{large}$	560M	55.30	62.83	69.85	21.54	34.27	28.06	32.68
	mE5$_{large-instruct}$	560M	60.99	68.34	76.75	19.77	32.60	26.52	32.44
	BGE-M3	568M	60.76	69.02	79.81	19.40	31.38	25.38	32.08
	jina-embeddings-v3	572M	64.05	71.67	78.76	20.51	34.52	27.09	34.19
	E5$_{mistral-7b}$	7B	69.41	74.53	84.06	27.43	40.22	34.82	41.07
	BGE-Mult.-Gemma2	9B	71.44	77.81	83.73	30.06	43.72	36.36	43.46
	voyage-3	-	77.71	**82.68**	**88.76**	38.78	**54.60**	45.96	52.51
	embedding-3-large	-	75.47	80.70	87.58	33.72	46.51	40.54	47.33
✓	CamemBERT$_{base}$	111M	77.10	80.63	86.37	39.08	46.99	44.25	50.95
✓	FlauBERT$_{base}$	138M	**78.15**	81.59	85.84	**42.11**	49.82	**46.69**	**53.48**
✓	XLM-Roberta$_{base}$	279M	63.31	70.70	77.76	30.57	37.84	34.90	40.82

Table 2: Retrieval performance of different models on the French subset of bBSARD (test set). Evaluations are zero-shot for the dense models, except for the last 3 models (check-marked) which are fine-tuned.

1 and 2). In addition, we directly compare model effectiveness between two languages leveraging the parallel nature of the dataset.

5.1 Dutch Subset

As Table 1 shows, BM25 proves to be a strong baseline for the Dutch subset, with zero-shot dense models only fully outperforming it starting from 300 million parameters.

In the zero-shot setting, we observe a consistent improvement in performance as the model size grows, with the exception of LaBSE, which shows relatively lower results. The small-sized models (below 200M parameters), mE5$_{small}$ and mContriever, outperform the context-independent models (i.e. word2vec and fastText), and while mContriever achieves higher recall (R@100, R@200, R@500), mE5$_{small}$ is better across all other metrics. mE5$_{small}$ even outperforms the larger DPR-XM model in almost all metrics.

In the next zero-shot category (around 300M parameters), mGTE significantly outperforms mE5$_{base}$ in recall (R@100, R@200, R@500), while doing marginally worse across other metrics. For models up to 1 billion parameters, BGE-M3 and jina-embeddings-v3 show comparable results and are the best performers in recall (R@100, R@200, R@500), but E5$_{large}$ demonstrates superior performance in MAP@100, MRR@100, and nDCG (@10, @100). Finally, the largest open models, E5$_{mistral-7b}$ and BGE-Multilingual-Gemma2, outperform all other open models by a large margin. However, they lag behind proprietary models, voyage-3 and embedding-3-large, which are the best performers for the zero-shot setup.

As the lower section of the table shows, the high performance of proprietary models can be matched or topped by fine-tuning small models. In particular, fine-tuned RobBERT-2023$_{base}$ outperforms these models in MAP, MRR and nDCG metrics. Additionally, language-specific models demonstrate a significant advantage over the multilingual XLM-Roberta.

5.2 French Subset

Table 2 shows the results for the French subset of bBSARD. We observe trends similar to Dutch, with BM25 remaining competitive with the zero-shot dense models up to 300 million parameters.

Similarly, we observe a steady increase in performance in the zero-shot setup as the average model size increases, with the exception of LaBSE. Interestingly, the context-independent model word2vec outperforms not only the sub-200M models mE5$_{small}$ and mContriever, but also the larger DPR-XM model, while beating mE5$_{base}$ in recall. In the 300M-parameter category, mGTE outperforms the larger mE5$_{large}$ model in recall (R@100, R@200, R@500), and competes with BGE-M3 in MAP@100, MRR@100, and nDCG (@10, @100). Among the models up to 1 billion parameters, jina-embeddings-v3 achieves the highest performance in recall (R@100, R@200), MRR@100, and nDCG@100, while BGE-M3 performs better in recall@500, and E5$_{large}$ demonstrates the best results in nDCG@10. The largest open models, E5$_{mistral-7b}$ and BGE-Multilingual-Gemma2, show superior performance over other open options. However, the proprietary models, voyage-3 and embedding-3-large, outperform them by a large margin, with voyage-3 showing the best overall performance. Finally, we see the competitive performance of small fine-tuned models, with FlauBERT$_{base}$ beating voyage-3 in 4 out of 7 metrics.

5.3 Cross-Language Comparison

As bBSARD is a parallel dataset, we can directly compare Tables 1 and 2 to gain deeper insights into performance discrepancies between the French and Dutch models.

On average, models show a higher performance on the French subset compared to the Dutch subset (see Table 4 in Appendix C). This is perhaps most notable in BM25 and word2vec, where French models outperform their Dutch counterparts by more than 10 recall points (the clear outlier is fastText, which performs significantly better on the Dutch subset.) In addition, mGTE, jina-embeddings-v3, and voyage-3 do significantly better on the French subset than the Dutch. Other models gain 2-3 additional recall points in Dutch and perform comparably well across other metrics for both languages, with the exception of E5$_{mistral-7b}$ and DPR-XM. These models show slightly lower recall (R@100, R@200, R@500) in Dutch but achieve higher scores in other metrics. Finally, the best fine-tuned performer in French, FlauBERT$_{base}$, outperforms the top performer in Dutch, RobBERT-2023$_{base}$, and XLM-Roberta gains 3-5 points higher results when trained and evaluated on French.

In addition to potential translation issues (see 3.2) which particularly affect lexical models, one intuitive hypothesis on the origin of this advantage concerns the significant difference in data availability between the two languages. For example, while the original RobBERT model was pre-trained on a 39 GB corpus (Delobelle et al., 2020), CamemBERT and FlauBERT used 138 GB and 71 GB of data, respectively[10] (Le et al., 2020). However, further analysis is required to determine the significance of this factor, as well as other contributing parameters.

6 Conclusions and Future Work

In this paper we presented bBSARD, the bilingual version of the BSARD dataset (Louis and Spanakis, 2022). To curate bBSARD, we scraped parallel Dutch and French articles from the online Justel database and translated the BSARD questions into Dutch. In addition to a parallel bilingual legal corpus, bBSARD offers a much-needed retrieval benchmark for the Dutch language, allowing for more accurate and reliable evaluation of Dutch retrieval models.

Based on our dataset, we conducted extensive benchmarking of the retrieval task (ranking passages by their relevance to a given query) for Dutch and French, both in zero-shot and fine-tuned scenarios. These experiments confirm the status of simple lexical methods like BM25 as strong baselines, the superiority of closed-source commercial models like Voyage and OpenAI in zero-shot setting, and the possibility of outperforming them via fine-tuning small language-specific models like RobBERT and FlauBERT. We also observed an overall advantage for French compared to Dutch, in both zero-shot and fine-tuning scenarios.

We hope that our work encourages and facilitates the development of better Dutch retrieval models in the legal domain, which are an essential part of popular LLM-based methods like RAG. In the future, we would like to first improve bBSARD's quality by manually checking/correcting all translated questions, and then expand our work beyond the legal domain by curating and providing a more comprehensive benchmark for the retrieval task in Dutch. Another interesting research avenue considers the cross-lingual training potentials offered by a bilingual parallel dataset. In our experiments, we observed that XLM-Roberta performs better in Dutch when finetuned for 50+50 epochs on French+Dutch data, compared to 100 epochs on the Dutch subset. This suggests the possibility of leveraging the bilingual structure for additional gains in performance, specially for the lower-resource language, but to examine and explore its real significance more experiments need to be conducted.

[10] 282 GB and 270 GB before filtering/cleaning.

Limitations

We primarily inherit limitations from BSARD, as this dataset serves as the foundation for our work. The retrieval corpus is limited to the 32 Belgian codes from federal (Belgian) and Walloon law. As a result, bBSARD does not cover the whole of Belgian law, particularly omitting codes specific to Flanders. In addition, these limitations make the retrieval process incomplete as a part of relevant articles might be missing. Since we scraped the Belgian articles from around May 2021, bBSARD does not contain the updated version of the Belgian law.

Given these limitations, bBSARD is not intended for obtaining any comprehensive legal information or advise. Its primary purpose is to benchmark retrieval models and gain insights into the current state of the art. In accordance with the BSARD license (`cc-by-nc-sa-4.0`), we release our dataset under the same terms.

Acknowledgments

This research received funding from the Flemish Government under the “Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen” programme. We would like to thank Franci Haest for contributing to the data collection and to Luna De Bruyne for checking the quality of the translations. We also acknowledge the use of the GPT-4o model for assisting with error checking and proofreading of this paper.

References

Charu C Aggarwal and ChengXiang Zhai. 2012. A survey of text clustering algorithms. *Mining text data*, pages 77–128.

Paheli Bhattacharya, Kripabandhu Ghosh, Saptarshi Ghosh, Arindam Pal, Parth Mehta, Arnab Bhattacharya, and Prasenjit Majumder. 2019. Fire 2019 aila track: Artificial intelligence for legal assistance. In *Proceedings of the 11th annual meeting of the forum for information retrieval evaluation*, pages 4–6.

Paheli Bhattacharya, Parth Mehta, Kripabandhu Ghosh, Saptarshi Ghosh, Arindam Pal, Arnab Bhattacharya, and Prasenjit Majumder. 2021. Fire 2020 aila track: Artificial intelligence for legal assistance. In *Proceedings of the 12th Annual Meeting of the Forum for Information Retrieval Evaluation*, FIRE '20, page 1–3, New York, NY, USA. Association for Computing Machinery.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

Marius Büttner and Ivan Habernal. 2024. Answering legal questions from laymen in german civil law system. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2015–2027.

Andong Chen, Feng Yao, Xinyan Zhao, Yating Zhang, Changlong Sun, Yun Liu, and Weixing Shen. 2023. Equals: A real-world dataset for legal question answering via reading chinese laws. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, pages 71–80.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2402.03216.

Pieter Delobelle and François Remy. 2024. Robbert-2023: Keeping dutch language models up-to-date at a lower cost thanks to model conversion. *Computational Linguistics in the Netherlands Journal*, 13:193–203.

Pieter Delobelle, Thomas Winters, and Bettina Berendt. 2020. RobBERT: a Dutch RoBERTa-based Language Model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3255–3265, Online. Association for Computational Linguistics.

European Union. 2012. Charter of fundamental rights of the european union. `https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:12012P/TXT`. [Accessed: 5 October 2024].

Jean-Philippe Fauconnier. 2015. French word embeddings.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic bert sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891.

Gregor Geigle, Nils Reimers, Andreas Rücklé, and Iryna Gurevych. 2021. Tweac: transformer with extendable qa agent classifiers. *arXiv preprint arXiv:2104.07081*.

Randy Goebel, Yoshinobu Kano, Mi-Young Kim, Juliano Rabelo, Ken Satoh, and Masaharu Yoshioka. 2024. Overview and discussion of the competition on legal information, extraction/entailment (coliee) 2023. *The Review of Socionetwork Strategies*, 18(1):27–47.

Édouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomáš Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *arXiv preprint arXiv:1705.00652*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Mi-Young Kim, Juliano Rabelo, Randy Goebel, Masaharu Yoshioka, Yoshinobu Kano, and Ken Satoh. 2022. Coliee 2022 summary: methods for legal document retrieval and entailment. In *JSAI International Symposium on Artificial Intelligence*, pages 51–67. Springer.

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, and Didier Schwab. 2020. Flaubert: Unsupervised language model pre-training for french. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2479–2490, Marseille, France. European Language Resources Association.

Chaofan Li, MingHao Qin, Shitao Xiao, Jianlyu Chen, Kun Luo, Yingxia Shao, Defu Lian, and Zheng Liu. 2024. Making text embedders few-shot learners. *arXiv preprint arXiv:2409.15700*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Antoine Louis, Vageesh Saxena, Gijs van Dijck, and Gerasimos Spanakis. 2024a. Colbert-xm: A modular multi-vector representation model for zero-shot multilingual information retrieval. *arXiv preprint arXiv:2402.15059*.

Antoine Louis and Gerasimos Spanakis. 2022. A statutory article retrieval dataset in french. In *Proceedings of the 60th Annual Meeting of the Association for*

Computational Linguistics (Volume 1: Long Papers), pages 6789–6803.

Antoine Louis, Gijs van Dijck, and Gerasimos Spanakis. 2024b. Interpretable long-form legal question answering with retrieval-augmented large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 22266–22275.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. Camembert: a tasty french language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Niklas Muennighoff, Nouamane Tazi, Loic 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*, pages 2014–2037.

OpenAI. 2024. New embedding models and api updates. Accessed: 2024-10-31.

James O'Neill, Polina Rozenshtein, Ryuichi Kiryo, Motoko Kubota, and Danushka Bollegala. 2021. I wish i would have loved this one, but i didn't–a multilingual dataset for counterfactual detection in product review. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7092–7108.

Keqin Peng, Liang Ding, Qihuang Zhong, Li Shen, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. 2023. Towards making the most of ChatGPT for machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5622–5633, Singapore. Association for Computational Linguistics.

Juliano Rabelo, Randy Goebel, Mi-Young Kim, Yoshinobu Kano, Masaharu Yoshioka, and Ken Satoh. 2022. Overview and discussion of the competition on legal information extraction/entailment (coliee) 2021. *The Review of Socionetwork Strategies*, 16(1):111–133.

Juliano Rabelo, Mi-Young Kim, Randy Goebel, Masaharu Yoshioka, Yoshinobu Kano, and Ken Satoh. 2021. Coliee 2020: methods for legal document retrieval and entailment. In *New Frontiers in Artificial Intelligence: JSAI-isAI 2020 Workshops, JURISIN, LENLS 2020 Workshops, Virtual Event, November 15–17, 2020, Revised Selected Papers 12*, pages 196–210. Springer.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Conference on Empirical Methods in Natural Language Processing*.

François Remy, Pieter Delobelle, Bettina Berendt, Kris Demuynck, and Thomas Demeester. 2023. Tik-to-tok: Translating language models one token at a time: An embedding initialization strategy for efficient language adaptation. *arXiv preprint arXiv:2310.03477*.

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).

Gerard Salton and Chung-Shu Yang. 1973. On the specification of term values in automatic indexing. *Journal of documentation*, 29(4):351–372.

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. Carer: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 3687–3697.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Saba Sturua, Isabelle Mohr, Mohammad Kalim Akram, Michael Günther, Bo Wang, Markus Krimmel, Feng Wang, Georgios Mastrapas, Andreas Koukounas, Nan Wang, et al. 2024. jina-embeddings-v3: Multilingual embeddings with task lora. *arXiv preprint arXiv:2409.10173*.

Weihang Su, Yiran Hu, Anzhe Xie, Qingyao Ai, Zibing Que, Ning Zheng, Yun Liu, Weixing Shen, and Yiqun Liu. 2024. Stard: A chinese statute retrieval dataset with real queries issued by non-professionals. *arXiv preprint arXiv:2406.15313*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Stéphan Tulkens, Chris Emmery, and Walter Daelemans. 2016. Evaluating unsupervised dutch word embeddings as a linguistic resource. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4130–4136.

VoyageAI. 2024. Voyage 3. Accessed: 2024-10-31.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*.

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

Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Yansong Feng, Xianpei Han, Zhen Hu, Heng Wang, et al. 2018. Cail2018: A large-scale legal dataset for judgment prediction. *arXiv preprint arXiv:1807.02478*.

Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Tianyang Zhang, Xianpei Han, Zhen Hu, Heng Wang, et al. 2019. Cail2019-scm: A dataset of similar case matching in legal domain. *arXiv e-prints*, pages arXiv–1911.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, Meishan Zhang, Wenjie Li, and Min Zhang. 2024. mgte: Generalized long-context text representation and reranking models for multilingual text retrieval. *Preprint*, arXiv:2407.19669.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2024. Dense text retrieval based on pretrained language models: A survey. *ACM Transactions on Information Systems*, 42(4):1–60.

Haoxi Zhong, Chaojun Xiao, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Yansong Feng, Xianpei Han, Zhen Hu, Heng Wang, et al. 2018. Overview of cail2018: Legal judgment prediction competition. *arXiv preprint arXiv:1810.05851*.

Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. Jec-qa: a legal-domain question answering dataset. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9701–9708.

A Appendix: Scraping and Aligning the Articles

Table 3 shows a detailed summary of codes scraped from the **Justel** portal for bBSARD, compared to the original BSARD dataset. For alignment, we first leverage the article names/codes (e.g. Art. 14bis), and then use an automatic pipeline (consisting of a length comparison filter followed by ChatGPT queries) to spot the absent, misaligned, or non-aligned articles, which we then add and/or align manually. The alignment issues are mainly due to rare discrepancies in the way articles are registered in French and Dutch pages, or between French pages and BSARD dataset (for example 'Art. 14.2' vs. 'Art. 14/2').

B Appendix: Translating the Questions

To translate the questions into Dutch, we prompt `GPT-4o`[11] with the following instruction and context for each question (temperature = 0).

Prompt:`"You will be provided with a legal question and a related article from Belgian legislation. Your task is to translate the question from French to Dutch. The article serves solely as context to ensure the accuracy in legal understanding and terminology, so do not include any part of it in the translation. Return only the translation of the question without any additional information.`

```
<article>: {article} </article>
<question>: {question} </question>
question translated to Dutch:"
```

We also translate the 3 meta-fields available for each question in BSARD, i.e. `category`, `subcategory`, `extra_description` (although they are not used in the experiments). For this, we first refer to the `www.helderrecht.be` website (the Dutch version for `www.droitsquotidiens.be`), and extract the available corresponding categories and subcategories (35% of the total). We then use these translation pairs as examples to prompt `GPT-4o` to translate the rest of the phrases.

C Appendix: Comparison of French and Dutch Results

Table 4 shows the average retrieval performance for different model types on the French and Dutch subsets of bBSARD (test set).

[11] gpt-4o-2024-08-06

Authority	Code	BSARD #Articles	BSARD #Relevant	bBSARD #Articles	bBSARD #Relevant
Federal	Judicial Code	2285	429	2283	429
	Code of Economic Law	2032	98	2032	98
	Civil Code	1961	568	1961	568
	Code of Workplace Welfare	1287	25	1287	25
	Code of Companies and Associations	1194	0	1193	0
	Code of Local Democracy and Decentralization	1159	3	1158	3
	Navigation Code	977	0	977	0
	Code of Criminal Instruction	719	155	719	155
	Penal Code	689	154	689	154
	Social Penal Code	307	23	307	23
	Forestry Code	261	0	261	0
	Railway Code	260	0	260	0
	Electoral Code	218	0	217	0
	The Constitution	208	5	208	5
	Code of Various Rights and Taxes	191	0	189	0
	Code of Private International Law	135	4	134	4
	Consular Code	100	0	100	0
	Rural Code	87	12	87	12
	Military Penal Code	66	1	0	0
	Code of Belgian Nationality	31	8	31	8
Regional	Walloon Code of Social Action and Health	3650	40	3643	40
	Walloon Code of the Environment	1270	22	1143	22
	Walloon Code of Territorial Development	796	0	795	0
	Walloon Public Service Code	597	0	597	0
	Walloon Code of Agriculture	461	0	461	0
	Brussels Spatial Planning Code	401	1	401	1
	Walloon Code of Basic and Secondary Education	310	0	310	0
	Walloon Code of Sustainable Housing	286	20	279	20
	Brussels Housing Code	279	44	279	44
	Brussels Code of Air, Climate and Energy Management	208	0	208	0
	Walloon Animal Welfare Code	108	0	108	0
	Brussels Municipal Electoral Code	100	0	100	0
	Total	22633	1612	22417	1611

Table 3: Distribution of codes in BSARD and bBSARD (this work). "Relevant" articles are meant with respect to the question set.

T	Model Type	Lang.	R@100	R@200	R@500	MAP@100	MRR@100	nDCG@10	nDCG@100
	Lexical	FR	**46.75**	**54.00**	**62.87**	**12.88**	**19.44**	**16.44**	**22.49**
		NL	39.70	47.17	53.67	12.30	18.44	16.23	20.16
	CI dense	FR	**37.39**	**47.33**	**57.50**	**9.25**	**15.74**	**12.36**	**17.14**
		NL	36.27	44.64	54.01	7.78	13.97	10.28	15.46
	CD dense	FR	**56.79**	**64.24**	**72.81**	**20.54**	**31.83**	**26.09**	**31.89**
		NL	56.00	63.02	71.51	19.17	29.79	24.35	30.52
✓	CD dense	FR	**72.85**	**77.64**	**83.32**	**37.25**	**44.88**	**41.95**	**48.42**
		NL	70.35	75.54	80.70	35.45	41.80	39.39	46.12

Table 4: Average retrieval performance per model type on bBSARD (test set). CI and CD refer to context-independent and context-dependent models, respectively. All dense models are evaluated in zero-shot setting, except for the lower section (check-marked) which are fine-tuned.

Unifying Large Language Models and Knowledge Graphs for efficient Regulatory Information Retrieval and Answer Generation

Kishore Vanapalli
Carleton University, Ottawa, Canada
kishorevanapalli@cmail.carleton.ca

Aravind Kilaru
Carleton University, Ottawa, Canada
aravindkilaru@cmail.carleton.ca

Omair Shafiq
Carleton University, Ottawa, Canada
omairshafiq@cunet.carleton.ca

Shahzad Khan
Gnowit Inc., Ottawa, Canada
shahzad@gnowit.com

Abstract

In a rapidly changing socio-economic landscape, regulatory documents play a pivotal role in shaping responses to emerging challenges. An efficient regulatory document monitoring system is crucial for addressing the complexities of a dynamically evolving world, enabling prompt crisis response, simplifying compliance, and empowering data-driven decision-making. In this work, we present a novel comprehensive analytical framework, PolicyInsight, which is based on a specialized regulatory data model and state-of-the-art NLP techniques of Large Language Models (LLMs) and Knowledge Graphs to derive timely insights, facilitating data-driven decision-making and fostering a more transparent and informed governance ecosystem for regulators, businesses, and citizens.

1 Introduction

1.1 Problem Statement

Regulatory policy monitoring (Waterman and Wood, 1993) refers to the systematic process of observing, tracking, and analyzing the policies and regulations established by regulatory bodies. The primary goal is to stay informed about any changes, updates, or new developments in regulatory policies that may affect various sectors, industries, or the general public. This monitoring process involves continuous observation, change detection, impact analysis, and compliance monitoring.

1.2 Importance

Monitoring and tracking regulatory policies are highly important for businesses for several reasons, such as regulatory compliance, risk mitigation, strategic planning, operational efficiency, and market intelligence.

1.3 Difficulty

However, regulatory policy monitoring can be a challenging task due to various factors which include frequent policy changes, diverse regulatory frameworks, legislative complexity, lack of centralized information, data security and privacy challenges, and technological and automation challenges.

1.4 Solution

In this work, our objective was to develop an efficient and comprehensive regulatory document monitoring framework with the following features:
Real-time monitoring: The framework involves real-time monitoring of regulatory policy documents, ensuring that the information is always up-to-date.
Adaptability to Changes: With a novel policy data model, the system is designed to seamlessly adapt to changes in the structure or content of policy documents. It can dynamically adjust to modifications in document formats, new policy sections, or alterations in the way information is presented.
Intelligent Analytical Insights: State-of-the-art NLP techniques and LLMs (Pouyanfar et al., 2018; Zhou et al., 2020) are leveraged for better understanding and categorization of policy content and derive change detection and impact analysis.
Responsive User Interface: The user interface of the monitoring system is responsive and user-friendly. It allows users to interact with the data that are interested in a dynamic manner, facilitating efficient exploration, analysis, and tracking of policy updates.
Automated Alerts and Notifications: The dynamic approach includes the implementation of automated alert systems. Users can receive notifications in real time when significant policy changes occur, allowing for prompt response and analysis.
Scalability and performance: The system is

Proceedings of the 31st International Conference on Computational Linguistics, pages 22–30
January 19–24, 2025.

designed to scale efficiently, accommodating an increasing volume of policy documents and users. Performance optimization is a key aspect to ensure that the dynamic monitoring process remains efficient even as the dataset grows.

1.5 Scope

The scope of the paper is limited to the development of the following foundational features: Design and development of a novel and efficient data model (Devedžić, 1999) to organize, store, access and efficiently manage policy data. Using this data model, relationships between different policies or different versions of the same policy can be easily derived and utilized. Also, the new data model with the aid of relationships and constraints helps to derive key insights from the underlying policy data. Development of an advanced, intelligent and configuration driven Policy Monitoring Component which can collect, extract and store various policy data. Development of a sophisticated Policy Analytical System based on LLMs and Knowledge Graphs to achieve policy deduplication, policy impact analysis and policy change predictions. The policy data model and Knowledge Graph populated data, Cypher queries, sample LLM prompts, and evaluation results are shared in the project GitHub page[1].

2 Literature Review

Pan et al. (2024) proposed a futuristic roadmap for the unification of LLMs and Knowledge Graphs (KGs) to simultaneously leverage their advantages and proposed a roadmap which consists of three broad frameworks, specifically, 1) KG-enhanced LLMs, which leverage KGs during the pre-training and inference phases of LLMs, or for improving understanding of the knowledge gained by LLMs; 2) LLM-augmented KGs, that incorporates LLMs for different KG tasks such as embedding, graph-to-text generation, construction, completion and question answering; and 3) Synergized LLMs + KGs, in which LLMs and KGs, both provide equal contributions and work in a mutually beneficial way to improve both LLMs and KGs for bidirectional reasoning driven by both data and knowledge. Overall, the authors highlighted how LLMs and KGs complement each other in effectively addressing common challenges in several downstream tasks like Question-Answering, Hallucination detection and Reasoning.

Knowledge Graphs (KGs), which represent semantic relationships between entities, have shown significant relevance for NLP. Schneider et al. (2022) presented the results of an extensive survey, offering a multi-perspective review of tasks, research types, and contributions. It provides a structured overview of the research landscape, including a broad categorization of tasks, a summary of findings, and highlighted directions for future work after systematically analyzing over five hundred papers on Knowledge Graphs in NLP. The findings indicate that a wide range of tasks related to KGs in NLP have been studied across various domains, including emerging topics like knowledge graph embedding and augmented language models.

In the survey paper on Knowledge Graphs (KGs), Ji et al. (2022) provided a comprehensive review of knowledge graph covering overall research topics about 1) knowledge graph representation learning, 2) knowledge acquisition and completion, 3) temporal knowledge graphs, and 4) knowledge-aware applications, and summarize recent breakthroughs and perspective directions to facilitate future research. However, the paper fails to address some key aspects of KGs particularly while building and maintaining KGs and the way to overcome such challenges.

The survey by Abu-Salih (2021) is pioneering in providing a comprehensive definition of a domain-specific Knowledge Graph. Additionally, the paper conducts an extensive review of state-of-the-art approaches from academic works across seven domains of knowledge. However, it remains unclear why the discussed challenges cannot be generalized to domain-agnostic KGs, making it difficult to apply the solutions universally to any Knowledge Graph.

Dessì et al. (2021) introduced an innovative architecture that leverages natural language processing and machine learning (ML) techniques to extract entities and relationships from research publications, integrating them into a large-scale knowledge graph. However, as the paper notes, there are some limitations to the developed pipeline. For example, the current version does not fully utilize the semantic characterization of research entities to verify the resulting triples.

Johann Höchtl and Schöllhammer (2016) seeks

[1]Project GitHub page: https://github.com/Kishorevb/policyinsight

to bridge the gap between e-governance and public administration theories, moving beyond the predominantly service delivery-focused approach in much of e-government research. By utilizing the policy cycle as a model for policy processes and development, the article presents an innovative perspective on policy decision-making through the use of ICT and Big Data. It explores the delicate balance between the socially beneficial uses of Big Data and the potential harm to privacy and other values. This raises complex questions about how to detect, measure, and address discriminatory effects that may arise from automated decision-making processes.

Bui et al. (2021) framed the extraction of detailed personal data phrases and associated data collection or sharing practices as a sequence-labeling problem, addressable through an entity-recognition model. The authors developed an entirely automated system named PI-Extract, which uses a neural model to accurately extract privacy practices and significantly outperforms strong rule-based baselines.

Valle-Cruz et al. (2020) aimed at evaluating the public policy-cycle framework in the context of AI, focusing on the actual and anticipated changes that these emerging technologies will introduce at different stages of the policy-making process.

To achieve intelligent analysis of a large number of regulatory policies, Wang et al. (2023) proposes a discourse parsing technique designed for an in-depth understanding of Chinese government documents (CGDs). Utilizing Superstructure Schema and Rhetorical Structure Theory (RST), the paper examines the stylistic characteristics and macrostructure patterns of CGDs, and it develops a discourse analysis framework to define their functional structure and semantic system. Experimental results indicate that the parsing model, which incorporates inherent CGD discourse features, outperforms baseline models. However, despite its high accuracy, the proposed approach may face challenges when applied to cross-format government policies in the real world.

3 Overall Architecture

In this section, we first provide an overview of PolicyInsight's high-level architecture. Then, we dive into the main design decisions in the framework.

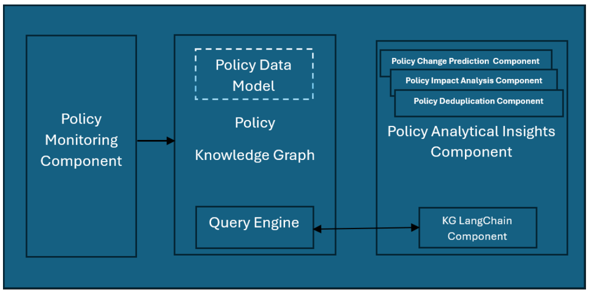

Figure 1: The system architecture of PolicyInsight

3.1 Overview

The PolicyInsight framework is based on four foundational functional components: a dynamic policy data model, a policy knowledge graph built from policy data model entities and relationships, a policy monitoring component and an analytical insights component.

3.2 Policy Data Model

Designing a dynamic data model to represent regulatory policies requires careful consideration of the evolving nature of policies, the diverse range of policy components, and the need for flexibility and scalability. When designing a policy data model, several key considerations must be taken into account to ensure its effectiveness, adaptability, and security. Firstly, it's crucial to identify key entities and attributes within the policy domain, capturing essential elements of policies and their associated metadata. Additionally, defining policy states and incorporating versioning and history tracking mechanisms allows for the monitoring and management of policy changes over time. Finally, prioritizing data integrity and security measures safeguards sensitive policy information, ensuring confidentiality, integrity, and availability throughout the data life cycle.

The policy data model is designed for the bylaws open data (of Ottawa, 2024). A bylaw is a rule or regulation enacted by a local authority, such as a city council or municipal government, to govern conduct, activities, and operations within a specific jurisdiction. Bylaws are subordinate to higher-level laws and are typically enacted to address local issues, maintain order, and regulate various aspects of community life. Moreover, Policies from different government bodies, such as federal, state, and local authorities, are interconnected and often interact in complex ways due to the shared jurisdictional responsibilities, overlapping regulatory

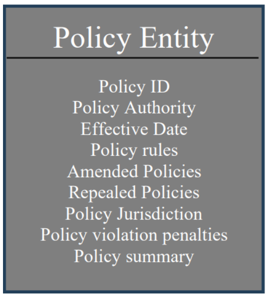

Figure 2: A partial view of Policy Data Model.

Figure 3: Policy lifecycle flow.

frameworks, and intergovernmental relations.

Our policy data model consists of several data entities and their relations (sample in Figure 2). Example entities include Policy entity, Stakeholder entity, Policy Document entity and so on. Similarly, example entity relationships include Policy entity to Stakeholder entity and Policy entity to Policy Document entity. For complete policy data model please refer to project GitHub page.

A typical policy lifecycle consists of several stages or phases that a policy undergoes from its initial conceptualization to its eventual termination or replacement. While the specific stages may vary depending on the context, jurisdiction, and nature of the policy, the following are common stages observed in many policy lifecycles.

3.3 Policy Knowledge Graph

Knowledge graphs in the system help capturing real-time policy data and mitigate issues such as hallucination and poor explainability. Unlike LLMs, which rely on static training data and may generate responses that are not grounded in reality, KGs can be updated in real time to reflect changing policy circumstances. This allows KGs to provide more accurate and reliable information, reducing the risk of hallucination. Additionally, KGs' transparent and interpretable structure enables explainability, as relationships between entities are explicitly defined, making it clear why a particular response was generated. By incorporating real-time data into KGs, organizations can ensure that their decision-making processes are informed by the most up-to-date information, reducing the likelihood of errors and biases associated with LLMs.

PolicyInsight Policy Knowledge Graph is based on the popular graph database Kùzu (Salihoglu, 2023; Inc., 2023), a highly scalable, extremely fast and easy-to-use embeddable database which allows graph-based modeling and querying, graph-optimized storage and graph-optimized query execution. As an extension to the database and querying module, we built a GUI for user input and querying.

Building a knowledge graph in Kùzu from the prepared policy data consists of two primary steps: Creating schema with the designed entities and relationships as Tables and populating tables with prepared CSV data files. As outlined in Section 3.2, which focuses on the design of the policy model schema, we established a data model of entities and their relationships, resulting in the creation of triplets in the form of (entity1, relationship1, entity2) that comprise the knowledge graph. The complete details of Knowledge Graphs schema can be found in the project GitHub page. With the schema fully defined and populated, the knowledge graph is now primed for querying and analysis.

Cypher (Kùzu, 2023) is Kùzu's graph query language that enables data retrieval from the graph. Much like SQL for relational databases, it was inspired by SQL, allowing you to concentrate on the desired data from the graph without worrying about the retrieval process. Given a query objective, like SQL, Cypher also provisions several ways to perform queries to retrieve desired outcome using several languages constructs like query and subquery clauses (Kùzu, 2023).

3.4 Policy Monitoring Component

When designing a policy monitoring component, several critical considerations must be addressed to ensure its effectiveness in tracking policy developments, assessing impacts, and facilitating adaptive governance processes. Firstly, real-time or near real-time updates are essential to provide timely information on policy changes, enabling stakeholders to stay informed and responsive to evolving policy landscapes.

We designed a policy monitoring tool based on a web crawler designed to systematically and automatically collect, analyze, and aggregate policy-related information from various online sources, including regulatory websites, legislative databases, news portals, and other relevant platforms. The tool is configured to identify and prioritize specific sources of policy information, such as regulatory websites, legislative databases, regulatory agencies, and reputable news outlets. This ensures that the collected data is reliable, authoritative, and up to date. Upon extraction, the tool performs content analysis and classification to categorize policy-related information based on predefined topics, keywords, or themes. The tool provides real-time updates and alerts on policy developments, changes, and announcements.

The extracted policy data is stored in a Knowledge Graph for easy access, retrieval, and analysis. Overall, a policy monitoring tool based on a web crawler streamlines the process of collecting, analyzing, and monitoring policy-related information from online sources, empowering policymakers, analysts, and stakeholders to stay informed, responsive, and proactive in addressing policy challenges and opportunities.

3.5 Analytics Insights Component

The primary goal of designing a policy analytical insights component was to enable comprehensive analysis and decision-making support for policymakers and stakeholders. The Analytics Insights Component consists of three subcomponents: Policy Changes Summarization component, Policy Impact Analysis component and Policy Change Predictions component.

Firstly, the component should incorporate policy change summarization capabilities to distill complex policy updates into concise, digestible summaries, facilitating quick understanding of key changes and their implications. Policy changes summarization design flow consists of data preprocessing where the policy documents are preprocessed to remove noise, such as headers, footers, and boilerplate text, and tokenize the text into sentences and paragraphs. And then an LLM was used to generate summaries of policy changes. This involves providing the model with input text (e.g., a section of a policy document) and prompting it to generate a concise summary of the content. The model generates summaries by predicting the most relevant and informative sentences or phrases based on the input context. For the evaluation, the quality of the generated summaries is evaluated using metrics such as ROUGE (Recall-Oriented Understudy for Gisting Evaluation), which measures the overlap between the generated summaries and reference summaries (e.g., human-authored summaries).

Similarly, using an LLM for Policy Impact Analysis involves leveraging its capabilities in NLU and generation to assess the effects and implications of policy interventions. The first step involves gathering relevant data sources, including policy documents, legislative texts, regulatory reports, news articles, and social media discussions related to the policy under analysis. These sources provide context and information about the policy's objectives, implementation, and outcomes. Then, fine-tune a pre-trained LLM on a dataset containing policy-related texts and documents. Provide prompts or queries to the fine-tuned LLM to prompt it to generate assessments or predictions about the policy's impact. For example, prompt the model with questions such as "What are the potential economic effects of implementing this policy?" or "How might this policy impact different demographic groups?" The LLM generates impact analyses by predicting potential outcomes, consequences, and implications of the policy under consideration. Evaluate the quality and validity of the generated impact analyses using expert review, validation against empirical data, or comparison with existing impact assessments.

For Policy Change Predictions, the designed workflow involves gathering a comprehensive dataset of historical policy documents, legislative texts, regulatory reports, news articles, and other relevant sources that document past policy changes and developments. This dataset serves as the training data for the LLM. Then, fine-tune a pre-trained LLM on the historical policy dataset. For example, prompt the model with questions such as "What policy changes are likely to occur in the next year based on historical trends?" or "Which policy areas are expected to see significant changes?" The LLM generates policy change predictions by analyzing patterns, correlations, and signals in the historical data. Evaluate the quality and accuracy of the generated policy change predictions using metrics such as precision, recall, and F1-score. Validate the predictions against empirical data or expert judgments to assess their reliability and usefulness.

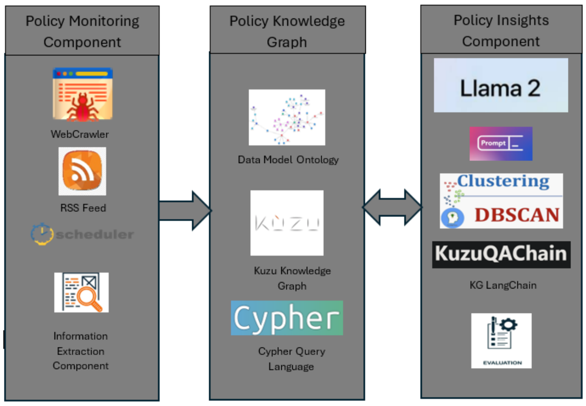

Figure 4: An overview of PolicyInsight.

4 Implementation Paradigm

The overall structure of PolicyInsight and the connections between different modules are illustrated in Figure 4. In this section, we discuss the detailed implementation of three main components of PolicyInsight: the monitoring subsystem, streaming data management subsystem, and the three-layered monitoring subsystem. We show how to combine different technologies to achieve a high-performance data-analytics system for PolicyInsight.

4.1 Policy Monitoring Component

At the core of the system is a custom web crawler designed to efficiently traverse regulatory websites, regulatory portals, legislative databases, and other online sources to collect policy-related data. The web crawler employs intelligent algorithms to navigate complex website structures, extract relevant information, and filter out noise and irrelevant content. Depth-First crawling strategy was used with Time-based rate limiting considering the overnight update of policies.

In addition to web crawling, the system incorporates RSS feed mechanisms to subscribe to policy-related feeds from authoritative sources, government agencies, industry publications, and news outlets.

Additionally, integrating Llama 2, an LLM, with suitable prompts further enhances the system's capabilities. Llama 2 can be utilized for NLP tasks such as policy summary generation, obligations detection, and risks identification. Leveraging LLM's capabilities allows for comprehensive analysis of policy text, enabling the generation of concise summaries and the extraction of obligations (e.g., regulatory requirements, compliance mandates) and potential risks associated with policy provisions.

To maintain data integrity and reliability, quality assurance measures are implemented to validate the accuracy, completeness, and relevance of extracted policy insights. Validation checks, error handling mechanisms, and human-in-the-loop review processes are incorporated to ensure the reliability and integrity of the output generated by the system.

4.2 Policy Knowledge Graph

In the process of building a policy knowledge graph, the system leverages a pre-designed policy data model to structure the information extracted from the JSON output generated in the previous step of the policy monitoring component. This predesigned data model serves as a blueprint for organizing policy-related entities, relationships, and attributes in a structured and consistent manner.

The first crucial step in this process involves mapping the entities identified in the JSON output to the corresponding entity types defined in the policy data model. Entities such as policies, regulations, stakeholders, and risks are matched with their counterparts in the data model, ensuring alignment between the extracted information and the predefined entity schema.

Once the entities are mapped, the system proceeds to establish relationships between them based on the predefined relationship types defined in the policy data model. Relationships such as "is_related_to", "imposes_obligation_on," and "addresses_risk" are identified and established between entities, capturing the connections and dependencies between different policy elements.

With the entities and relationships mapped and established, the system populates the knowledge graph, accordingly, creating nodes for each entity type and edges for each relationship type.

In implementing Cypher queries to extract crucial insights from the policy knowledge graph, the system capitalizes on the expressive capabilities of Cypher, a graph query language specifically designed for graph databases.

4.3 Policy Analytical Insights Component

The implementation of the policy analytical insights component leverages the emergent abilities of LLMs to analyze extensive repositories of policy documents, legislative texts, and regulatory frameworks.

4.3.1 Implementing Policy Change Summarization Component

The implementation of policy change summarization began with the extraction of article summaries using LLM Llama 2 (Touvron et al., 2023) by applying prompt techniques (Liu et al., 2023) (Varadarajan and Hristidis, 2006), which enabled the system to distill key insights and highlights from a vast array of policy documents and legislative texts. However, the initial approach of clustering these summaries led to a significant number of false positives, as similar policy articles were erroneously grouped together due to semantic overlaps or contextual similarities. To address this challenge, the system augmented the policy summary data with rich metadata sourced from the knowledge graph, a technique known as KG-enhanced LLMs (Pan et al., 2024), encompassing attributes such as policy maker, jurisdiction, regulatory domain, and effective date.

To incorporate policy metadata from the knowledge graph into the summaries generated by LLM we used a LangChain (Topsakal and Akinci, 2023) based tool for KuzuDB called KuzuQAChain (langchain ai, 2024), so that the system can gain additional contextual information and domain-specific insights that facilitated more accurate deduplication of policy articles.

Through this iterative approach, the system achieved a significant reduction in false positives and improved the accuracy of policy deduplication by leveraging the complementary capabilities of article summaries and policy metadata from the knowledge graph.

4.3.2 Implementing Policy Impact Analysis Component

Policy impact analysis was implemented through a multi-faceted approach that began with the generation of policy core areas or topics derived from the analysis of policy documents and regulatory frameworks using Llama 2 using appropriate prompt technique (Liu et al., 2023) (Varadarajan and Hristidis, 2006). Subsequently, following a different approach of LLM-augmented KGs to unify LLMs with KGs (Pan et al., 2024), these policy core areas were stored within a knowledge graph, enriching the graph with contextual information and semantic relationships that facilitated comprehensive impact analysis.

4.3.3 Implementing Policy Prediction Component

The policy prediction component was implemented to harness the synergistic capabilities of both LLMs and knowledge graphs by using a technique called Synergized LLMs + KGs (Pan et al., 2024), for the predictive analytics in the policy domain. At its core, this component employed advanced NLP techniques powered by Llama 2 to analyze vast repositories of unstructured textual data comprising policy documents, legislative texts, and regulatory frameworks. By training on historical policy data and learning from nuanced linguistic patterns, Llama 2 could generate plausible scenarios, anticipate emerging policy trends, and forecast future regulatory changes with remarkable accuracy.

4.4 Evaluation Results

In this section, we would like to present evaluation results of two use cases to assess the efficacy of unifying the capabilities of LLMs and Knowledge Graphs in policy analysis which revealed remarkably high accuracy results for both the policy deduplication and policy impact analysis tasks.

Use case 1: Policy deduplication results Objective: The primary objective of this task is to identify and remove duplicate policies from a dataset containing policies from overlapping jurisdictions but serving the same purpose.

Test Data: The test data comprises a curated selection of policy samples sourced from overlapping jurisdictions, enacted for both similar and disparate purposes, and meticulously hand-labeled for evaluation purposes. Experiment 1 approach: DBScan clustering was performed to cluster policy summaries generated by employing Llams 2 with prompts.

Results: An overall accuracy of 85% was achieved by using LLMs only due to the huge number of false positives (Figure 5).

Experiment 2 approach: DBScan clustering was performed to cluster policy summaries generated by employing Llams 2 with prompts. But this time policy summaries are augmented with corresponding policy metadata like policy maker and jurisdiction, policy effective data etc. coming from the policy knowledge graph.

Results: Overall accuracy was boosted to 95% with a reduced number of false positives (Figure 6).

Use case 2: Policy Impact Analysis results

Objective: The primary objective of this task

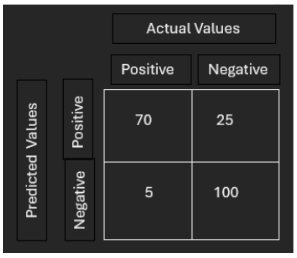

Figure 5: Confusion matrix when only LLM capabilities are employed for policy deduplication.

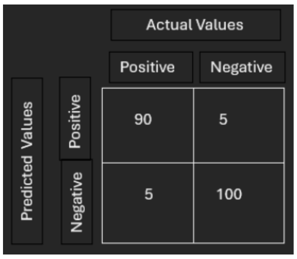

Figure 6: Confusion matrix when KG-enhanced LLMs are employed for policy deduplication.

is to identify the customers impacted by a policy change.

Test Data: The test data comprises a curated selection of policy samples sourced from a policy body and labeled automatically using breadcrumb approach for evaluation purposes.

Experiment approach: LLM-augmented KGs approach was employed in which Llama 2 was used with prompt to identify key impacted areas of a given policy and fed that information to Policy KG along with other derived policy information. During inference, Policy KG was queried to match with customer business domains to identify impacted customers.

Results: An overall accuracy of 89% was achieved by this approach.

5 Conclusions

In conclusion, our work introduces PolicyInsight, a novel analytical framework designed to address the evolving challenges of regulatory document monitoring in a rapidly changing socio-economic landscape. By leveraging a sophisticated policy data model and state-of-the-art NLP and knowledge graph techniques in a combined fashion, PolicyInsight enables stakeholders to continuous monitoring and derive timely insights from policy documents, fostering data-driven decision-making. Incorporating a novel dynamic policy data model for a scalable and efficient knowledge graph, PolicyInsight leverages an innovative unified approach to combining capabilities of both LLMs and KGs to achieve remarkable accuracy for policy deduplication, policy impact analysis and policy changes prediction. By providing stakeholders with access to actionable insights derived from policy data, PolicyInsight empowers policymakers, businesses, and citizens to make informed decisions, respond effectively to crises, and comply with regulatory requirements. Looking ahead, the continued refinement and expansion of PolicyInsight holds immense potential for driving positive change in governance practices. Future research endeavors may focus on enhancing the scalability, interoperability, and predictive capabilities of PolicyInsight, thereby enabling stakeholders to anticipate regulatory changes, identify emerging trends, and proactively address societal challenges. Our future work also addresses the few remaining items from the framework. In summary, PolicyInsight stands at the forefront of innovation in policy monitoring and analysis, offering a powerful tool for navigating the complexities of the modern regulatory landscape and fostering a more transparent, informed, and responsive governance ecosystem.

References

Bilal Abu-Salih. 2021. Domain-specific knowledge graphs: A survey. *Journal of Network and Computer Applications*, 185:103076.

Duc Viet Bui, Kang G. Shin, Jong-Min Choi, and Junbum Shin. 2021. Automated extraction and presentation of data practices in privacy policies. *Proceedings on Privacy Enhancing Technologies*, 2021:88 – 110.

Danilo Dessì, Francesco Osborne, Diego Reforgiato Recupero, Davide Buscaldi, and Enrico Motta. 2021. Generating knowledge graphs by employing natural language processing and machine learning techniques within the scholarly domain. *Future Generation Computer Systems*, 116:253–264.

V. Devedžić. 1999. A survey of modern knowledge modeling techniques. *Expert Systems with Applications*, 17(4):275–294.

Kùzu Inc. 2023. Kùzu: A highly scalable, extremely fast, easy-to-use embeddable graph database. `https://kuzudb.com/`.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2022. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):494–514.

Peter Parycek Johann Höchtl and Ralph Schöllhammer. 2016. Big data in the policy cycle: Policy decision making in the digital era. *Journal of Organizational Computing and Electronic Commerce*, 26(1-2):147–169.

Cypher: Kùzu. 2023. Cypher manual. `https://docs.kuzudb.com/cypher/`.

langchain ai. 2024. Langchain kùzu. `https://python.langchain.com/docs/integrations/graphs/kuzu_db/`.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

City of Ottawa. 2024. By-laws, licences and permits. `https://ottawa.ca/en/living-ottawa/laws-licences-and-permits`.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599.

Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and S. S. Iyengar. 2018. A survey on deep learning: Algorithms, techniques, and applications. *ACM Comput. Surv.*, 51(5).

Semih Salihoglu. 2023. Kùzu: A database management system for "beyond relational" workloads. *SIGMOD Rec.*, 52(3):39–40.

Phillip Schneider, Tim Schopf, Juraj Vladika, Mikhail Galkin, Elena Simperl, and Florian Matthes. 2022. A decade of knowledge graphs in natural language processing: A survey. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 601–614, Online only. Association for Computational Linguistics.

Oguzhan Topsakal and T. Cetin Akinci. 2023. Creating large language model applications utilizing langchain: A primer on developing llm apps fast. *International Conference on Applied Engineering and Natural Sciences*, 1:1050–1056.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melissa Hall Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

David Valle-Cruz, J. Ignacio Criado, Rodrigo Sandoval-Almazán, and Edgar A. Ruvalcaba-Gomez. 2020. Assessing the public policy-cycle framework in the age of artificial intelligence: From agenda-setting to policy evaluation. *Government Information Quarterly*, 37(4):101509.

Ramakrishna Varadarajan and Vagelis Hristidis. 2006. A system for query-specific document summarization. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, CIKM '06, page 622–631, New York, NY, USA. Association for Computing Machinery.

Xiaoyu Wang, Hong Zhao, Hongzhi Zhu, and Fang Wang. 2023. Towards intelligent policy analysis: A discourse structure parsing technique for chinese government document. *Information Processing & Management*, 60(4):103363.

Richard W Waterman and B Dan Wood. 1993. Policy monitoring and policy analysis. *Journal of Policy Analysis and Management*, 12(4):685–699.

Ming Zhou, Nan Duan, Shujie Liu, and Heung-Yeung Shum. 2020. Progress in neural nlp: Modeling, learning, and reasoning. *Engineering*, 6(3):275–290.

A Hybrid Approach to Information Retrieval and Answer Generation for Regulatory Texts

Jhon Rayo
Universidad de los Andes
Bogotá, Colombia
j.rayom@uniandes.edu.co

Raúl de La Rosa
Universidad de los Andes
Bogotá, Colombia
c.delarosap@uniandes.edu.co

Mario Garrido
Universidad de los Andes
Bogotá, Colombia
m.garrido10@uniandes.edu.co

Abstract

Regulatory texts are inherently long and complex, presenting significant challenges for information retrieval systems in supporting regulatory officers with compliance tasks. This paper introduces a hybrid information retrieval system that combines lexical and semantic search techniques to extract relevant information from large regulatory corpora. The system integrates a fine-tuned sentence transformer model with the traditional BM25 algorithm to achieve both semantic precision and lexical coverage. To generate accurate and comprehensive responses, retrieved passages are synthesized using *Large Language Models* (LLMs) within a *Retrieval Augmented Generation* (RAG) framework. Experimental results demonstrate that the hybrid system significantly outperforms standalone lexical and semantic approaches, with notable improvements in Recall@10 and MAP@10. By openly sharing our fine-tuned model and methodology, we aim to advance the development of robust natural language processing tools for compliance-driven applications in regulatory domains.

1 Introduction

Information retrieval (IR) systems are concerned with efficiently querying large corpora to retrieve relevant results. Traditional systems, such as search engines, often depend on term-frequency statistical methods like *tf-idf*, which measures the importance of a term in a document relative to its frequency in the corpus (Melucci and Baeza-Yates, 2011). BM25 (Robertson et al., 1996), a well-established ranking function, builds on similar principles to provide a scalable and effective retrieval framework. However, such methods are inherently limited when addressing complex domains like regulatory texts, where the semantics often outweigh simple term matching.

Regulatory content is particularly challenging due to its specialized terminology and nuanced language. Synonyms, paraphrasing, and domain-specific jargon frequently obscure the relationship between queries and relevant documents, reducing the effectiveness of lexical retrieval methods.

Semantic search addresses these limitations by using dense vector-based retrieval where we encode documents and queries as vectors, also known as *embeddings*, capturing the semantic meaning of the text in a condensed high-dimensional space (Karpukhin et al., 2020). This approach enables the system to measure similarity based on meaning rather than exact word matches, grouping related content together even with different terminology.

Recent advances in pre-trained language models, like BERT (Devlin et al., 2018), have introduced high-quality contextual *embeddings* for words, sentences, and paragraphs which can be leveraged in semantic search applications.

Despite these advances, building an effective IR system for regulatory texts poses unique challenges. Pre-trained language models are typically trained on general-purpose datasets and may lack the domain-specific knowledge required for accurate retrieval in specialized fields. Fortunately, various methods for transfer learning have demonstrated that these base models can be fine-tuned to close this gap (Houlsby et al., 2019).

In this paper, we present a hybrid information retrieval system that integrates both lexical and semantic approaches to address the limitations of traditional IR in the regulatory domain. Our method combines BM25 for lexical retrieval with a fine-tuned Sentence Transformer model (Reimers and Gurevych, 2019) to improve semantic matching. Additionally, we implement a Retrieval Augmented Generation (RAG) system (Lewis et al., 2021) that leverages the hybrid retriever to provide comprehensive and accurate answers to user queries using a Large Language Model (LLM).

Through extensive experiments, we demonstrate that the hybrid retriever achieves superior perfor-

Proceedings of the 31st International Conference on Computational Linguistics, pages 31–35
January 19–24, 2025. ©COLING 2025

mance compared to standalone lexical or semantic systems, as evidenced by improvements in Recall@10 and MAP@10. Furthermore, the RAG system effectively synthesizes retrieved content, delivering detailed responses that address the compliance requirements of regulatory questions. Our contributions aim to advance regulatory information retrieval and lay the foundation for more effective question-answering systems in specialized domains.

2 Regulatory Information Retrieval

The development of an effective information retrieval (IR) system for regulatory content requires addressing the unique challenges of compliance-related queries. These systems must return a set of ranked passages from the corpus that accurately address the compliance aspects of a given question. Previous work by Gokhan et al. (2024) utilized BM25, a widely-used algorithm that ranks results based on query term frequency and other statistical features. While BM25 is effective for lexical retrieval, it struggles to capture semantic relationships, particularly in regulatory domains where terminology often varies for the same concepts.

Our approach enhances BM25 by integrating a text embedding model, enabling semantic matching. This hybrid system identifies semantically relevant content that BM25 alone might overlook, offering a significant advantage in handling the complexities of regulatory language.

2.1 Dataset

The dataset used for this study, ***ObliQA***, consists of 27,869 regulatory questions extracted from 40 documents provided by Abu Dhabi Global Markets. This regulatory authority oversees financial services within the European Economic Area, making the dataset highly relevant for compliance-related tasks (Gokhan et al., 2024).

The dataset is divided into three subsets: training (22,295 questions), testing (2,786 questions), and validation (2,788 questions). Each question is paired with one or more passages that contain the relevant information needed to answer it. The data is stored in JSON format, where each entry includes the question, associated passages, and their metadata. An example is shown below.

```
{
  "QuestionID": "a10724b5-ad0e-4b69-8b5e-792aef214f86",
  "Question": "What are the two specific conditions related to the maturity of a financial instrument that would trigger a disclosure requirement?",
  "Passages": [
    {
      "DocumentID": 11,
      "PassageID": "7.3.4",
      "Passage": "Events that trigger a disclosure. For the purposes of Rules 7.3.2 and 7.3.3, a Person is taken to hold Financial ..."
    }
  ],
  "Group": 1
}
```

2.2 Model Fine-tuning

We fine-tuned the *BAAI/bge-small-en-v1.5* (Xiao et al., 2023), a BERT-based model trained on general-purpose data. The fine-tuning process employed a loss function designed to maximize the similarity between questions and their associated passages. The architecture comprises a word embedding layer followed by pooling and normalization layers. To better capture semantic nuances in regulatory texts, we increased the embedding dimension from 384 to 512.

Training was conducted on an NVIDIA A40 GPU with 24GB of memory using the *SentenceTransformer* library (Reimers and Gurevych, 2019). The model was trained over 10 epochs with a batch size of 64, using a learning rate of $2x10^{-4}$ to preserve the model's general-purpose knowledge while fine-tuning it for the domain. The *MultipleNegativesRankingLoss* (Reimers and Gurevych, 2023) loss function was employed, assuming all unpaired examples in the batch as negatives, which is particularly suited for scenarios with positive pairs only.

Performance evaluation was conducted using the *InformationRetrievalEvaluator* (Reimers and Gurevych, 2021) to compute metrics such as Recall@10, Precision@10, and MAP@10 during training. To further optimize the process, we employed warmup steps to gradually increase the learning rate, and Automatic Mixed Precision (AMP) (Zhao et al., 2021) to reduce memory usage and enhance training speed.

Table 1 summarizes the results, showing a significant performance improvement of the fine-tuned model over the base model in the regulatory domain. The fine-tuned model has been made avail-

Model / Dataset	Recall@10	MAP@10
Base Model / Validation	0.7135	0.5462
Base Model / Testing	0.7017	0.5357
Custom Model / Validation	0.8158	0.6315
Custom Model / Testing	**0.7111**	**0 .6261**

Table 1: Performance comparison between the base model and the fine-tuned model.

able on Hugging Face Hub, alongside the complete implementation in our GitHub repository.

2.3 Information Retrieval

To enhance retrieval performance, we developed a data processing pipeline with the following steps:

1. **Expand contractions**: Convert contractions (e.g., *don't* to *do not*) for consistency.
2. **Normalization**: Lowercase text and remove non-alphanumeric characters using regular expressions.
3. **Space removal**: Eliminate redundant spaces for uniformity.
4. **Preserve legal format**: Retain special characters critical for legal documents.
5. **Stopwords**: Remove common words using *nltk* and *scikit-learn* sets.
6. **Stemming**: Apply the *Snowball Stemmer* (Porter, 2001) to reduce words to their root forms.
7. **Tokenization**: Generate unigrams and bigrams to capture both individual terms and word combinations.

Using this pipeline, we implemented three retrieval approaches:

1. BM25 (Baseline): Configured with $k = 1.5$ and $b = 0.75$.
2. Semantic Retriever: Leveraged the fine-tuned model for semantic matches only.
3. Hybrid System: Combined BM25 and the fine-tuned model, computing an aggregated score using Equation 1:

$$\begin{aligned} \text{Score} = {} & \alpha \cdot \text{Semantic Score} \\ & + (1 - \alpha) \cdot \text{Lexical Score} \end{aligned} \tag{1}$$

Model	Recall@10	MAP@10	Recall@20	MAP@20
BM25 (Baseline)	0.7611	0.6237	0.8022	0.6274
BM25 (Custom)	0.7791	0.6415	0.8204	0.6453
Semantic system	0.8103	0.6286	0.8622	0.6334
Hybrid system	**0.8333**	**0.7016**	**0.8704**	**0.7053**

Table 2: Performance comparison between information retrieval systems.

We empirically set $\alpha = 0.65$ to give slightly higher weight to semantic matching while maintaining meaningful contribution from lexical search. This normalization step ensures that neither approach dominates the final ranking purely due to differences in score distributions.

Table 2 compares the performance of these approaches. The hybrid system demonstrates the highest effectiveness, combining the strengths of lexical and semantic retrieval methods.

3 Answer Generation

Retrieval Augmented Generation (RAG) is a cutting-edge technique that enhances *Large Language Models* (LLMs) by integrating external retrieval capabilities, enabling them to generate responses based on information they were not explicitly trained on (Lewis et al., 2021). This approach has emerged as a powerful tool in open-domain question-answering applications, combining retrieval-based and generation-based methods to improve answer relevance and quality (Siriwardhana et al., 2023).

In our system, RAG is used to answer regulatory questions by leveraging the hybrid information retrieval system described earlier. The retrieved passages provide the contextual foundation for generating answers that address compliance-related aspects comprehensively and accurately.

Given a regulatory question, similar to the approach followed in (Gokhan et al., 2024), the system retrieves up to 10 relevant passages from the corpus. To ensure high-quality input for the answer generation process, only passages with a relevance score of at least 0.72 are considered. Additionally, passage processing is terminated when the relevance score drops by more than 0.1 from the previous passage, maintaining the relevance and coherence of the input data.

These selected passages are fed into an LLM to synthesize a concise and coherent answer. For this task, we experimented with three different models: *GPT 3.5 Turbo* and *GPT-4o Mini* through Azure OpenAI batch deployment, and *Llama 3.1* using Groq's API. When evaluated on our test dataset, *GPT 3.5 Turbo* achieved the highest RePASs score of 0.57, significantly outperforming both *GPT-4o Mini* (0.44) and *Llama 3.1* (0.37), leading to its selection as our primary model. We designed the system prompt to guide response generation in the regulatory domain, emphasizing accuracy, completeness, and alignment with the provided passages. The prompt reads:

> "*As a regulatory compliance assistant. Provide a **complete**, **coherent**, and **correct** response to the given question by synthesizing the information from the provided passages. Your answer should **fully integrate all relevant obligations, practices, and insights**, and directly address the question. The passages are presented in order of relevance, so **prioritize the information accordingly** and ensure consistency in your response, avoiding any contradictions. Additionally, reference **specific regulations and key compliance requirements** outlined in the regulatory content to support your answer. **Do not use any extraneous or external knowledge** outside of the provided passages when crafting your response.*"

We selected the top 3 answers with the highest RePASs scores to enhance the prompt using few-shot techniques, aiming to improve its performance. Below is a demonstration of how we used this prompting method.

> " *Question: What percentage of the Insurer's Net Written Premium is used to determine the non-proportional reinsurance element? Passage: The non proportional reinsurance element is calculated as of the Insurer's Net Written Premium Your response should read: The non-proportional reinsurance element is determined by calculating 52 percent of the Insurer's Net Written Premium.*"

Regulatory Passage Answer Stability Score (RePASs), introduced by Gokhan et al. (2024) assesses the stability and accuracy of generated answers across three key dimensions:

1. Entailment Score (E_s): Measures the extent to which each sentence in the generated answer is supported by sentences in the retrieved passages.

2. Contradiction Score (C_s): Evaluates whether any sentence in the generated answer contradicts the information in the retrieved passages.

3. Obligation Coverage Score (OC_s): Checks if the generated answer covers all obligations present in the retrieved passages.

System	Es	Cs	OCs	RePASs
Baseline	0.78	0.24	0.20	0.58
Hybrid retriever + GPT-4o Mini	0.38	0.23	0.17	0.44
Hybrid retriever + Llama 3.1	0.34	0.45	0.22	0.37
Hybrid retriever + GPT 3.5 Turbo	**0.58**	**0.21**	**0.33**	**0.57**

Table 3: Performance comparison of answer generation systems using RePASs metrics.

The composite RePASs score is derived from these metrics, offering a holistic measure of the system's answer quality. Table 3 summarizes the evaluation results, comparing our approach to the baseline.
Table 3 shows that while our system achieves moderate improvements in obligation coverage (OC_s) and slightly better contradiction handling (C_s), its entailment score (E_s) reveals areas for further optimization. The hybrid retrieval system enhances answer relevance by incorporating semantic and lexical matches, but the synthesis process using *GPT 3.5 Turbo* shows reduced performance in capturing the degree to which generated answers are supported by the retrieved passages, as evidenced by the lower entailment score.

4 Conclusion

This work tackles the significant challenges of retrieving and synthesizing information from complex regulatory texts by demonstrating the effectiveness of hybrid approaches that integrate lexical and semantic retrieval methods. Our results show the importance of combining classical algorithms, such as BM25, with embedding-based models to address the nuanced language and diverse terminologies inherent in regulatory domains. The hybrid system consistently outperforms standalone lexical or semantic approaches, achieving notable improvements in metrics like Recall@10 and MAP@10.
We further demonstrate the potential of LLMs to synthesize concise and comprehensive answers. These models effectively utilize the structured information retrieved by the hybrid system to address regulatory queries with improved coherence and relevance. However, the evaluation using RePASs reveals opportunities for refinement, particularly in improving entailment metrics.
Future directions include fine-tuning LLMs on domain-specific corpora to enhance alignment with regulatory contexts, optimizing retrieval thresholds for better semantic coverage, and exploring advanced scoring mechanisms to balance precision and recall.

Acknowledgments

This work was supported by the NLP Group at Universidad de los Andes. We thank Abu Dhabi Global Markets for providing access to their regulatory documents. Special thanks to our dedicated professor Rubén Francisco Manrique.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

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.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. *Preprint*, arXiv:1902.00751.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense passage retrieval for open-domain question answering. *Preprint*, arXiv:2004.04906.

Patrick 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. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Massimo. Melucci and Ricardo. Baeza-Yates. 2011. *Advanced Topics in Information Retrieval*, 1st ed. 2011. edition. The Information Retrieval Series, 33. Springer Berlin Heidelberg, Berlin, Heidelberg.

Martin F. Porter. 2001. Snowball: A language for stemming algorithms.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2021. Information retrieval evaluator. `https://sbert.net/docs/package_reference/evaluation.html#sentence_transformers.evaluation.InformationRetrievalEvaluator`.

Nils Reimers and Iryna Gurevych. 2023. Sentence transformers documentation: Losses. `https://sbert.net/docs/package_reference/losses.html#multiplenegativessymmetricrankingloss`.

Stephen E Robertson, Steve Walker, MM Beaulieu, Mike Gatford, and Alison Payne. 1996. Okapi at trec-4. *Nist Special Publication Sp*, pages 73–96.

S. Siriwardhana, R. Weerasekera, E. Wen, T. Kaluarachchi, R. Rana, and S. Nanayakkara. 2023. Improving the domain adaptation of retrieval augmented generation (rag) models for open domain question answering. *Transactions of the Association for Computational Linguistics*, 11:1–17.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

C. Zhao, Ting Hua, Y. Shen, L. Qian, and H. Jin. 2021. Automatic mixed-precision quantization search of bert. *Preprint*, arXiv:2112.14938.

1-800-SHARED-TASKS at RegNLP: Lexical Reranking of Semantic Retrieval (LeSeR) for Regulatory Question Answering

Jebish Purbey
Pulchowk Campus, IoE
jebishpurbey@gmail.com

Drishti Sharma
Cohere For AI Community
drishtishrma@gmail.com

Siddhant Gupta *
IIT Roorkee
siddhant_g@me.iitr.ac.in

Khawaja Murad *
NUST, Pakistan
khawajamurad@outlook.com

Siddartha Pullakhandam
University of Wisconsin
pullakh2@uwm.edu

Ram Mohan Rao Kadiyala
University of Maryland
rkadiyal@umd.edu

Abstract

This paper presents the system description of our entry for the COLING 2025 RegNLP RIRAG (Regulatory Information Retrieval and Answer Generation) challenge, focusing on leveraging advanced information retrieval and answer generation techniques in regulatory domains. We experimented with a combination of embedding models, including Stella, BGE, CDE, and Mpnet, and leveraged fine-tuning and reranking for retrieving relevant documents in top ranks. We utilized a novel approach, LeSeR, which achieved competitive results with a recall@10 of 0.8201 and map@10 of 0.6655 for retrievals. This work highlights the transformative potential of natural language processing techniques in regulatory applications, offering insights into their capabilities for implementing a retrieval augmented generation system while identifying areas for future improvement in robustness and domain adaptation.

1 Introduction

Regulatory documents pose significant challenges for organizations seeking to ensure compliance owing to their complexity and ever-changing nature. It is important for organizations to adhere to regulations to maintain legal compliance. With the recent advances in Natural Language Processing (NLP), there is an opportunity to tackle these issues and automate the process of information retrieval, regulatory comparisons, and compliance verifications. Regulatory NLP (RegNLP) focuses on improving access to and understanding of regulatory rules and obligations by leveraging NLP techniques. Within RegNLP, usage of language models for the retrieval of regulatory guidelines for Question Answering (Q/A) has shown great potential (Abualhaija et al., 2022).

In light of this, this paper focuses on our submission to the COLING 2025 Regulatory Information Retrieval and Answer Generation (RIRAG) challenge, involving two key tasks: retrieving top-k relevant passages for the given set of queries and using the relevant passages to formulate answers with language models. Our approach enhances the capabilities of semantic retrievals for RIRAG by fine-tuning an embedding model on positive data pairs and reranking it using lexical retrieval techniques.

LeSeR (Lexical reranking of Semantic Retrieval) is a novel hybrid approach that combines dense semantic retrieval with classical lexical reranking for enhanced retrieval performance. It leverages dense embeddings fine-tuned on query-passage pairs and integrates BM25 (Robertson et al., 1994) scores to improve ranking precision. This dual approach enables robust retrieval in complex regulatory domains, outperforming both pure lexical and semantic models. We test a multitude of open-source models and select the best model for LeSeR. Our work contributes to developing specialized retrieval systems for Q/A in regulatory domains.

2 Dataset & Task

The RIRAG task aims to enhance the efficiency and accuracy of compliance-related tasks by addressing two critical subtasks in RegNLP: passage retrieval (Sub-task 1) and answer generation (Sub-task 2). The first Sub-task (1) is to identify and extract the most relevant passages, specifically obligations and related rules, from ADGM regulations and guidance documents. Building on this, the second Sub-task (2) focuses on the ability to gen-

* equal contribution

Proceedings of the 31st International Conference on Computational Linguistics, pages 36–40
January 19–24, 2025.

Split	Questions	1 Passage	2 Passages	3 Passages	4 Passages	5 Passages	6 Passages
Train	22,295	16,946	4,016	975	202	100	56
Test	2,786	2,126	506	105	36	9	4
Dev	2,888	2,215	514	116	30	12	1

Table 1: Distribution of passages per question across train, test, and development splits

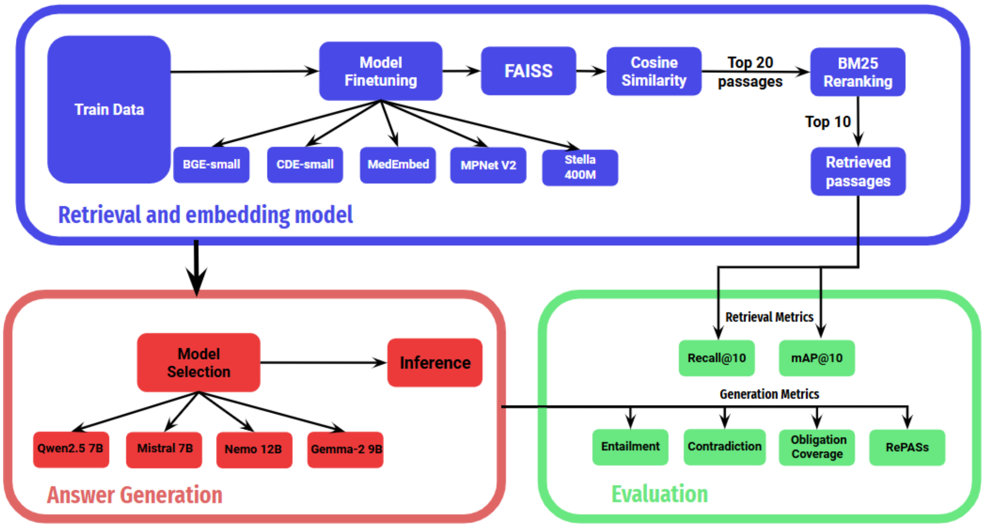

Figure 1: System design workflow

erate clear and concise information from varying sources to fully address the compliance and obligation requirements of the query. RIRAG utilizes the ObliQA Dataset (Gokhan et al., 2024) which is a RegNLP resource built from the Abu Dhabi Global Markets (ADGM) regulations. The dataset incorporates comprehensive and meticulously organized documents, preserving the intricate structure and terminology characteristics of legal and regulatory texts. The dataset includes 22,295 training, 2,888 development, and 2,786 test examples. Each instance consists of a natural language question, relevant regulatory passages annotated with DocumentID and PassageID, and contextual group identifiers. ObliQA is a multi-retrieval dataset and its distribution is given in Table 1.

3 Methodology

For the passage retrieval task, our approach leverages a combination of dense and sparse retrieval methods to maximize the relevance and diversity of the retrieved passages as shown in Figure 1. Hybrid retrieval approaches combine the strengths of semantic and lexical techniques to enhance retrieval quality. In these methods, semantic embeddings are often used for relevance matching, while lexical models ensure precision by addressing nuances like exact term matching and linguistic variation. Such approaches generally merge semantic and lexical scores during retrieval, rather than isolating the two stages. We propose LeSeR (Lexical-Semantic Retrieval), a novel take on hybrid retrieval that uniquely decouples these phases. Semantic embeddings retrieve high-recall candidates, which are then reranked lexically for precision. This strict modularity ensures optimal performance tailored to the challenges of regulatory information retrieval.
We utilize a dense vector-based search mechanism using the FAISS library (Douze et al., 2024), with embeddings generated by fine-tuning an embedding model. A total of 20 top-ranked passages are then retrieved based on cosine similarity scores. To enhance retrieval performance further, we integrate BM25, a classical sparse lexical retrieval method, as a reranking tool. Passages retrieved using the fine-tuned embedding model are re-ranked by combining their dense semantic scores with sparse relevance scores generated by BM25 using a weighted aggregation approach, and the top-10 results are passed as context for answer generation.
The embedding model was fine-tuned on a dataset derived from ObliQA for a maximum of 10 epochs,

employing a batch size of 64 and a learning rate of 2×10^{-5}. The dataset consists of anchor-positive pairs. We used Multiple Negative Symmetric Ranking Loss (MNSR) for contrastive learning, which treats every in-batch example as a potential negative example for all other queries, maximizing efficiency during training. The "symmetric" aspect means it considers bidirectional relationships (query-to-passage and passage-to-query) to improve the alignment of representations. The dev dataset was used for creating the evaluation dataset for fine-tuning, in order to load the best checkpoint at the end of the training. The model fine-tuned under this approach includes BGE-small-en-v1.5 (Xiao et al., 2023), Contextual Document Embeddings (CDE) Small (Morris and Rush, 2024), MedEmbed (Balachandran, 2024), MPNet V2 (Song et al., 2020), and Stella 400M English (Zhang, 2024). The best model is used for retrieving relevant passages using the LeSeR approach.
For the answer generation task, we test four open-source models, namely Qwen2.5 7B (Qwen Team, 2024), Mistral 7B (Jiang et al., 2023), Mistral Nemo 12B (MistralAI, 2024), and Gemma-2 9B (GemmaTeam, 2024). The prompts for answer generation models are designed to incorporate the retrieved passages in Sub-task 1 as contexts and the inference is done using batch size of 1. For faster inference, we use Unsloth's FastLanguageModel (UnslothAI, 2024) for 2x inference performance.
For assessing the performance of answer generation, we use RePASs metric (Gokhan et al., 2024) which measures the overall quality of answer generation using query, retrieved passages, and answer, based on Entailment and Contradictions scores.
For assessing the performance of retrievals, we used Recall@10, which measures the proportion of relevant passages retrieved within the top-10 results and mean Average Precision@10 (mAP@10), which evaluates the precision of ranked passages.

4 Results

During the fine-tuning phase, various retrieval models were assessed on the test dataset to identify the top-performing systems for the retrieval task (Table 2). The baseline BM25 model achieved a Recall@10 of 0.7611 and mAP@10 of 0.6237, setting a strong benchmark for comparison. Among the other models, Stella achieved a Recall@10 of 0.7756 and mAP@10 of 0.1036, demonstrating its strong retrieval performance, but poor ranking performance. Additionally, BGE reached a Recall@10 of 0.7040 and mAP@10 of 0.0960. The dense search models performed very poorly in terms of average precision compared to the baseline lexical model, suggesting that exact keyword matching might be more appropriate for the tasks.

Model	**Recall@10**	**mAP@10**
BM25 (baseline)	0.7611	0.6237
MPNet	0.6897	0.0949
CDE	0.1012	0.0232
MedEmbed	0.6830	0.0938
Stella	0.7756	0.1036
BGE	0.7040	0.0960
MPNet_MNSR	0.7977	0.1081
CDE_MNSR	0.7030	0.1029
MedEmbed_MNSR	0.8049	0.1108
Stella_MNSR	0.7973	0.1089
BGE_MNSR	0.8068	0.1077
BGE_LeSeR	**0.8201**	**0.6655**

Table 2: Results of the retrieval task on the test dataset. Models with '_MNSR' represent fine-tuned versions of the model and '_LeSeR' represents retrieval with the LeSeR approach.

Fine-tuning the dense model improved their performance in recall significantly. BGE_MNSR (fine-tuned with MNSR loss) performed the best with Recall@10 of 0.8068, outperforming the baselines model. MedEmbed model, which itself is a fine-tuned version of BGE, performed similarly to BGE with recall@10 of 0.8049. However, the semantic retrieval models still lagged behind baseline BM25 in terms of mAP@10 massively, with MedEmbed having the best mAP@10 of 0.1108, compared to baseline mAP@10 of 0.6237. Because of its highest recall score, BGE_MNSR is implemented in the LeSeR approach. Its performance improved massively compared to its previous counterparts. With recall@10 of 0.8201 and mAP@10 of 0.6655, it outperforms all other models, including the baseline model. This shows the effectiveness of the LeSeR approach in regulatory retrieval systems.
For assessing the performance of the answer generation task, we use the answers generated by the model for unseen questions, giving an account of the real-world performance of the system. The performance of models integrating the BGE_LeSeR retrieval system with various large language models (LLMs) was assessed using the RePASs metric and is shown in Table 3. Among the models tested, Qwen2.5 7B outperformed the others

Method	E	C	OC	RePASs
BGE_LeSeR + Mistral 7B	0.5229	0.5408	0.0329	0.3383
BGE_LeSeR + Nemo 12B	0.4283	0.4804	0.0353	0.3277
BGE_LeSeR + Gemma-2 9B	0.5407	**0.3262**	0.0678	0.4274
BGE_LeSeR + Qwen2.5 7B	**0.5730**	0.3480	**0.0772**	**0.4340**

Table 3: Results of answer generation task using RePASs on the unseen questions set. E, C, OC, and RePASs represent Entailment, Contradiction, Obligation Coverage, and RePAS scores, respectively.

across all metrics, achieving the highest score for Entailment (0.5730), second lowest Contradiction score (0.3480), highest Obligation Coverage (0.0772), and highest RePASs (0.4340). These results demonstrate Qwen2.5's effectiveness in generating high-quality answers, making it the top performer in this evaluation. Gemma-2 9B came close to the performance of the Qwen model with a RePASs score of 0.4274 and had the lowest Contradiction score of 0.3262. Mistral 7B, and Nemo 12B showed comparatively lower performance across the board, with Qwen2.5 consistently outperforming them.

5 Conclusion

Our results highlight the significant impact of leveraging hybrid approaches to improve performance in complex retrieval and answer generation tasks. The BGE_LeSeR when paired with Qwen2.5 7B, demonstrated superior performance in both recall and answer quality, outperforming other models such as Mistral 7B, Nemo 12B, and Gemma-2 9B across multiple metrics. Our LeSeR approach demonstrated significant improvements in both recall and precision of retrievals. This progression from traditional retrieval models to advanced LLM-based fine-tuning with reranking illustrates the importance of iterative adaptation, allowing models to specialize in retrieving relevant information while simultaneously enhancing their ability to generate coherent, contextually relevant answers.

The superior performance of Qwen2.5, particularly in the RePASs evaluation, underscores the potential of integrating fine-tuned retrieval systems with high-performing generative models to address nuanced tasks such as answer synthesis. This work emphasizes the importance of combining robust retrieval mechanisms with effective answer-generation strategies to create AI systems capable of delivering high-quality, actionable insights. Integration of sophisticated embeddings and large-scale language models within the LeSeR framework demonstrates the transformative potential in improving compliance monitoring and regulatory interpretation workflows. Moving forward, future research could explore advanced fine-tuning techniques, ensemble models, newer reranking mechanisms, and domain-specific adaptations to further enhance the scalability and interpretability of these systems in regulatory domains.

Limitations

The proposed framework, while demonstrating significant advantages, has certain limitations that should be considered. First, while dense retrieval models such as BGE_MNSR showed substantial improvements in recall after fine-tuning, they underperformed in ranking precision, as evidenced by lower mAP@10 scores compared to the baseline BM25 model. This indicates a challenge in effectively prioritizing the most relevant passages, which is critical for practical applications requiring precise rankings. Second, dense retrieval models exhibited limitations in capturing fine-grained semantic nuances compared to lexical-based models like BM25. This shortfall may stem from the complex and diverse terminology characteristic of regulatory texts, where exact keyword matches often play a critical role. Finally, metrics such as Recall@10 and mAP@10 evaluate different aspects of retrieval performance. Recall@10 emphasizes the breadth of retrieval but does not reflect the relevance or ranking order of retrieved passages as effectively as mAP@10. The divergence in these metrics underscores the trade-offs between recall-oriented and precision-oriented evaluations, complicating the interpretation of model effectiveness. Future research should explore hybrid retrieval methods, optimize semantic understanding in dense models, and refine evaluation metrics to balance recall and precision more effectively. The answer generation could also be improved further by appending only relevant contexts in the input prompt, instead of top-10 or top-20 retrievals.

References

Sallam Abualhaija, Chetan Arora, Amin Sleimi, and Lionel C. Briand. 2022. Automated question answering for improved understanding of compliance requirements: A multi-document study. In *2022 IEEE 30th International Requirements Engineering Conference (RE)*, pages 39–50.

Abhinand Balachandran. 2024. Medembed: Medical-focused embedding models.

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

GemmaTeam. 2024. Gemma: Open models based on gemini research and technology. *Preprint*, arXiv:2403.08295.

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.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

MistralAI. 2024. Mistral nemo.

John X. Morris and Alexander M. Rush. 2024. Contextual document embeddings. *Preprint*, arXiv:2410.02525.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Stephen Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at trec-3. In *3rd Text REtrieval Conference (TREC-3)*, page 109–126.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Preprint*, arXiv:2004.09297.

UnslothAI. 2024. Fastlanguage: A fast nlp model. Accessed: 2024-12-02.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

Dun Zhang. 2024. Stella: A multilingual text embedding model. Accessed: 2024-11-25.

MST-R: Multi-Stage Tuning for Retrieval Systems and Metric Evaluation

Yash Malviya*, **Karan Dhingra*** and **Maneesh Singh**
Indic aiDias Team (Independent Researchers)
{yashmalviya98,kdhingra307}@gmail.com
Correspondence: dr.maneesh.singh@ieee.org

Abstract

Regulatory documents are rich in nuanced terminology and specialized semantics. FRAG systems: Frozen retrieval-augmented generators utilizing pre-trained (or, frozen) components face consequent challenges with both retriever and answering performance. We present a system that adapts the retriever performance to the target domain using a multi-stage tuning (MST) strategy. Our retrieval approach, called MST-R (a) first fine-tunes encoders used in vector stores using hard negative mining, (b) then uses a hybrid retriever, combining sparse and dense retrievers using reciprocal rank fusion, and then (c) adapts the cross-attention encoder by fine-tuning only the top-k retrieved results. We benchmark the system performance on the dataset released for the RIRAG challenge (as part of the RegNLP workshop at COLING 2025). We achieve significant performance gains obtaining a top rank on the RegNLP challenge leaderboard. We also show that a trivial answering approach *games* the RePASs metric outscoring all baselines and a pre-trained Llama model. Analyzing this anomaly, we present important takeaways for future research. We also release our code base[1].

1 Introduction

Automated Q&A systems hold tremendous potential in not only improving access to, and comprehension of regulatory obligations, but also help organizations achieve regulatory compliance with reduced costs and latency. Currently, compliance workflows are largely manual and organizations need to employ a large number of costly subject matter experts. High recall is especially critical in this domain, as the cost of false negatives i.e. missing crucial regulatory information can lead to severe financial penalties, legal repercussions, and reputational harm. Retrieval-augmented generation (RAG) offers a promising solution but their performance falls short when FRAGs (frozen RAGs or RAGs with pre-trained (frozen) components) are directly applied since regulatory documents utilize specialized, domain-specific terminology and nuanced legal semantics. Domain-specific adaptations are needed to make these systems viable.

This paper primarily focuses on the retriever part of the system, presenting a simple domain adaptation approach to significantly improve the performance of the retriever part of the system by fine-tuning on the target domain. Our contributions include: (a) MST-R A multi-stage retrieval system domain adapted using a multi-stage fine-tuning approach. (b) State of the art retrieval performance on the RIRAG challenge with a improvement of 12.1% in $Recall@10$ and 23% in $MAP@10$ compared to the BGE baseline from Gokhan et al. 2024. (c) Analysis of the RePASs metric with a solution that *games*, with important takeaways.

2 Prior Work

Passage retrieval is a critical step in RAG systems. Early methods relied on sparse representations such as TF-IDF and BM25 (Robertson et al., 2009). Dense encoders(Li et al., 2024; Wang et al., 2022) using late interaction(Khattab and Zaharia, 2020), enable document embedding caching. In contrast, approaches such as (Reimers and Gurevych, 2021; Izacard and Grave, 2021; Lu et al., 2022) use query-document interaction for nuanced semantic alignment but incur higher computational costs due to per-sample processing. Recently, hybrid search algorithms (Askari et al., 2023; Cormack et al., 2009) combining lexical patterns with semantic relationships between queries and passages have emerged. Advanced retrieval systems like Re2G (Glass et al., 2022) employ multi-level architectures to optimize performance and efficiency. Inspired by the above, we propose a multi-level architecture which fuses

*These authors contributed equally to this work.
[1]https://github.com/Indic-aiDias/MST-R

Proceedings of the 31st International Conference on Computational Linguistics, pages 41–51
January 19–24, 2025.

the results of a variety of approaches from the literature to leverage their complementary strengths.

The retrieval stage in QA systems is typically evaluated using metrics like Recall@k and Mean Average Precision (MAP@k). For answer quality, metrics such as BLEU and ROUGE focus on n-gram statistics but neglect semantic equivalence in abstractive generation. Semantic-focused metrics (Wang et al., 2020a; Es et al., 2023; Yue et al., 2023; Laban et al., 2022) emphasize alignment and coherence but lack comprehensive coverage, especially in regulation domains. To address this, RePaSs (Gokhan et al., 2024) enforces coverage of all obligations within relevant passages, ensuring both relevance and regulatory compliance.

3 Methodology

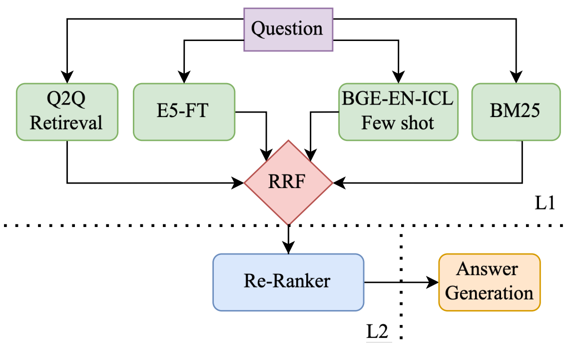

Figure 1: Multi Stage Tuning System Architecture

Our system, depicted in Fig. 1, is a simple, single pass, feed-forward RAG system, similar to Glass et al. 2022. A two-stage retriever selects the relevant passage(s) to provide to a Q&A module. For the latter, we use a pre-trained[2] prompt-tuned LLM.

3.1 Passage Retrieval

The retriever part of our system comprises of two stages (levels). Level 1 (L1) uses the user query to return a ranked list of results. We use a custom hybrid index adapted to the regulatory domain. Level 2 (L2) employs a reranker using a costly cross-attention mechanism to perform a fine-grained analysis of the relevance of the retrieved answer to the query. We adapt the reranker to the target domain by fine-tuning it on the results retrieved from Level 1. This two-level approach allows us to obtain a good trade-off between cost and performance.

[2]Arguably, domain adaptation of the Q&A module will improve the performance of our system further.

3.1.1 Level 1: A Domain-Adapted Hybrid Retriever

Dense retrievers utilize ANN search in semantic spaces where document chunks are embedded using a pre-trained DNN encoder. Dense retrievers have shown excellent performance invariant to the exact formulation of the text as long as the meaning is retained. Fine tuning such retrievers (not training from scratch) is expected to result in partial domain adaptation. To cover for such a shortfall, we also utilize sparse retrievers which use lexical similarity between the question and document chunks. Hence, we design Level 1 to comprise the following retrievers.

(a) **BM25**: is a sparse retriever model leveraging a modified TF-IDF formulation to retrieve passages, and capture lexical similarity between questions and passages. We expect its performance to be co-variant with the domain. (b) **BGE-EN-ICL**: is a dense retriever model allowing for prompt-based few-shot learning. We adapt it to the target domain by conditioning it on five random samples from the ObliQA dataset. (c) **E5-FT**: is a version of the E5 dense retriever model(Wang et al., 2022) created by fine-tuning on the ObliQA dataset, thus adapting it to the target domain. We used contrastive learning with the triplet loss(Schroff et al., 2015) and performed online hard mining(Shrivastava et al., 2016). We provide more details in the Appendix A.2. (d) **Q2Q**(Lewis et al., 2021): A retriever is a map between the space of queries and the space of relevant passages. A new query is an index in this map. Assuming standard regularity conditions, an ANN can be used to retrieve similar queries previously encountered by the system (e.g. questions in the training data). Ground truth passages corresponding to the most relevant previously seen questions are then used. For encoding, MPNet(Song et al., 2020) model was fine-tuned similar to E5-FT.

To combine the results from the above four retrievers, we used reciprocal rank fusion (RRF) (Cormack et al., 2009). The RRF score $S(k)$ of the kth passage is given in terms of its ranks $R(k,i)$ by the ith ranker, by, $S(k) = \sum_{i=1}^{n} \frac{1}{R(k,i)+\beta}$. Please refer to the Appendix A.1 for an analysis of this design choice.

3.1.2 Level 2: Domain-Adapted Reranker

For re-ranking, the *ms-marco-MiniLM-L-6-v2* model(Reimers and Gurevych, 2021; Wang et al., 2020b), a cross-encoder trained on the MSMARCO dataset (Bajaj et al., 2016) for document re-ranking, is fine-tuned by replacing its task head with a binary classification head for relevance prediction[3]. The probability of belonging to the *relevant* class was utilized as the reranking score.

The fine-tuning dataset is constructed using relevant passages from the training ground truth combined with negative sampling. Hard negative samples are selected from the top-K documents of different $L1$ retrievers that are not relevant in the ground truth. Easy negative samples are generated through random sampling passages from the corpus.

3.2 Answer Generation

Ideally, the answering LLM should also be adapted to the target domain. In this work, we have just used Llama3.1 Instruct 8B(Grattafiori et al., 2024) with the prompt mentioned in (Gokhan et al., 2024) & Appendix A.7. Since both (a) better models can be used, and (b) they can be adapted to the target domain, the performance obtained here should be considered as minimum achievable performance. In addition, we evaluated two other *default* strategies of passing the input directly to the output without using an LLM to formulate an answer[4]. Thus, in total, we evaluated three strategies - (a) **LLM - RegNLP Prompt**(Gokhan et al., 2024) using Llama3.1 Instruct 8B(Grattafiori et al., 2024): Appendix A.7. (b) **Passage Concatenation** (PC): simply concatenates the retrieved passages and provisions them as the answer. (c) **Single line** (SL): removes the sentence terminators from the above, converting the entire answer into a single line answer.

4 Evaluation

We now present the results of the proposed system on the ObliQA dataset (Gokhan et al., 2024). A simple pre-processing step was applied to remove section headers, etc. by filtering out passages with fewer than 10 tokens.

Following the guidelines in the challenge, we used $Recall@10$ and $MAP@10$ to evaluate retrieval performance and the RePASs metric to evaluate the goodness of the generated answer.

RePASs metric is defined using entailment, contradiction, and obligatory coverage scores as follows: $RePASs = \frac{E_s - C_s + OC_s + 1}{3}$. Entailment score, E_s (or Contradiction score, C_s) measures whether an answer sentence is entailed (or contradicted) by a retrieved context sentence. OC_s (Obligation Coverage score) measures the percentage of obligations present in the retrieved context that are covered by the answer.

Algorithm	$Recall@10$	$MAP@10$
MST-R (L1+L2)	**0.8746**	**0.7601**
RRF(L1)	0.832	0.6914
BGE (5 shot)	0.7796	0.6178
BM25	0.7611	0.6236

Table 1: Retrieval performance on the ObliQA dataset. Detailed ablation of L1 is presented in Appendix: A.3.

Retrieval results are presented in Table 1. While the two baselines, BM25 and BGE (with 5 shot) have $Recall@10$ scores of 0.76 and 0.78 respectively, L1 level of our system gives a score of 0.83 - a boost of 6.7% relative to BGE. Incorporating the reranker boosts it to 0.87, an additional 5.12% relative to L1.

Method	(RePASs,E_s,C_s,OC_s)
Llama3.1-Instruct-8B	(0.41, 0.215, 0.091, 0.129)
Single Line	(0.801, 0.715, 0.098, 0.786)
Passage Concat	(**0.947**, **0.986**, **0.076**, **0.932**)

Table 2: Performance of various answer generation strategies on the ObliQA dataset.

Answer generation performance is presented in Table 2. Note that the Llama model gives a fairly low score of 0.41 primarily due to low entailment and low coverage. A better LLM, finetuned on the domain, can arguably improve the performance significantly. This is not yet a part of our study. On the other hand, we noticed something peculiar: the RePASs metric is not a *complete* metric. To demonstrate this, we tried two simple baselines: Passage Concatenation (*PC*) & Single Line.

Astonishingly, *PC* achieves a rather high score of 0.947, 130% relative improvement in the RePASs metric over the Llama model. Even when we convert the entire answer into a single line, we get a rather high performance of 0.801. These results highlight the limitations of the RePASs metric for in evaluating answer quality.

5 Analysis: The RePASs Metric

RIRAG uses standard metrics to evaluate the retrieval performance but defines a novel metric

[3]The choice of task head was motivated by experimental exploration.

[4]Refer to Section 5.

called RePASs to evaluate the answer. RePASs is reference-free, using neither the ground truth answer nor the input question. In this section, we analyze the properties of the metric itself.

5.1 Trivial Optimizers?

The RePASs metric is optimized by an answer with a high entailment score and obligatory coverage and with no contradiction. Arguably, an answer constructed from concatenating passages meets the above criterion as long as its content is not self-contradictory. Even a meaningless concatenation of all retrieved passages into a single line outscores the Llama model (results in Table 2). More detailed analysis is provided in Appendix A.5.

5.2 Reasoning Context: RePASs-N?

The RePaSs metric averages over sentence level entailment and contradiction. Stripping source and passage sentences from the surrounding context makes reasoning hard leading to an erratic metric. We investigate below the behavior of the entailment and contradiction scores from NLI models as a function of larger window sizes.

Since the ObliQA dataset doesn't contain GT answers, we used the CNN/Daily Mail dataset (Nallapati et al., 2016). We provision the source and GT summaries to the Deberta v3 NLI model(He et al., 2021) using the same exhaustive, sliding-window strategy as RePASs but with varying context size N (N+1 sentences). We call this extension to the RePASs metric, RePASs-N.

Results are shown in Figure 2. A reasonable expectation is for the entailment (contradiction) score to increase (decrease) with larger N. Only the 'Large' model is able to adequately reason using the larger context. While more exhaustive experimentation is needed, results with $N = 3$ in (Table 4) in the Appendix shows a relative improvement of 20% in RePASs-3 metric over RePASs (or RePASs-0), with corresponding relative improvements of 68% in entailment and contradiction scores.

5.2.1 Better Metrics Needed?

The existence of trivial optimizers for RePASs-N metrics posits the need for additional metrics which ensure relevancy, accuracy and succinctness of the answer and whether it conforms to good form (style, morphology, etc.). The reader is encouraged to take a look at the good body of research on metrics(Es et al., 2023). An emergent trend is to use 'LLM-as-a-Judge' metric to evaluate answer quality.

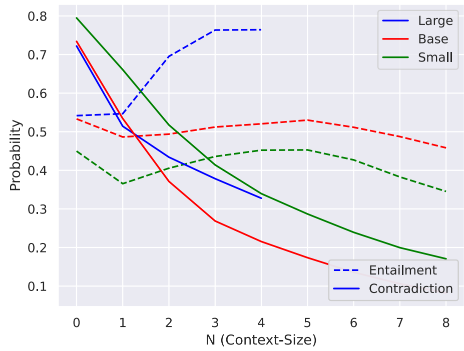

Figure 2: E_s and C_s for RePASs-N. NLI Deberta v3(He et al., 2021) used with context length N.

(Zheng et al., 2023) proposes a reference-free metric requiring the original question and the generated answer.

We evaluated *Passage Concat* and *Llama* answers on this metric. Specifically, we used the prompt in (Roucher) to ask the LLM to judge whether the answer is relevant, direct, detailed, and addresses all the concerns in the question. Prompt details are provided in A.7. Results show that *Passage Concat* has a high RePASs score (0.947) but a lower LLM-as-a-Judge score (2.823). Conversely, answers from Llama have a low RePASs score (0.41), but a high LLM-as-a-Judge score (3.91). The above exploration shows that better metrics are needed to evaluate the goodness of the answer in the regulatory domain: either a single more comprehensive metric, or a list of metrics covering various aspects of answer goodness.

6 Conclusion

The regulatory domain presents significant challenges due to its complex language and contextual requirements needing good domain adaptation strategies. We presented a domain-adapted, multilayered retrieval system showing significant performance gains. While we didn't adapt the performance of the answer generator (leaving it for future work), we present the need for better evaluation metrics as a precursor to engineering better answering models.

Limitations: Considerable research is needed to engineer per-formant answering systems for regulatory domains. While we present an engineered system, it has not been comprehensively evaluated to be deployment-ready. On the other hand, our paper is indicative of the research needed for creating such systems.

References

Arian Askari, Amin Abolghasemi, Gabriella Pasi, Wessel Kraaij, and Suzan Verberne. 2023. Injecting the bm25 score as text improves bert-based re-rankers. *Preprint*, arXiv:2301.09728.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 758–759.

Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2023. Ragas: Automated evaluation of retrieval augmented generation. *arXiv preprint arXiv:2309.15217*.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Rajaram Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2g: Retrieve, rerank, generate. *arXiv preprint arXiv:2207.06300*.

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*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, and Abhinav Pandey et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. *Preprint*, arXiv:2006.03654.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. *Preprint*, arXiv:2007.01282.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. *Preprint*, arXiv:2004.12832.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. Summac: Re-visiting nli-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. Paq: 65 million probably-asked questions and what you can do with them. *Preprint*, arXiv:2102.07033.

Chaofan Li, MingHao Qin, Shitao Xiao, Jianlyu Chen, Kun Luo, Yingxia Shao, Defu Lian, and Zheng Liu. 2024. Making text embedders few-shot learners. *Preprint*, arXiv:2409.15700.

Jing Lu, Keith Hall, Ji Ma, and Jianmo Ni. 2022. Hyrr: Hybrid infused reranking for passage retrieval. *Preprint*, arXiv:2212.10528.

Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *Preprint*, arXiv:1602.06023.

Nils Reimers and Iryna Gurevych. 2021. The curse of dense low-dimensional information retrieval for large index sizes. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 605–611, Online. Association for Computational Linguistics.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2021. Contrastive learning with hard negative samples. *Preprint*, arXiv:2010.04592.

Aymeric Roucher. Using llm-as-a-judge for an automated and versatile evaluation. `https://huggingface.co/learn/cookbook/en/llm_judge#3-improve-the-llm-judge`.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823.

Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. 2016. Training region-based object detectors with online hard example mining. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Preprint*, arXiv:2004.09297.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020a. Asking and answering questions to evaluate the factual consistency of summaries. *Preprint*, arXiv:2004.04228.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020b. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Preprint*, arXiv:2002.10957.

Xiang Yue, Boshi Wang, Ziru Chen, Kai Zhang, Yu Su, and Huan Sun. 2023. Automatic evaluation of attribution by large language models. *Preprint*, arXiv:2305.06311.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

A Appendix

A.1 Reciprocal Rank Fusion

To understand this design choice, note that (a) RRF is dependent on only the ranks of the retrieved results, allowing us to avoid calibrating and fusing distances from different spaces, and (b) RRF can be thought of as a naive fusion model where document relevance, rel, decays exponentially with the reciprocal of its retrieval rank π: $P(rel|\pi) = exp(1/(\pi + \beta))$. Results from K retrievers can be fused using a naive notion of conditional independence: $logP(rel|\{\pi_i\}_{i=1..K}) \approx log\prod_{i=1}^{K} p(rel|\pi_i) = \sum_{i=1}^{K} \frac{1}{\pi_i+\beta}$. In our implementation, we use four retrievers ($K = 4$) and β, a regularizing parameter, is set to 4.

A.2 Domain Adaptation of Dense L1 Retrievers

We used contrastive learning(Robinson et al., 2021) to fine tune the dense retriever to the ObliQA dataset. This involves iteratively fine tuning the encoder with triplet loss(Schroff et al., 2015). At the start of each iteration, top-K passages are retrieved and the distractors are selected from the non-GT retrievals which neighbor GT for smooth gradient optimization. Subsequently, training is run for b batches. At the end of the iteration, the fine tuned model is used to retrieve top-K and the process repeats. We used $k = 10$, $b = 400$ batches and it repeats $n = 200$ times, resulting in $8E+4$ training steps. We used a batch size of 8 samples. This follows standard practice as suggested in (Robinson et al., 2021; Shrivastava et al., 2016).

We tried two retrievers: MPNet and E5. Finetuning E5 improved $Recall@10$ from 0.71 to 0.79 & $MAP@10$ from 0.56 to 0.61. Our eventual system uses E5 as a dense retriever directly and MPNet in the Q2Q retriever module. Using E5 in Q2Q is expected to give better performance but was not evaluated.

A.3 Ablation Study of L1 Retrieval Methods

We performed an exhaustive analysis, trying all possible combinations (15) for four different types of retrievers, each bringing into play a different approach for domain adaptation. The results of this analysis are shared in Table 3.

Note that for singleton retrievers, Q2Q is the weakest at 35%, and the domain adapted E5 is the strongest at 79%. The sparse retriever, BM25, lags behind the dense retrievers by $3\% - 4\%$ in performance. Adding BM25 or BGE improves the performance of E5 by around $1\% - 1.5\%$. Fusing the results of the top three best performing models boosts the performance to 81.89%. Adding the Q2Q model achieves a differential gain of 1.5% giving a final performance of 83.2%, leveraging Q-to-Q coherence for retrieving (memorized) information to similar questions encountered by the system in the past.

A.4 Qualitative Analysis of Retrieval Failure Cases

An ideal retrieval scenario is where there is a unique hit for each query. In other words, the GT retrieval (containing all relevant information to answer the question) is highly correlated with the query while all other information in the knowledge store isn't. While there are several reasons to consider this an unrealistic assumption, it is important to investigate the chunks which are more correlated to the query than the GT chunk and the reasons thereof.

To do this analysis. we identified all non-GT $Top@10$ retrievals where the contradiction score is low ($C_s < 0.2$). Arguably, these chunks should be correlated with the query, with a low contradiction score, and potentially a high entailment score. In Figure 3, we plot a histogram of entailment scores for these chunks. A strong mode at $(0 <= E_s < 0.1)$ is heartening as it shows that NLI can reject non-entailed but highly correlated retrievals. A small peak around 0.5 may indicate model confusion. However, the peak for the bucket-$(0.9 < E_s <= 1)$ is intriguing- it seems to indicate the presence of 'duplicates'in the datasets carrying similar information as the GT. The retriever is correctly retrieving these chunks but is getting wrongly penalized.

Algorithm	$Recall@10$	$Recall@20$	$Recall@40$	$MAP@20$	$MAP@10$	$MAP@40$
BM25,Q2Q,BGE,E5-FT	**0.832**	0.8822	0.9172	0.6963	0.6914	0.6984
Q2Q,BGE,E5-FT	0.8288	0.8822	**0.9198**	0.6539	0.6487	0.6559
BM25,Q2Q,E5-FT	0.8276	0.8766	**0.9173**	0.6689	0.6641	0.6712
BM25,Q2Q,BGE	0.8191	0.8616	0.9041	0.6594	0.6552	0.6616
BM25,BGE,E5-FT	**0.8189**	0.8621	0.8933	0.6935	0.6895	0.6952
BGE,E5-FT	0.8165	0.8655	0.895	0.6654	0.6609	0.6671
BM25,E5-FT	0.8108	0.8576	0.889	0.6832	0.6789	0.6849
Q2Q,E5-FT	0.7968	0.8654	0.9158	0.5263	0.5199	0.5289
BM25,BGE	0.797	0.8341	0.8653	0.6629	0.6593	0.6644
Q2Q,BGE	0.7873	0.8562	0.9089	0.5337	0.5274	0.5364
BM25,Q2Q	0.7798	0.8422	0.9005	0.555	0.5493	0.5579
BGE	0.7796	0.8228	0.8564	0.6215	0.6178	0.6233
E5-FT	0.7926	0.8446	0.8845	0.6196	0.6148	0.6217
BM25	0.7611	0.8022	0.8348	0.6272	0.6236	0.6288
Q2Q	0.3539	0.5896	0.787	0.1904	0.1729	0.1993

Table 3: Retriever performance for different combinations of L1 retrievers. For more than one retriever, we use RRF to fuse the retrieval results. The table is sorted for $Recall@10$.

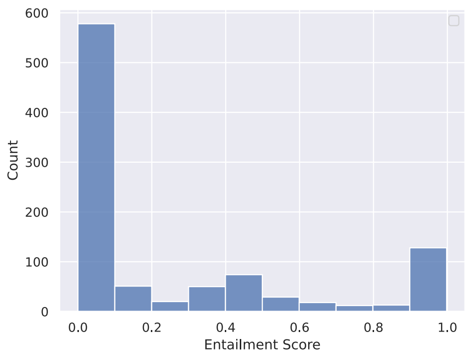

Figure 3: Distribution of entailment scores, E_s (using NLI Deberta v3 Large), for $Top@10$ non-GT retrievals.

We show five examples below which range from having almost duplicate wordings between the GT passage and another $Top@10$ retrieval, to having significant overlap or sharing of phrases/ keywords.

- Table 5 shows passages from **same** document which are near duplicate.
- Table 6 shows passages from **different** document which are near duplicate.
- Tables 7, 8, 9 show passages from **same** document with significant overlap.

These examples show that perhaps taking a binary view of retrieval where the retriever is penalized for not retrieving the passage marked as ground truth to be the top-ranked retrieval is perhaps not a proper metric to (a) evaluate, and (b) improve the performance of retrievers.

Algo	RePASs	E_s	C_s	OC_s
RePASs-0	0.41	0.215	0.091	0.129
RePASs-3	0.49	0.362	0.029	0.137

Table 4: Answer generation performance of the Llama model on the ObliQA dataset. Comparison of metrics with different reasoning context sizes (N).

A.5 RePASs Metric and Optimality

The RePASs metric was defined in Gokhan et al. 2024 as

Type	Text
Ground Truth (Part 4.40.(7), 17)	The Regulator may require an Applicant to provide information which the Applicant is required to provide to it under this section in such form, or to verify it in such a way, as the Regulator may direct.
Non-GT Retrieval. (Part 11.Chapter 2.108.(3), 17)	The Regulator may require an Applicant to provide information which the Applicant is required to provide to it under this section in such form, or to verify it in such a way, as the Regulator may direct.

Table 5: Near duplicate passages from the same document.

$$RePASs = \frac{\text{E}_s - \text{C}_s + \text{OC}_s + 1}{3} \quad (1)$$

$$E_s = \frac{1}{N}\sum_{i=1}^{N} \max_j P_{\text{entailment}}(p_j, a_i) \quad (2)$$

$$C_s = \frac{1}{N}\sum_{i=1}^{N} \max_j P_{\text{contradiction}}(p_j, a_i) \quad (3)$$

where N is the number of sentences in the generated answer, $P_{\text{entailment(contradiction)}}(p_j, a_i)$ denotes the probability that the i-th sentence of the answer (a_i) is entailed (contradicted) by the j-th sentence of the retrieved passage (p_j), and $\max_j$ identifies the maximum probability for each answer sentence among all sentences in the retrieved passage.

$$OC_s = \frac{1}{M}\sum_{k=1}^{M} \mathbb{1}\left(\max_l P_{\text{entailment}}(o_k, a_l) > 0.7\right) \quad (4)$$

where M is the number of obligation sentences in the retrieved passage, $P_{\text{entailment}}(o_k, a_l)$ denotes the probability that the k-th obligation sentence from the passage (o_k) is entailed by the l-th sentence in the answer (a_l), and the indicator function $\mathbb{1}$ outputs 1 if the maximum entailment score surpasses 0.7, signifying that the obligation is covered.

We now show that under reasonable simplifying assumptions, the RePASs metric can attain the maximal value for a trivial answering model. The following two assumptions imply that the regulatory corpus is reasonable and doesn't contain material that is self-contradictory.

Assumption 1. *A sentence from a regulatory corpus entails and doesn't contradict itself, i.e.* $P_{entailment}(p_i, p_i) = 1$ *and* $P_{contradiction}(p_i, p_i) = 0$.

Assumption 2. *Given necessary (maybe unknown) contexts, different sentences across a regulatory corpus should not contradict each other across, so* $P_{contradiction}(p_i, p_j|context_i, context_j) = 0$ *and* $P_{contradiction}(p_j, p_i|context_j, context_i) = 0$.

Using the above assumptions, we can make the following claim (subject to the availability of relevant contexts. Strictly speaking, it changes the definition of the RePASs metric but in a way that we keep to its intended use.

Claim 1. **Passage Concat**, *a trivial answering model, which passes through the retrieved passages concatenating them, attains the maximum RePASs score of 1.*

Proof. The Proof follows from the following statements:

- $E_s = 1$ since by Assumption 1, for all answer sentences a_i, $P_{\text{entailment}}(a_i, a_i) = 1$.
- $C_s = 0$ since by Assumption 2, since no two sentences in a regulatory corpus can contradict each other (given the necessary context): $P_{\text{contradiction}}(a_i, a_j|\text{context}_i, \text{context}_j) = 0$.
- The **Passage Concat** model covers all context sentences and by Assumption 1, entails them. Hence, $\text{OC}_s = 1$.

□

Let's now consider what the above means for practical systems.

Assumption 1 requires that (a) the sentences in a regulatory corpus be meaningful and well formatted; and, (b) an NLI model should have the property that when the premise and hypothesis are exactly the same, it gives an entailment score of 1. While there are reasonable expectations, we see that E_s is not 1, though agreeably reasonably high at 0.986, for the **Passage Concat** model in Table 2. The fact that the answer from Llama model gives such a poor entailment score (0.215) points towards (a) issues due to paraphrasing, perhaps mixing facts from different sentences, and (b) the inability of the NLI model to handle such complex situations.

Assumption 2 is expected to be harder to meet for practical systems. While the non-self-contradiction requirement is reasonable, RsPASs metric requires the property to hold for all sentence

Type	Text
Ground Truth (6.6.13. Guidance, 12)	"If a Return is not submitted by the date on which it becomes due & the Person is in breach of a Rule and the Regulator is entitled to take action including & but not limited to & taking steps to withdraw authorisation to conduct Regulated Activities."
Non-GT Retrieval. (2.3.8. Guidance, 13)	If a return is not submitted by the date on which it becomes due, the Person is in breach of a Rule and the Regulator is entitled to take action including, but not limited to, taking steps to withdraw authorisation to conduct Regulated Activities.

Table 6: Near duplicate passages from different documents.

pairs. Further, an NLI model may require an unknown context for an NLI model to make such a deduction. As shown in Table 2, all the answer generation strategies - trivial ones as well as Llama at contradiction scores in the range $0.07 - 0.1$ (without the appropriate context). We show some examples of contradicting sentences across passages in Appendix A.6.

A.6 Self-Contradicting Sentences in ObliQA?

In Table 10, 11, 12, we show three example pairs containing highlighted sentence pairs having contradiction scores (using NLI Deberta v3 Large) of at least 0.5. While these examples may not be particularly illuminating, they do point to the challenges in NLI and the potential complexities in the real world data. In all these cases, the NLI model should've slotted them into the NEI class.

Passage 1 (PART 4.9.1.1.Application.Guidance.1 of Document 6) and Passage 2 (PART 3.7.1.1 of Doc 6) in Table 10 refer to the obligatory requirements for a 'Foreign Fund Manager'. It is unreasonable to deduce that the highlighted sentences present any kind of contradiction.

Similarly in the scenario in Table 11 where the NLI model shows high contradiction between passage 97 of Doc 31 and passage 4) of Doc 31, and

Table 12 showing between passage APP 4.50. of Doc 11 and 9.2.7 of Doc 11. These short phrases/ titles can't be deemed to indicate contradictions, and point to shortcomings of the NLI model.

A.7 Prompts

RegNLP Answer Generation Prompt Quoted from Gokhan et al. 2024

```
You are a regulatory compliance
    assistant. Provide a detailed answer
     for the question that fully
    integrates all the obligations and
    best practices from the given
    passages. Ensure your response is
    cohesive and directly addresses the
    question. Synthesize the information
     from all passages into a single,
    unified answer.
question: {question}
passages: {context}
answer:
```

LLM as a Judge Prompt Quoted from Roucher

```
You will be given a user_question and
    system_answer couple.
Your task is to provide a 'total rating'
     scoring how well the system_answer
    answers the user concerns expressed
    in the user_question.
Give your answer on a scale of 1 to 4,
    where 1 means that the system_answer
     is not helpful at all, and 4 means
    that the system_answer completely
    and helpfully addresses the
    user_question.

Here is the scale you should use to
    build your answer:
1: The system_answer is terrible:
    completely irrelevant to the
    question asked, or very partial
2: The system_answer is mostly not
    helpful: misses some key aspects of
    the question
3: The system_answer is mostly helpful:
    provides support, but still could be
     improved
4: The system_answer is excellent:
    relevant, direct, detailed, and
    addresses all the concerns raised in
     the question

Provide your feedback as follows:

Feedback:::
Evaluation: (your rationale for the
    rating, as a text)
Total rating: (your rating, as a number
    between 1 and 4)

You MUST provide values for 'Evaluation
    :' and 'Total rating:' in your
    answer.

Now here are the question and answer.

Question: {question}
Answer: {answer}

Provide your feedback.
Feedback:::
Evaluation:
```

Type	Text
Ground Truth (2.4.2, 9)	"Recognised Bodies. Unless otherwise stated in these Islamic Finance Rules & a Recognised Body will be entitled to carry on all & or any part & of its business as Islamic Financial Business provided that: (a) it has complied with all other applicable provisions of the Rulebooks issued by the Regulator in relation to the part of its business to be carried on as Islamic Financial Business; and (b) the carrying on of such part of its business as an Islamic Financial Business has been approved by its Shari'a Supervisory Board."
Non-GT Retrieval. (2.4.2. Guidance.(i), 9)	Whether or not all, or any part, of a Recognised Body's business is to be carried on as Islamic Financial Business, that business must be carried out in compliance with all other relevant parts of the Rulebooks issued by the Regulator.

Table 7: Significant Overlap (Example 1)

Type	Text
Ground Truth (D.6., 36)	"Principle 6 – Incorporation of climate-related financial risks into capital and liquidity adequacy processes. Relevant financial firms should incorporate material climate-related financial risks in their internal capital and liquidity adequacy assessment processes."
Non-GT Retrieval. (D.6.2., 36)	Principle 6 – Incorporation of climate-related financial risks into capital and liquidity adequacy processes. As part of their internal capital and liquidity adequacy assessment processes, relevant financial firms should consider climate-related financial risks that may impact their capital and liquidity positions over relevant time horizons (e.g., through their impact on traditional risk categories).

Table 8: Significant Overlap (Example 2)

Type	Text
Ground Truth (1.2.4. Guidance.1., 4)	"The amount of any supplementary fee will reflect the Regulator's reasonable estimate of the additional time & effort and resources & including those of third parties & necessary to address an issue. Matters which may cause the Regulator to require the payment of a supplementary fee could include & for example: a. complex applications by reason of the Applicant's start-up profile & origin & ownership structure or proposed business model; b. cases where it may be necessary to conduct intense supervisory scrutiny of an entity or individual from a risk perspective; c. complex restructurings or changes in an Authorised Person's or Recognised Body's structure or activities; d. waiver or modification requests which are particularly complex or novel & in the opinion of the Regulator; e. novel proposals and applications that cover untested ground or untested areas of the financial services regulatory regime; or f. assessing complex business models & the supervision of which will require the Regulator to incur material additional expenses & such as & but not limited to & businesses which operate in & or rely upon activities performed in jurisdictions with which & in the view of the Regulator & insufficient arrangements for co-operation exist between the Regulator and the relevant Non-ADGM Financial Services Regulator(s) in that jurisdiction(s)."
Non-GT Retrieval. (1.2.4, 4)	"Supplementary fees The Regulator may require a Person to pay to the Regulator a supplementary fee in circumstances where it expects to incur substantial additional costs or expend substantial additional effort in dealing with an application & authorisation & filing or when conducting on-going supervision."

Table 9: Significant Overlap

Type	Text
Passage 1	refer to A Foreign Fund Manager must also comply with the requirements in this Chapter, because it is managing a Domestic Fund.
Passage 2	Subjecting to the Abu Dhabi Global Market jurisdiction. A Foreign Fund Manager to whom this Chapter applies must: (a) be subject to regulation by, or registration with, a Financial Services Regulator in a .....

Table 10: Contradiction Example - 1

Type	Text
Passage 1	INTERACTION OF CHAPTER 12 WITH OTHER RULE DISCLOSURE OBLIGATIONS. Offers, and Admission to FSRA Official List of Securities Considering the circumstances above, and the positioning of the FSRA in relation to these matters, the FSRA suggests that Issuers/Petroleum Reporting Entities (and their advisors) contact the FSRA as early as possible to discuss.
Passage 2	INTRODUCTION. In the context of the obligations and disclosures by Petroleum Reporting Entities, the FSRA operates as the Listing Authority within ADGM and is therefore charged with supervising Petroleum Reporting Entity disclosures under FSMR, MKT and by incorporation in Chapter 12 of MKT, the PRMS.

Table 11: Contradiction Example - 2

Type	Text
Passage 1	Audit committee. A separate section of the annual report should describe the work of the audit committee in discharging its responsibilities. The annual report should also explain to Shareholders how, if the auditor provides non audit services, auditor objectivity and independence is safeguarded. Principle 5 - Shareholder rights and effective dialogue Rule 9.2.7 ....
Passage 2	Principle 5 - Shareholder rights and effective dialogue. The Board must ensure that the rights of Shareholders are properly safeguarded through appropriate measures that enable the Shareholders to exercise their rights effectively, promote effective dialogue with Shareholders and other key stakeholders as appropriate, and prevent any abuse or oppression of minority Shareholders.

Table 12: Contradiction Example - 3

AUEB-Archimedes at RIRAG-2025: Is obligation concatenation really all you need?

Ioannis Chasandras [1], **Odysseas S. Chlapanis**[1,2] **and Ion Androutsopoulos**[1,2]
[1]Department of Informatics, Athens University of Economics and Business, Greece
[2]Archimedes/Athena RC, Greece

Abstract

This paper presents the systems we developed for RIRAG-2025, a shared task that requires answering regulatory questions by retrieving relevant passages. The generated answers are evaluated using RePASs, a reference-free and model-based metric. Our systems use a combination of three retrieval models and a reranker. We show that by exploiting a neural component of RePASs that extracts important sentences ('obligations') from the retrieved passages, we achieve a dubiously high score (0.947), even though the answers are directly extracted from the retrieved passages and are not actually generated answers. We then show that by selecting the answer with the best RePASs among a few generated alternatives and then iteratively refining this answer by reducing contradictions and covering more obligations, we can generate readable, coherent answers that achieve a more plausible and relatively high score (0.639).

1 Introduction

The Regulatory Information Retrieval and Answer Generation (RIRAG)[1] shared task focuses on the development of systems that can effectively retrieve relevant information from regulatory texts to generate accurate answers for obligation-related queries. It is divided into two subtasks: *passage retrieval*, where systems identify the ten most relevant passages from regulatory documents, and *answer generation*, which requires synthesizing comprehensive answers from the retrieved passages.

We participated with three systems and released our code publicly.[2] Each one of them uses a Rank Fusion (Wang et al., 2021) combination of three retrieval models: BM25 (Robertson et al., 1994), and two neural domain-specific retrievers, based on a law- and a finance-specific embedding model, respectively. We also apply a neural reranker to the top-N retrieved passages.

For answer generation, our first system adversarially exploits the evaluation metric of the task, called RePASs, by using one of its neural components. Specifically, we extract important sentences ('obligations') from the retrieved passages and then concatenate these sentences to get an 'answer'. Even though the produced answers may be incoherent and may not answer the question directly, this system achieves a perfect score, much higher than the score of human experts. The second system extends this approach with an LLM that generates an answer (for each question) by iteratively reformulating (as parts of an answer) the extracted obligations of the previous system. This results in more readable answers, but performance deteriorates to RePASs scores below those of the challenge's baseline (Gokhan et al., 2024).

Our third system works by a) generating multiple candidate answers and using RePASs to select the best answer, and b) iteratively refining the selected answer by removing contradictions and adding 'obligation' sentences that increase RePASs. This system performs worse than the adversarial (first) system, but much better than the baseline, and the answers are coherent and readable.

2 Task setup

Dataset: The dataset of the task consists of train, development, and test sets (22k, 2.8k, 2.7k questions respectively). Passages are retrieved from a corpus of 40 regulatory documents from the Abu Dhabi Global Markets (ADGM) collection. The task organizers used a separate hidden test set, with 446 questions, to evaluate the participants.

Evaluation: Passage retrieval is evaluated using recall@10 and MAP@10. Answer generation is evaluated using RePASs, a reference-free metric (Gokhan et al., 2024). To calculate RePASs, *en-*

[1]https://regnlp.github.io/
[2]https://github.com/nlpaueb/verify-refine-repass

Proceedings of the 31st International Conference on Computational Linguistics, pages 52–58
January 19–24, 2025. ©COLING 2025

tailment and *contradiction* scores are obtained by comparing each sentence of the retrieved passages (used as premises) with each sentence of the generated answer (hypothesis) using an NLI model. For each generated sentence (of the answer), the highest probabilities for entailment and contradiction (comparing to retrieved sentences) are selected, and the scores are averaged over all the sentences of the answer. Additionally, *obligation*-sentences are extracted from the retrieved passages using a LegalBERT model (Chalkidis et al., 2020) fine-tuned on a synthetic dataset (Gokhan et al., 2024). For an obligation to be considered *covered* by the generated answer, a sentence of the answer must entail the obligation-sentence with a confidence above a certain threshold, according to another NLI model.

3 Passage retrieval

All three of our systems use the same passage retrieval, which improves upon the baseline retrieval system of the shared task (Gokhan et al., 2024) in three ways: a) we use domain-specific neural retrieval models, b) we extend the Rank Fusion approach (Wang et al., 2021) to include three models instead of two, and c) we use a reranker.

3.1 Retrieval models

We experiment with BM25 (Robertson et al., 1994) and three of the best[3] text embedding models: `text-embedding-3-large` (OL3) from `OpenAI` (Neelakantan et al., 2022), `voyage-law-2` (VL2), and `voyage-finance-2` (VF2) from `Voyage`.[4] The OL3 embedding model is only used for comparison; it is not included in our final systems, because domain-specific embedding models worked better. We also use the `voyage-rerank-2` reranker.

3.2 Rank Fusion

The task combines the financial and legal domains, which motivates using two domain-specific neural retrievers. Also, according to Wang et al. (2021), BM25 should be fused with neural retrievers, because it captures exact term matching better. Hence, we expand Rank Fusion to handle three retrievers instead of two, as follows.

$$f(p) = a\hat{s}_x(p) + b\hat{s}_y(p) + (1-(a+b))\hat{s}_z(p) \quad (1)$$

[3]MTEB-law: `https://huggingface.co/spaces/mteb/leaderboard?task=retrieval&language=law`

[4]`https://docs.voyageai.com/docs/embeddings`

Here p is a retrieved passage, a and b are fusion weights, and $\hat{s}_x(p)$, $\hat{s}_y(p)$, $\hat{s}_z(p)$ are the normalized relevance scores of the three fused retrievers.

3.3 Experimental results for retrieval

We conduct three experiments on the public test set. In Table 1, we compare the scores of the four single retrieval models. We see that the domain-specific voyage-law-2 (VL2) and voyage-finance-2 (VF2) perform better than BM25 and the generic OL3.

Model	**Recall@10**	**MAP@10**
BM25	0.6994	0.5584
OL3	0.7385	0.5736
VL2	0.7705	0.6275
VF2	**0.7895**	**0.6559**

Table 1: Comparison of single retrieval models.

In the second experiment (Table 2), we compare Rank Fusion configurations, again on the public test set. The newly introduced triple Rank Fusion, with BM25, VL2 and VF2, is the best. The values of a, b were selected by trying a few combinations.

Rank Fusion	**a**	**b**	**R@10**	**M@10**
BM25, OL3	0.30	-	78.9	65.0
VL2, VF2	0.40	-	79.4	66.0
BM25, VL2	0.25	-	79.9	66.5
BM25, VF2	0.30	-	80.4	67.6
BM25, VL2, VF2	0.25	0.2	**81.1**	**69.0**

Table 2: Comparison of Rank Fusion configurations.

In the third experiment (Fig. 1), we investigate the effect of reranking the top-N retrieved passages, for different N values, by computing Recall@10 on the public test set. The best value is $N = 50$.

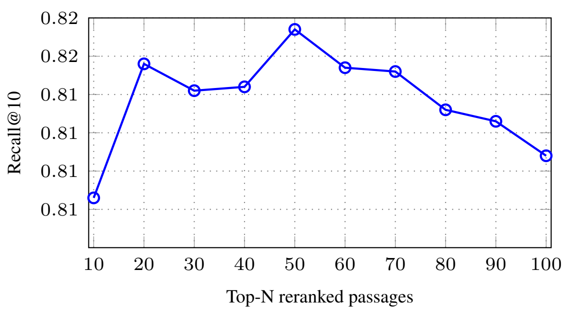

Figure 1: Recall@10 scores of our best retriever (Rank Fusion of BM25, VL2, VF2) when reranking the top-N retrieved passages, for different N values.

Our final retrieval model is a triple Rank Fusion model (BM25, VL2, VF2) with reranking (voyage-rerank-2, $N = 50$), which ranked 4th in the retrieval subtask, achieving 69.4 Recall@10, and 59.4 MAP@10 on the hidden test set.

4 Answer generation

The answer generators of this section use our best retriever (Section 3, BM25, VL2, VF2, reranker).

4.1 Preprocessing

Filtering: We follow Gokhan et al. (2024), i.e., we rank the retrieved passages by decreasing relevance scores; we then keep only passages that satisfy two conditions: (i) their score must be above a certain *threshold*, and (ii) their score must not fall below the previous passage's score more than *max drop*.
Extracting obligations: To obtain obligations from the retrieved passages, we use the same fine-tuned LegalBERT model used in RePASs (Section 2) for obligation extraction. If a passage does not contain any obligations, we use it as is.

4.2 Experimental results for preprocessing

To select the values of the filtering *threshold* and *max drop* (Section 4.1), we conducted two experiments using GPT-4o-mini[5] for answer generation. The first experiment shows that the recommended values of $0.70, 0.20$ of Gokhan et al. (2024) are outperformed by $0.90, 0.10$, respectively (Table 3).

Threshold	Max Drop	RePASs
0.70	0.20	0.4708
0.75	0.05	0.5006
0.80	0.05	0.5050
0.85	0.15	0.5001
0.90	**0.10**	**0.5117**

Table 3: Performance of the baseline answer generator for different values of *threshold* and *max drop*, using our best retriever (BM25, VL2, VF2, reranker).

The second experiment compared the performance of the task's baseline when (a) the entire retrieved passages were given to the LLM, or (b) only the obligations were given, or (c) only the obligations were given, but with a tailored prompt. No significant difference was noticed between (a) and (b), but (c) was significantly better in RePASs (Table 4), due to the increase in *obligation coverage* and *entailment*, even though *contradiction* was worse. All prompts can be found in Appendix B.

[5] https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

Context	RePASs	Obl.	Ent.	Con.
Passages	0.411	0.147	0.177	**0.090**
Obligations	0.413	0.156	0.172	**0.090**
+ prompt	**0.512**	**0.278**	**0.366**	0.109

Table 4: Performance of the baseline system for different kinds of inputs (entire retrieved passages, obligations only, obligations with tailored prompt).

4.3 Naive Obligation Concatenation (NOC)

Our first answer generator (NOC) adversarially exploits the extracted obligations (Section 4.1). It simply concatenates and outputs them as the 'answer'. From the definition of RePASs (Section 2), this answer should get an almost perfect obligation score. Additionally, we expect a low contradiction score, as obligations should not conflict.

4.4 LLM Obligation Concatenation (LOC)

The answers of NOC (Section 4.3) do not answer the question directly; they are just excerpts from retrieved passages. To alleviate this, we create a variation of NOC, called LOC: for each extracted obligation, we prompt an LLM (GPT-4o-mini) to answer the given question using this obligation. If the generated answer does not *cover* (Section 2) the original obligation, then the LLM is prompted again, until a certain number of tries K has been reached (we use $K = 3$). Finally, the per-obligation answers are concatenated to form a complete answer.

4.5 Verify and Refine with RePASs (VRR)

Our third answer generator (VRR) first 'verifies' the correctness of the answers, then iteratively 'refines' them. The first stage (verification) is loosely inspired by self-consistency (Wang et al., 2023); it involves the generation of many alternative answers by the LLM and the selection of the one with the highest RePASs score. The selected answer is then iteratively refined by reducing *contradictions* and increasing *obligations*, as explained below.

4.5.1 Verification step

In the verification step, we obtain N alternative answers from the LLM (using all the extracted obligations and the question as input) and evaluate them using RePASs. We choose the alternative answer with the best RePASs score.

4.5.2 Refinement step

Contradiction removal: To remove contradictions: a) we compute the average contradiction

System / Group Name	RePASs	Obligation	Entailment	Contradiction
GPT-4o baseline*	0.583	0.220	0.769	0.238
Human experts*	0.859	**1.000**	0.837	0.260
Indic aiDias	**0.973**	0.993	**0.987**	0.062
Ocean's Eleven	0.971	0.991	0.986	0.065
AUEB NLP Group - NOC	0.947 (0.951)	0.951 (0.963)	0.986 (0.986)	0.096 (0.096)
AUEB NLP Group - VRR	0.639 (0.646)	0.502 (0.524)	0.446 (0.446)	**0.031** (0.031)
AICOE	0.601	0.230	0.827	0.254
AUEB NLP Group - LOC	0.562 (0.568)	0.423 (0.439)	0.375 (0.375)	0.110 (0.110)

Table 5: Leaderboard results for Subtask 2. Results computed by ourselves for our systems are shown in brackets. Differences are attributed to using different GPUs. *Scores taken from Gokhan et al. (2024).

score over all the answers (over all the best alternative answers for all questions) across the dataset using the same NLI model as in RePASs, and b) we remove the sentences of the answer that get a contradiction score higher than the average.

Obligation insertion: To locate missing obligations, we extract obligations from the retrieved passages and the current answer. Obligations from the retrieved passages that are not *covered* (Section 2) by the current answer are *missing* obligations. We prompt GPT-4o to insert the missing obligations by correcting a sentence or adding a new one to the current answer (complete prompt in Appendix B).

4.6 Experimental results for generation

In the following experiments we use the hidden test set, GPT-4o-mini as the generator for LOC, and GPT-4o[6] as the generator for VRR.

Table 5 compares the task's baseline and human expert performance, as reported by Gokhan et al. (2024), to our three submissions (NOC, VRR, LOC) and to the best submissions of the top three competitors. NOC achieves an almost perfect RePASs score (0.947), surpassing human experts (+0.088). As expected, *obligation* and *contradiction* scores are excellent for the adversarial NOC, but surprisingly *entailment* scores are even better without directly optimizing towards them. Similar results are observed for the methods of the top scoring competitors. However, as already mentioned, NOC's answers are just verbatim sentences from the retrieved passages, which proves that RePASs can easily be deceived. LOC on the other hand, which rewrites the 'obligations' using GPT-4o-mini, performs even worse than the baseline model, which shows that RePASs is also very sensitive to the style of the answer. VRR, which actually generates answers from the retrieved passages, improves upon the task's baseline substantially (+0.056) and ranks first among systems that do not exceed human performance; we suspect that systems with super-human performance may trick the RePASs measure, like our NOC system.

[6]https://openai.com/index/hello-gpt-4o/

VRR	RePASs	Improvement
Baseline (Ours)	0.506	-
+ Verification	0.611	**+ 0.105**
+ Refinement	**0.646**	+ 0.025

Table 6: Contribution of VRR stages, using GPT-4o.

The next experiment (Table 6) measures the contribution of the verification and refinement processes of VRR. Both processes are beneficial, but verification's improvement is more important.

5 Conclusion

We introduced three systems for the RIRAG shared task. The retrieval backbone of all systems combined BM25 with two domain specific neural retrievers and a reranker. We achieved a near-perfect score with an adversarial system that exploits the neural model for *obligation* extraction of RePASs, highlighting the difficulty of developing a robust reference-free metric for RAG evaluation. Our best non-adversarial system (VRR) first generates multiple alternative answers from the retrieved obligations, selects the alternative answer that maximizes RePASs, then iteratively improves it by maximizing obligation coverage and minimizing contradictions. This system produces coherent answers, and obtains the highest RePASs score among competitors that do not exceed human performance (which may be a sign of gaming RePASs).

Limitations

We demonstrated that reference-free model-based metrics, such as RePASs, used for evaluating Retrieval-Augmented Generation (RAG) systems, can be susceptible to adversarial attacks. Specifically, we showed that it is possible to provide answers that receive a high score from the metric, but may not be useful to non-experts. The attack was tailored to RePASs and a specific domain, and it may not apply to other domains or metrics.

VRR requires an accurate verifier, such as RePASs, which is not always available. The *obligation extraction* component in RePASs is fine-tuned using a synthetic dataset (Gokhan et al., 2024), which in turn requires a powerful LLM teacher to solve the task with few-shot prompting alone. This is quite rare for hard domain-specific problems.

Acknowledgments

This work has been partially supported by project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the NextGenerationEU Program. All experiments were done using AWS resources which were provided by the National Infrastructures for Research and Technology GRNET and funded by the EU Recovery and Resiliency Facility.

References

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.

Catalina Goanta, Nikolaos Aletras, Ilias Chalkidis, Sofia Ranchordás, and Gerasimos Spanakis. 2023. Regulation and NLP (RegNLP): Taming large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8712–8724, Singapore. 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. *Preprint*, arXiv:2409.05677.

Yichen Huang and Timothy Baldwin. 2023. Robustness tests for automatic machine translation metrics with adversarial attacks. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5126–5135, Singapore. Association for Computational Linguistics.

Patrick 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 *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, pages 9459–9474, Red Hook, NY, USA. Curran Associates Inc.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202, Online. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2024. Self-refine: iterative refinement with self-feedback. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 46534–46594, Red Hook, NY, USA.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, Johannes Heidecke, Pranav Shyam, Boris Power, Tyna Eloundou Nekoul, Girish Sastry, Gretchen Krueger, David Schnurr, Felipe Petroski Such, Kenny Hsu, Madeleine Thompson, Tabarak Khan, Toki Sherbakov, Joanne Jang, Peter Welinder, and Lilian Weng. 2022. Text and code embeddings by contrastive pre-training. *Preprint*, arXiv:2201.10005.

Xin Quan, Marco Valentino, Louise A. Dennis, and Andre Freitas. 2024. Verification and refinement of natural language explanations through LLM-symbolic theorem proving. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2933–2958, Miami, Florida, USA. 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*, pages 109–126, Gaithersburg, Maryland, USA.

Ante Wang, Linfeng Song, Ye Tian, Baolin Peng, Lifeng Jin, Haitao Mi, Jinsong Su, and Dong Yu. 2024. Self-consistency boosts calibration for math reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6023–6029, Miami, Florida, USA. Association for Computational Linguistics.

Shuai Wang, Shengyao Zhuang, and Guido Zuccon. 2021. Bert-based dense retrievers require interpolation with bm25 for effective passage retrieval. In

Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR '21, page 317–324, New York, NY, USA. Association for Computing Machinery.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: deliberate problem solving with large language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, pages 11809–11822, Red Hook, NY, USA. Curran Associates Inc.

A Related work

RAG: Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) systems can help tackle domain-specific problems that RegNLP (Goanta et al., 2023) presents, by incorporating information from large regulatory document collections.
Verify and Refine: VRR is loosely inspired by LLM methods that select the best answer from multiple candidates and iteratively refine these answers (Wang et al., 2024; Madaan et al., 2024; Yao et al., 2024; Quan et al., 2024), frameworks like Explanation-Refiner (Quan et al., 2024) that use theorem proving to validate and refine explanations, and WizardLM (Xu et al., 2024) that evolves instruction data to enhance model performance.
Adversarial attacks: Many works implement adversarial attacks that are similar to our NOC system. BERT-ATTACK (Li et al., 2020) leverages a pretrained BERT model to deceive other models. Huang and Baldwin (2023) show that popular model-based evaluation metrics for machine-translation are susceptible to inconsistencies when given adversarially-degraded translations.

B Prompts

For all our prompts we have used `GPT-4o` to improve them, and then kept those that performed the task better (according to our opinion) in a few (2-3) sample questions.

Baseline prompt (Gokhan et al., 2024)

You are a regulatory compliance assistant. Provide a detailed answer for the question that fully integrates all the obligations and best practices from the given passages. Ensure your response is cohesive and directly addresses the question. Synthesize the information from all passages into a single, unified answer.

Prompt for obligations in the context (VRR)

You are a regulatory compliance assistant. Your task is to provide a brief but concise and detailed answer to the Question, ensuring that all Obligations are fully addressed. Directly integrate each obligation into the response, ensuring no obligation is missed or implied. Avoid adding information beyond what is explicitly stated in the Obligations, and cite specific rules when necessary. Use the exact terminology and structure from the obligations where applicable, to ensure high alignment and logical consistency. Focus solely on the provided obligations to craft a response that is well-structured, concise, and free of contradictions.

Prompt for inserting obligations (VRR)

You are a regulatory compliance assistant. Your task is to integrate the following Obligations that are missing from the Answer. You may change sentences or add new ones to cover all Obligations. Avoid adding changes or sentences that contradict the Answer and/or the Obligations.

Prompt that rewrites an obligation (LOC)

You are a regulatory compliance assistant. Your task is to construct a brief but concise response that addresses the Question by focusing exclusively on the specified Obligation. Ensure your response clearly identifies and explains the obligation, including any relevant conditions or restrictions. Avoid addressing unrelated aspects of the Question, and limit your response strictly to what is explicitly stated in the provided passage.

C Detailed experiments for VRR

Table 7 shows the progression of RePASs throughout the execution of the VRR algorithm. The Verification step leads to an increase in all metrics. Obligation Refinement ('Ref. Obl.') alone does not lead to an increased score, Contradiction Refinement ('Ref. Contr.') is necessary. Even though

Obligation Coverage ('Obl.') increases at the expense of the Entailment ('Ent.') score, RePASs improves overall.

Step	RePASs	Obl.	Ent.	Con.
Preprocessing	0.506	0.246	0.408	0.136
Verify	0.611	0.389	0.527	0.083
Ref. Contr. 1	0.638	0.389	**0.554**	0.030
Ref. Obl. 1	0.634	0.465	0.490	0.053
Ref. Contr. 2	0.643	0.464	0.497	0.032
Ref. Obl. 2	0.637	0.496	0.464	0.049
Ref. Contr. 3	0.643	0.494	0.467	**0.030**
Ref. Obl. 3	0.642	0.527	0.446	0.046
Ref. Contr. 4	**0.647**	0.525	0.446	0.031
Ref. Obl. 4	0.641	0.538	0.430	0.045

Table 7: RePASs progress during VRR execution.

Structured Tender Entities Extraction from Complex Tables with Few-short Learning

Asim Abbas[1*], Mark Lee[1], Niloofer Shanavas[2], Venelin Kovatchev[1], Mubashir Ali[1]
[1]School of Computer Science, University of Birmingham, Edgbaston, Birmingham, UK
[2]School of Computer Science, University of Birmingham, Dubai Campus, UAE
Correspondence: axa2233@student.bham.ac.uk

Abstract

Extracting structured text from complex tables in PDF tender documents remains a challenging task due to the loss of structural and positional information during the extraction process. AI-based models often require extensive training data, making development from scratch both tedious and time-consuming. Our research focuses on identifying tender entities in complex table formats within PDF documents. To address this, we propose a novel approach utilizing few-shot learning with large language models (LLMs) to restore the structure of extracted text. Additionally, handcrafted rules and regular expressions are employed for precise entity classification. To evaluate the robustness of LLMs with few-shot learning, we employ data-shuffling techniques. Our experiments show that current text extraction tools fail to deliver satisfactory results for complex table structures. However, the few-shot learning approach significantly enhances the structural integrity of extracted data and improves the accuracy of tender entity identification.

1 Introduction

Tenders are formal requests for proposals or bids, typically issued by a company, organization, or government agency seeking goods, services, or works to be provided (Siciliani et al., 2023b). In addition tender documents are the detailed specifications, terms, and conditions accompanying such requests, outlining the requirements and expectations for potential bidders. These documents ensure transparency, fairness, and accountability in the procurement process and are vital for decision in project management (Toikka et al., 2021).
The tender documents contains meaningful information that must be identified and extracted automatically to convert it into actionable knowledge to improve business decisions. Generally, this information is available in an unstructured or semi-structured format (Siciliani et al., 2023a), which is understandable by humans but difficult to understand by machines because of a lack of documents or text structure, contextual understanding, ambiguity and noise, and limited domain knowledge. Similarly, tender documents are large in size, often consist of over 100 pages each. Manually extracting relevant information from such huge documents requires a lot of energy and time and is a labor-intensive task often prone to errors and inefficiencies. To address these challenges, Natural Language Processing (NLP) based applications and techniques have emerged as a promising solution (Fu et al., 2020).
This study addresses two key contributions crucial for automating tasks in the tender domain: i) Structured text extraction from complex tables, a persistent challenge, is essential for tender documents as tables hold organized information vital for accurate analysis. Losing the structure during extraction can result in misinterpretation, affecting decision-making in the tendering process (Milosevic et al., 2019). ii) Tender Named Entities (TNE) recognition and classification, including addresses, project details, dates, and personnel, are critical for retrieving relevant information, generating recommendations, and automating systems like chatbots and IR systems (Ji et al., 2019; Siciliani et al., 2023b; Ji et al., 2019)
To address these challenges, we have introduced a novel approach that combines the capabilities of few-shot learning with large language models (LLMs) (Brown, 2020). Our approach aims to reconstruct the text structure after extraction, thereby facilitating the accurate identification of tender elements. Initially, we leverage existing pdf text extraction tools like pdfminor (PDFMinersix, 2024) to extract raw text from tender documents, focusing on entities within tables. Subsequently we identify the common terms that assist in document segmentation into header, body and footer. We discarded the body text to reduce the data dimensionality and

Proceedings of the 31st International Conference on Computational Linguistics, pages 59–67
January 19–24, 2025. ©COLING 2025

improve entities categorization. Further, header and footer text is concatenated, and few-shot learning with LLM is leveraged to restructure it. After restructuring text, we employed hand crafted rules and regular expressions to automatically classify the entities into explicit categories. Consequently, we achieve high accuracy towards tender entities extraction and classification through few-short learning approach compared to without few-short learning.
Furthermore, our study included comprehensive evaluation, comparison and limitation of existing tools utilized for structure text extraction from tender PDF tables. As a result we found that not a single tool provided desire performance towards structure text extraction from tender PDF tables because of the unstructured and dynamic structured of the tables. Similarly, every tool have their own strength geared specific task. In the same way, in our study, we experimented and combined these tools in one place, utilized their explicit features, and developed our own algorithm to automatically extract the tender elements using rules and regular expressions. Consequently, we achieved state-of-the-art accuracy by integrating a few-shot learning approach.
Further, this study is structured as Related Studies are presented in Section 2. Continuing this, we presented Proposed Approach in section 3. Additionally, in section 4 we presented Experiments, Evaluation and Results of the proposed solution. Finally, we conclude the study with limitation and future work in Section 5.

2 Related Studies

Over the years, a range of methodologies has been explored to enhance the accuracy of entity extraction from complex tender documents, including rule-based systems, machine learning (ML) approaches, and more recently, deep learning(DL) models. While these methods have demonstrated varying degrees of success, the heterogeneity and complexity of tender documents often lead to issues with text structure and format retention, making the task of extracting accurate and relevant entities even more challenging.
In the study (Mehrbod and Grilo, 2018), a rules and self-learning approach using a Conditional Random Field (CRF) model has been introduced to automatically create and update the dictionary over time for recognizing the product entities in tender documents. Moreover, an ontology-based approach for information extraction from construction tender documents has been introduced to convert human-readable document structures into machine-readable formats (Mohemad et al., 2011). However, It is a challenging task to develop universal rules for the entire system, capture intricate semantic links between words, and manage named entities, especially in dynamic specialized domains. Similarly, an incomplete dictionary can lead to a low recall while making a complete dictionary manually is tedious and time-consuming.
To address the challenges with Rule-based systems, state-of-the-art machine learning approaches have solved these challenges extensively in the Tender domain for task such as Named Entities Recognition (Hastie et al., 2009). In the study (Siciliani et al., 2023a) an open Information Extraction for Public Administration (OIE4PA) system was introduced for tender information retrieval from large databases. It extracts information based on triples (Subject, Predicate, Object) found in documents, using tools like UDPipe[1] and WikiOIE (Siciliani et al., 2021). In this process, two domain experts manually labeled these triples, which served as the training data for machine learning models such as Support Vector Machines (SVM), XGBoost, and logistic regression (LR). However, machine learning models may not always reach peak performance on labeled data due to potential issues of over-fitting or under-fitting. This sometimes requires the use of intricate feature engineering approaches.
Conspicuously, DL is differs from the classical ML approaches by diminishing the demand for manually designing features such as bag-of-words or n-grams (Wu et al., 2020; Medsker and Jain, 2001; Wolf et al., 2020). As discussed previously, this study (Chalkidis et al., 2017) is further extended by implementing deep learning models such as Bi-LSTM, LSTM, and Conditional Random Field (CRF) to extract the contract entities automatically (Chalkidis and Androutsopoulos, 2017). Moreover, a DL-based approach is introduced (Ji et al., 2019) to enhance the automatic identification of tender entities. The proposed architecture incorporates five main layers, for instance embedding layer using BERT, BiLSTM input BERT embedding as feature vector, feature fusion layer, attention layer, and CRF layer. Similarly, RNN-based architecture models have several constraints and lim-

[1] https://lindat.mff.cuni.cz/services/udpipe/

itations. These include the vanishing gradient problem during back-propagation through time, which makes it challenging for the model to learn long-range dependencies in sequential data. Furthermore, RNNs are sensitive to hyper-parameters such as learning rate, batch size, and sequence length.
To address the limitations of existing methodologies, we introduce a Large Language Model (LLM)-based approach that incorporates few-shot learning for tender entity extraction and classification. By leveraging LLMs, we eliminate the need to build AI models from scratch, which typically requires vast training datasets and the involvement of domain experts is costly and time-consuming process. Furthermore, we combine this AI-driven approach with rule-based methods to enhance accuracy in tender entity extraction and classification, while eliminating the need for ongoing lexicon and rule maintenance across the entire system.

3 Proposed Approach

This section outlines the methodology for structured tender text extraction and entity classification. We first used tools such as PDFMiner (PDFMiner-six, 2024) to extract text from complex PDF tables. A text analyzer was then designed to perform pre-processing, keyword identification, and document segmentation into header, body, and footer. After discarding the body, we concatenated the header and footer. Due to the loss of semantics, context, and sequence in extracted text, we employed a few-shot learning approach to restructure it. Finally, regular expressions and rules were applied to classify tender entities into defined categories.

3.1 Tender Table Structure and Entities

The tender entities of interest are found in the first section of each document, and formatted as complex and dynamic tables. Each tender typically begins with a preamble containing the Tendree (Buyer) title (see Figure 1), followed by the Tenderer (Supplier) information, including company name, address, date, tender number, and project name. The body of the tender includes the project description, supplied items, and terms and conditions. At the end, Tendree information, such as name, address, signatory details, and phone number, is listed.
These entities (Tendree and Tenderer) are located in specific table zones but within a dynamic structure. The width and height of table cells change depending on the length of the tender entities. For example, a longer Tenderer name or address will expand the table to cover more rows in a cell, but the information is always located in the start zone. Similarly, Tendree information is consistently found in the footer zone.
The table structure (Figure 1) is highly complex due to frequent merging and splitting of rows and columns, causing alterations during data extraction. Names, addresses, dates, job numbers, and other entities are scattered across multiple columns and rows, requiring careful extraction.

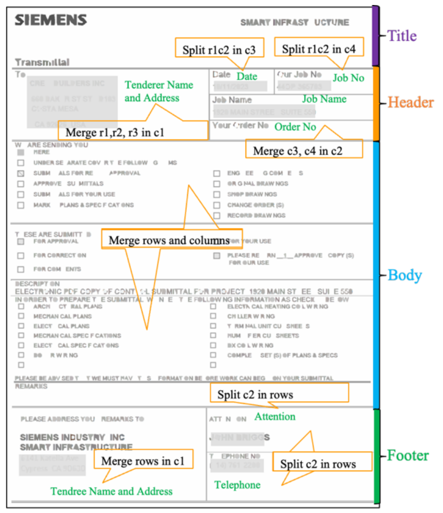

Figure 1: Typical structure of a table in tender document, with tender entities highlighted

3.2 Text Extraction Complexity

At the top of the table there is a title of Tendree, which appears on both the left and right corners in bold (Figure 1). If any part of the title, like "SIEMENS" on the left or "SMART INFRASTRUCTURE" on the right, is missed, existing tools often fail to recognize it. Similarly, the table header contains five tender entities: the tender name and address share a cell, while the tender date, number, and project name are in separate cells. Structurally, the table header has two columns with four cells, though row numbers vary based on the length of elements specifically tender name, address, and project name. In the same way, the footer section has two columns and three cells, with the tender address cell expanding due to longer addresses. Additionally, tender elements within cells may not align consistently, causing gaps between attribute and value horizontally or vertically. This inconsistency leads to the loss of text sequence, as existing tools read PDFs row-wise.

3.3 Text Analyzer

Subsequently, text extraction from tables using tools such as pdfminer (PDFMinersix, 2024) revealed a loss of text sequence, semantics, and context, making it difficult for both humans and machines to interpret. This also complicates applying rules and regular expressions for tender element identification. To address this, we introduced a text analyzer module, which incorporates text preprocessing, keyword identification, and keyword-based segmentation of tender information.

3.3.1 Text Processing

After extracting text from tender PDF tables, we observed that punctuation marks were misplaced, which caused issues with the regular expressions specifically designed for identifying tender elements. For example tender number in table is appear in *"44OP-123456"*. After extraction a punctuation dot(.) has added in the start or end of tender number such as *".44OP-123456"* or *"44OP-123456."*, that become challenging to maintain the regular expression. Similarly, after text extraction, extra spaces between words both horizontally and vertically were found, making the text more challenging to process, reducing readability, and causing inconsistencies during parsing.

3.3.2 Keywords Identification

After extracting text from the table, a lack of predefined format was observed. Identifying keywords helps recognize patterns and structure. Additionally, Keywords help in extracting relevant information, making it easier to categorize and segment the text. For instance to extract the header information the keyword *"PROJECT NAME"* help to extract all the text before this. Similarly extracting body information, the keywords “FOLLOWING ITEMS", and *"REMARKS TO"* helps to extract text between these words. Finally the keyword *"SIEMENS INDUSTRY"* etc help to extract the footer information.

3.3.3 Keywords based Document Segmentation

To accurately segment unstructured text into header, body, and footer, we introduced a keyword-based document segmentation approach to enhance algorithm efficiency and performance in extracting Tender Named Entities (TNE) from tender tables. Unnecessary text increases complexity, so identifying key phrases helps reduce this. As shown in Figure 1, TNEs are located before the phrase *"WE ARE SENDING YOU"* and extracted by recognizing the table attribute *"JoB Name"*. The header entities are extracted from text preceding this phrase, while the body is between *"WE ARE SENDING YOU"* and *"REMARKS TO"*. Similarly, footer TNEs are located between *"SIEMENS INDUSTRY"* and *"TELEPHONE NO"*. By discarding the body section and combining header and footer data, we reduce dimensionality, making the data more manageable, lowering storage needs, and speeding up processing.

3.4 Structured text generation employing few-Shot learning

When header and footer text is concatenated, the resulting text become unstructured with sequence, semantic and context lost, posing significant challenges for further processing. To address this, we proposed a few-shot learning approach leveraging LLM effectively. To enable the LLM to handle this unstructured data, a set of manually prepared structured example samples is created. These examples consists of pairs of unstructured data and their corresponding structured formats, showcasing the desired outcome. These example samples are then used to guide the LLM through prompt engineering, teaching it to understand and convert unstructured data into a structured format. Using in-context learning, new unstructured data can be processed by the model to generate structured outputs efficiently. This approach ensures that the LLM can effectively maintain the correct sequence and organization of data, transforming unstructured text into a well-structured format suitable for further use.

3.4.1 Case Study

The Figure 2 illustrates a case study on few-shot learning prompting using a ChatGPT-3.5 (LLM) for structure tender text generation.The diagram is splited into two main section, each demonstrating the conversion of unstructured input text into a structured format using example-based learning. The top section outlines a few-shot learning approach where example pairs of unstructured (Input) and structured (Output) texts are provided to the LLM with the specific instruction known as instructed prompt. The input data, such as *"TRANSMITTAL SMART INFRASTRUCTURE To: FANDERSON RAVE & BULCKLEY INC..."*, is processed to generate a corresponding structured output, showcasing the model’s ability to learn from

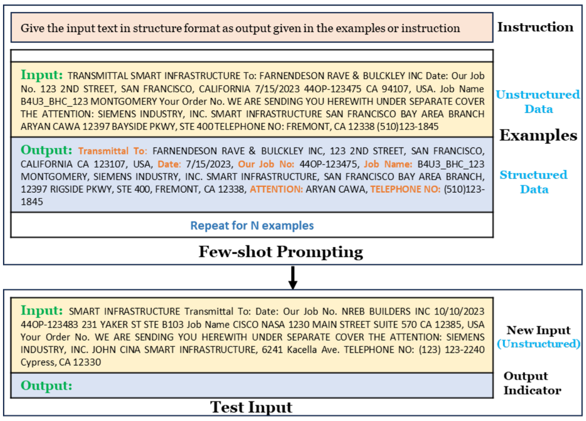

Figure 2: An example case study for understanding few-shot Learning approach towards structured data generation

minimal examples. Below this, the diagram repeats the process for multiple examples to reinforce the model's learning. The bottom section shows a new unstructured input(Test Input) being transformed into a structured output based on the learned examples. The prompt provided at the top guides the model to apply the learned structure to new inputs, demonstrating the efficacy of few-shot learning in generating accurate structured data from unstructured text with minimal examples. Ultimately, structure tender text output, heuristic approach is applied to identify and classify the tender elements into explicit categories.

3.5 Tender Entities Recognition and Classification

After concatenation of header and footer text, we acquired a text paragraph incorporated Tender Named Entities(TNE). We have designed a set of hand crafted rules and regular expressions for explicit entities recognition and classification. As shown in Table 1, the same regular expression is used for identifying the names and addresses of both the Tendree and Tendrer, as their naming patterns are similar. However, they have distinct content: the names of the Tendree and Tendrer typically end with "$INC|COMPANY$", while addresses start with a house or building number and end with a postal code followed by the country. Similarly, Tender date appeared in various format for example "$dd/mm/YYYY$" or "$dd - mm - YYYY$". Further, the Tender Number is always started with the two digit followed by two character and any six digit value after hypen. Moreover, the Tender Name is always located between *"Job Name"* and *"Your Order No*". However, sometimes the Tender Name may resemble the Tenderer's address, which can lead to incorrect classification as the Tenderer's address. Similarly, Tenderee Personal entity is challenging to accurately identify because every tender have variant personal name but it usually available in the text before *"TELEPHONE NO:"* pattern. Finally in footer Tenderee Telephone number is available in various format such as *"xxx-xxx-xxxx"* or *"(xxx)xxx-xxx"* or *"xxx.xxx.xxx"* etc. Our algorithm first attempts to extract entities using regular expressions. If any entities are missed, we have developed rules that work in conjunction with regular expressions to identify and extract the tender entities.

4 Experiments, Evaluation and Results

We have utilized 30 commercial confidential tender documents, that can not be released. Each document contains over 100 pages of information intended for architects, engineers, or project owners, submitted for approval by the contractor. The required tender entities are available in the form of complex table, which is difficult to extract accurately. In this section, we present the results of our Pre-fewshot and Post-fewshot learning approaches, which are combined with rules and regular expressions for structured text extraction from complex tables and the classification of tender entities.

4.1 Structured Text Extraction From Complex Table

To evaluate the structure text extraction utilizing existing tools by integrating few-shot learning approach, we explored two empirical approaches: a) Average Relative Distance (ARD) and b) BLEU score. Similarly, we computed the results in two different way: i) structured text extraction Pre-few-shot learning and ii) structured text extraction Post-few-shot learning. Similarly, we incorporated data shuffling technique to deterministically shuffle data 50% and 100% to asses the strength of LLMs in a few-shot learning environment towards structured data generation.

4.1.1 Average Relative Distance (ARD)

The Average Relative Distance (ARD) is a metric used to quantify the average movement of words from their positions in the original text to their positions in the extracted text. The resulting ARD value provides an indication of how much, on average, the words have shifted from their original locations

Table 1: Tender Named Entities Recognition(TNER) Regular Expressions

<table>
<tr><th>Named Entities</th><th>Regular Expression</th></tr>
<tr><td>Tenderee/Tenderer Name</td><td>.*INC$|. * COMPANY.$</td></tr>
<tr><td>Tenderee/Tenderer Address</td><td>\b\d+\s.*?\d+\b</td></tr>
<tr><td>Tender Date</td><td>(\d{1,2}\/\d{1,2}\/\d{4})|(\d{1,2} − \d{1,2} − \d{4})</td></tr>
<tr><td>Tender Number</td><td>(\d{2}[A − Z]{2} − d{6})</td></tr>
<tr><td>Tender Name</td><td>Job Name\s*(.*?)\s*Your Order No</td></tr>
<tr><td>Tenderee Personal</td><td>∧(.*?)\bTELEPHONE\s?NO :?\b</td></tr>
<tr><td>Tenderee Telephone</td><td>\b(?:\+ \d{1,2}?)?\(?\d{3}\)?[-.\s]?\d3[-.\s]?\d{4}\b</td></tr>
</table>

in the extracted text. The Average Relative Distance (ARD) emperical formula can be expressed as shown Eq.1.

$$\text{ARD} = \frac{1}{N} \sum_{i=1}^{N} |p_i - q_i| \quad (1)$$

where N is the number of words in the original text, p_i is the position of the i-th word in the original text, and q_i is the position of the i-th word in the extracted text. ARD score lower (↓) is better and score higher(↑) is worst. The Figure 3 demonstrates the effectiveness of applying few-shot learning approach to improve structured text extraction from complex tables in tender PDF documents by computing ARD. Before few-shot learning, all tools struggled to maintain the structure and sequence of the extracted text, as indicated by relatively high ARD values. Among the tools, PyMUPDF performed the best initially with an ARD of 2.88, while PyPDF2 had the worst performance with an ARD of 7.22. As the data was shuffled, the ARD values increased across all tools, with the highest distortion occurring at 100% shuffling, where PyMUPDF and PyPDF2 reached ARD values of 10.99 and 14.18, respectively. This indicated severe structure loss with increasing shuffling.

After applying few-shot learning using ChatGPT-3.5, there was a remarkable reduction in ARD across all tools, showing the approach's effectiveness in restructuring the text. UnStructured.io and PyMUPDF showed the most significant improvements, with post-few-shot ARD values dropping as low as 0.37 and 0.49, respectively, in non-shuffled conditions. Even with 100% shuffled data, these tools maintained relatively low ARD values of 3.41 for PyMUPDF and 3.84 for UnStructured.io. PDF Minor and PyPDF2, while benefiting from few-shot learning, still had slightly higher Post-few-shot ARD values (around 4.27 and 3.04, respectively) when dealing with fully shuffled data.

Overall, few-shot learning drastically improved text extraction performance, particularly for UnStructured.io and PyMUPDF, which consistently achieved the lowest ARD values across all conditions. The results clearly demonstrate that applying few-shot learning to these tools can effectively restore text structure, even when the data is shuffled, although some tools still exhibit minor distortions in highly shuffled scenarios.

4.1.2 Average BLEU Score

BLEU (BiLingual Evaluation Understudy), is a metric used in NLP and machine translation to evaluate the quality of candidate text against one or more high quality reference text . It measures how similar a Tools extracted text is to one or more manually structured text.

The BLEU score is based on precision of n-grams (up to a certain length) between the candidate sentence and reference sentences. Here's the mathematical formula for BLEU score:

$$\text{BLEU}_{\text{avg}} = \frac{BP \cdot \exp\left(\sum_{n=1}^{N} w_n \cdot \log(p_n)\right)}{T} \quad (2)$$

Where, BP is the brevity penalty, when the candidate text length less than/equal (≤) to the reference text length, w_n are the weights for each n-gram precision, p_n is the precision for n-grams, N is the maximum n-gram length considered and T is divide the sum of BLEU scores by the total number of sentence pairs. Notably, Average BLEU Score(ABS) higher (↑) is better and score lower (↓) is worst.

The Figure 4 shows the ABS for structured text extraction from complex tables in tender PDFs using five tools before and after applying few-shot learning with ChatGPT-3.5.

Before few-shot learning, BLEU scores dropped as shuffling increased, indicating a loss of text struc-

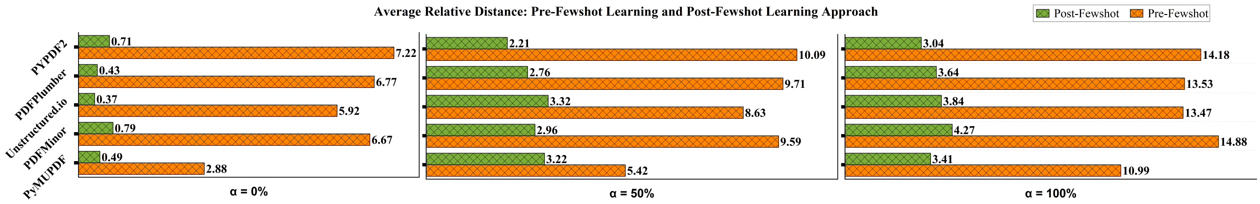

Figure 3: Performance of Structured Tender Text Extraction from Complex Tables: Pre- and Post-few-Shot Learning computed Average Relative Distance utilizing Text Extraction Tools and LLMs

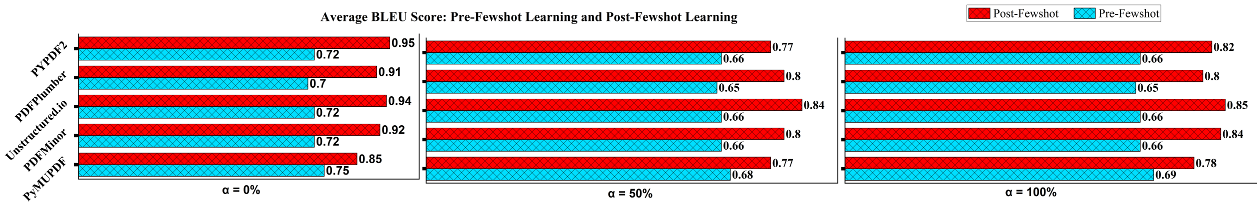

Figure 4: Performance of Structured Tender Text Extraction from Complex Tables: Pre- and Post-few-Shot Learning computed Average BLEU score utilizing Text Extraction Tools and LLMs

ture. PyMUPDF, for example, had a BLEU score of 0.75 for unshuffled data, which fell to 0.68 and 0.69 with partial and full shuffling, respectively. Similar patterns were observed with the other tools, where Pre-few-shot BLEU scores ranged from 0.65 to 0.72. After few-shot learning, all tools saw significant improvements in BLEU scores. PyMUPDF's score increased to 0.85 (unshuffled), 0.77 (50% shuffled), and 0.78 (100% shuffled), reflecting better text structure restoration. Likewise, UnStructured.io and PyPDF2 performed the best, with Post-few-shot scores of up to 0.95 and 0.94, respectively, in unshuffled and shuffled conditions. PDF Minor and PDFPlumber also improved but trailed slightly behind.

Inclusively, few-shot learning greatly improved structure preservation across all tools, with UnStructured.io and PyPDF2 emerging as the top performers, particularly in shuffled data scenarios.

4.2 Tender Entities Classification

We have developed a scoring-based method to evaluate the Named Entity Recognition and Classification of tenders, comparing the results obtained before and after applying a few-shot learning approach. The point scoring criteria is set as Full Match (F_i): 2 points, Partially Match(P_i): 1 points, Not Match (N_i): -1 point (small penalty for not matching) and Wrong Match (W_i): -2 points (higher penalty for incorrect matching) as shown in Equ. 3.

In Equ. 4, for a document with n categories, where each category can score a maximum of 2 points (for a Full Match) and minimum of -2 point (Wrong Match), So the maximum possible score is $S_{max} = 2*n$ and minimum possible score is $S_{min} = -2*n$. Similarly, we plan to evaluate Pre-few-shot learning and Post-few-shot learning approach at entities level and documents level.

$$S_{actual} = \sum_{i=1}^{n} S_i = \sum_{i=1}^{n} (2F_i + P_i - N_i - 2W_i), \quad (3)$$

Where S_i is calculated score of each category, and n is the number of categories.

$$Accuracy(\%) = \left(\frac{S_{actual} - S_{min}}{S_{max} - S_{min}} \right) \times 100 \quad (4)$$

The Table 2 presents a comprehensive analysis of accuracy improvements for nine tender entities before (Pre-Fewshot) only rules and regular expression and after (Post-Fewshot) applying the few-shot learning approach integrating rules and regular expression. These results are evaluated across five different text extraction tools: PDFMiner, PyMUPDF, Unstructured.io, PDFPlumber, and PyPDF2.

Overall, the results clearly indicate that the Post-few-shot learning approach significantly improves the accuracy of tender entity extraction across all tools. For instance, the entity "To Company" achieves 100% accuracy Post-few-shot learning across all tools, with notable improvements in tools like Unstructured.io (from 82.67% to 100%) and PDFPlumber (from 85.33% to 100%). Similarly, "To Address" shows considerable improvements,

Table 2: Entity-level tender documents evaluation Pre- and Post-Few-shot Learning Approach

Entities	PDFMiner (%)		PYMUPDF (%)		Unstructured.io (%)		PDFPlumber (%)		PYPDF2 (%)	
	Pre	Post	Pre	Post	Pre	Post	Pre	Post	Pre	Post
To Company	89	100	97.63	100	82.67	100	85.33	100	81.67	100
To Address	93	100	95	100	92	100	70.33	100	63	95.33
Date	100	100	100	100	95	100	100	100	95	100
Tender No	100	100	100	100	100	100	100	100	95	100
Tender Name	95	100	97.3	92	80	100	91.67	100	94.33	100
From Company	100	100	90	92	100	100	89.33	100	83.67	100
From Address	98.67	100	87	92	100	100	87.33	100	82	100
Attention	94.33	100	91.33	92	80.33	100	37.67	100	38	100
Telephone No	100	100	90	100	92.67	100	63.33	100	49.67	100

especially with PyMUPDF (from 95% to 100%) and Unstructured.io (from 92% to 100%). The "Date" and "Tender No" entities maintain 100% accuracy across all tools in both Pre- and Post-few-shot learning phases, indicating these entities are well-recognized due to pattern consistency regardless of the method used. However, other entities such as "Attention" and "Telephone No" benefit considerably from the few-shot learning approach. For example, "Attention" sees an impressive boost in PDFPlumber (from 37.67% to 100%) and PyPDF2 (from 38% to 100%).

The results also highlight the variability in pre-few-shot performance among the tools, with some, such as PyPDF2, initially struggling with lower accuracy rates across several entities, like "From Address" (82% pre, 100% post) and "Telephone No" (49.67% pre, 100% post). However, the application of few-shot learning substantially bridges these gaps across all tools, showing that the model can effectively generalize from a few examples.

Concisely, few-shot learning demonstrates its utility in improving entity extraction, particularly in tools like Unstructured.io, PDFPlumber, and PyPDF2, which showed weaker performance in the pre-few-shot phase. Across all tools and entities, the Post-few-shot learning results consistently approach or reach 100% accuracy, underscoring the approach's effectiveness in entities extraction from complex tender tables.

5 Conclusion, Limitation and Future Work

Structured text extraction from the complex table is an ongoing research challenge despite various AI tools and techniques. This study leverages LLMs in a few-shot learning environment to enhance tender entity classification from complex tables in PDF tender documents. We integrated text extraction tools, rules, and regular expressions with LLMs, and introduced text shuffling (50% and 100%) to assess LLMs capability in structured text extraction. After obtaining structured text, we applied hand-crafted rules and regular expressions for precise entity classification. Similarly, we assessed several text extraction tools towards structured data extraction, as a result we found that not a single tool provided desire performance. The experimental results demonstrate that after text extraction from tables employ few-shot learning significantly improves performance and accuracy, addressing the challenge of structured text extraction from complex tender tables.

However this research has several limitations to be addressed in future work. First, the model's performance is heavily dependent on large datasets, especially in the Pre-few-Shot learning phase, where accuracy may decline with smaller datasets. Additionally, the few-Shot learning approach needs to be tested on datasets from other domains for a more robust evaluation. Finally, exploring zero-shot and one-shot learning approaches would further validate our findings.

In future, we aim to address the limitations of our study as discussed by expanding the dataset size and diversity to improve the accuracy and reliability of few-shot learning approach for structured data extraction and tender name entity recognition.

References

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Ilias Chalkidis and Ion Androutsopoulos. 2017. A deep learning approach to contract element extraction. In *JURIX*, volume 2017, pages 155–164.

Ilias Chalkidis, Ion Androutsopoulos, and Achilleas Michos. 2017. Extracting contract elements. In *Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law*, pages 19–28.

Sunyang Fu, David Chen, Huan He, Sijia Liu, Sungrim Moon, Kevin J Peterson, Feichen Shen, Liwei Wang, Yanshan Wang, Andrew Wen, et al. 2020. Clinical concept extraction: a methodology review. *Journal of biomedical informatics*, 109:103526.

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. 2009. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

Yunfei Ji, Chao Tong, Jun Liang, Xi Yang, Zheng Zhao, and Xu Wang. 2019. A deep learning method for named entity recognition in bidding document. In *Journal of Physics: Conference Series*, volume 1168, page 032076. IOP Publishing.

Larry R Medsker and LC Jain. 2001. Recurrent neural networks. *Design and Applications*, 5(64-67):2.

Ahmad Mehrbod and António Grilo. 2018. Tender calls search using a procurement product named entity recogniser. *Advanced Engineering Informatics*, 36:216–228.

Nikola Milosevic, Cassie Gregson, Robert Hernandez, and Goran Nenadic. 2019. A framework for information extraction from tables in biomedical literature. *International Journal on Document Analysis and Recognition (IJDAR)*, 22:55–78.

Rosmayati Mohemad, Abdul Razak Hamdan, Zulaiha Ali Othman, and Noor Maizura Mohamad Noor. 2011. Ontological-based information extraction of construction tender documents. In *Advances in Intelligent Web Mastering–3: Proceedings of the 7th Atlantic Web Intelligence Conference, AWIC 2011, Fribourg, Switzerland, January, 2011*, pages 153–162. Springer.

PDFMinersix. 2024. Pdfminersix. https://pdfminersix.readthedocs.io/en/latest/. [Online; accessed 02-June-2024].

Lucia Siciliani, Pierluigi Cassotti, Pierpaolo Basile, Marco de Gemmis, Pasquale Lops, Giovanni Semeraro, and Aldo Moro. 2021. Extracting relations from italian wikipedia using self-training. In *CLiC-it*.

Lucia Siciliani, Eleonora Ghizzota, Pierpaolo Basile, and Pasquale Lops. 2023a. Oie4pa: open information extraction for the public administration. *Journal of Intelligent Information Systems*, pages 1–22.

Lucia Siciliani, Vincenzo Taccardi, Pierpaolo Basile, Marco Di Ciano, and Pasquale Lops. 2023b. Ai-based decision support system for public procurement. *Information Systems*, 119:102284.

Esa Toikka et al. 2021. Information extraction from procurement contracts. Master's thesis.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Stephen Wu, Kirk Roberts, Surabhi Datta, Jingcheng Du, Zongcheng Ji, Yuqi Si, Sarvesh Soni, Qiong Wang, Qiang Wei, Yang Xiang, et al. 2020. Deep learning in clinical natural language processing: a methodical review. *Journal of the American Medical Informatics Association*, 27(3):457–470.

A Two-Stage LLM System for Enhanced Regulatory Information Retrieval and Answer Generation

Fengzhao Sun[1], Jun Yu[1*], Jiaming Hou[2], Yutong Lin[1], Tianyu Liu[3]
[1]University of Science and Technology of China,
[2]Harbin Institute of Technology,
[3]Jianghuai Advance Technology Center
sunfz@mail.ustc.edu.cn, harryjun@ustc.edu.cn, 23s105157@stu.hit.edu.cn,
linyutong@mail.ustc.edu.cn, liutianyu18@mails.ucas.ac.cn

Abstract

This technical report describes our methodology for the Regulatory Information Retrieval and Answer Generation (RIRAG) Shared Task, a component of the RegNLP workshop at COLING 2025. The challenge aims to effectively navigate and extract relevant information from regulatory texts to generate precise, coherent answers for compliance and obligation-related queries. To tackle subtask1, we introduce a two-stage approach comprising an initial output stage and a subsequent refinement stage. Initially, we fine-tune the LLaMa-2-7B model using LoRA to produce a preliminary output. This is followed by the application of an expert mechanism to enhance the results. For subtask2, we design specific prompt to facilitate the generation of high-quality answers. Consequently, our approach has achieved state-of-the-art performance on the leaderboard, which serves as a testament to the effectiveness and competitiveness of our proposed methodology.

1 Introduction

Regulatory documents, issued by government bodies, detail compliance rules across various areas like environmental standards and data protection. They are complex, comprehensive, and frequently updated, making them challenging to interpret and keep up with. To manage these documents effectively, specialized NLP techniques, such as information retrieval and question answering, are essential for industries facing governance and compliance challenges.

At the same time, the rapid development of large language models (LLMs)(Brown et al., 2020; OpenAI, 2022; Achiam et al., 2023; Touvron et al., 2023a,b; Chiang et al., 2023) has been remarkable, with major breakthroughs in various fields. This progress implies that LLMs could offer innovative solutions and tools to enhance the processing and comprehension of regulatory documents.

However, simply deploying LLMs in mission-critical domains such as healthcare, law, and finance poses unique challenges that go beyond general AI optimization and alignment. A primary concern is the models' propensity to generate plausible but incorrect "hallucinatory" responses, especially in specialized domains where data is limited or complex. Furthermore, the vast expansion of online data, along with the substantial resources needed for data annotation and model training, makes it difficult for LLMs to stay up-to-date. Recent innovations are trying to tackle these issues. Retrieval-Augmented Generation (RAG)(Lewis et al., 2020) integrates information retrieval to update static knowledge. Chain of Thought(Wei et al., 2022) prompting has led to task-specific workflows. These are enhanced by parameter-efficient fine-tuning methods like Low-Rank Adaptation (LoRA)(Hu et al., 2022) and Reinforcement Learning from Human Feedback (RLHF)(Ouyang et al., 2022), which improve model performance with minimal parameter increases. However, these methods rely on substantial datasets. In response, an Automated Question Passage Generation task for RegNLP has been defined(Gokhan et al., 2024), creating the ObliQA dataset with 27,869 questions from the ADGM financial regulation documents, providing a rich resource for training and refining LLMs in regulatory compliance.

In this study, we introduce a two-stage framework for regulatory document processing: Parameter-Efficient Fine-tuning of Large Language Models (LLMs) and a Hybrid Expert Mechanism. Our key contributions are as follows: (1) Parameter-Efficient Fine-tuning of LLMs: We have fine-tuned the general-purpose LLaMa-2-7B model to specialize in domain-specific retrieval tasks. Our experiments confirm that this approach significantly

*Corresponding author.

Proceedings of the 31st International Conference on Computational Linguistics, pages 68–72
January 19–24, 2025. ©COLING 2025

enhances the model's ability to accurately retrieve information for regulatory-related queries. (2) Hybrid Expert Mechanism: Beyond conventional rank fusion techniques, we propose a novel expert mechanism designed to refine outputs from various experts. This mechanism ensures a higher level of precision and reliability in the final results.

2 Method

2.1 Problem Restatement

The Regulatory Information Retrieval and Answer Generation (RIRAG) task encompasses two distinct subtasks:
Subtask 1: Passage Retrieval. Given a regulatory question, identify and retrieve the most relevant passages, specifically obligations and related rules, from ADGM regulations and guidance documents.
Subtask 2: Answer Generation. Synthesize the retrieved data into precise and informative responses that comprehensively address the regulatory query.

2.2 Dataset

In this paper, we mainly use the ObliQA dataset(Gokhan et al., 2024), a multi-document, multi-passage Question Answering (QA) resource designed to advance research in Regulatory Natural Language Processing (RegNLP). The dataset's creation was a three-phase process encompassing Data Collection, Question Generation, and Question-Passages Validation using Natural Language Inference (NLI). Comprising 27,869 questions and their corresponding source passages, ObliQA is sourced entirely from the comprehensive regulatory documentation provided by Abu Dhabi Global Markets (ADGM), the regulatory authority for financial services in the UAE's free economic zones. This dataset is uniquely tailored to facilitate and enhance the development of models capable of retrieving and generating accurate regulatory information.

2.3 Passage Retrieval

In this section, we present a practical implementation of the two-stage framework through a case study in the regulatory sector. The overall architecture of our approach is shown in Fig. 1

Stage 1: Parameter-Efficient Fine-tuning of LLM. Our first stage is dedicated to specializing the general-purpose LLaMa-2-7B model for regulatory-related retrieval tasks, concurrently mitigating the "hallucination" issues common in LLMs.

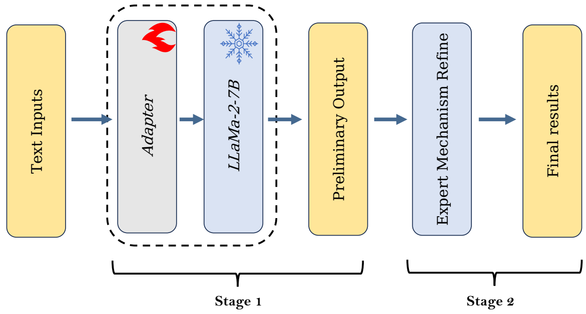

Figure 1: The overall pipeline of our method for Passage Retrieval.

This process aims to yield preliminary results that can be utilized in subsequent stages. During training, we employ the ObliQA dataset to fine-tune the LLaMa-2-7B model. Considering the model's vast parameter count of 7 billion, direct fine-tuning would significantly strain memory resources. To overcome this, we adopt Low-Rank Adaptation (LoRA), an efficient parameter fine-tuning method that allows us to freeze the existing weights and train only a few adapter layers on top of the base model. By adding these adapters to all linear layers of the model, we retain full control over the model's location and network. The following Table 1 provides a detailed overview of our training configuration.

Hyperparameter	LoRA
learning rate	3×10^{-4}
batch size	8
epochs	50
max length	128
r	4
dropout	1.00×10^{-3}
alpha	64

Table 1: LoRA hyperparameters.

Stage 2: Hybrid Expert Mechanism for Refinement Stage 2 refines Stage 1's results using an expert mechanism that leverages diverse retrieval strengths. It integrates Stage 1's advanced output with traditional methods like BM25, enhancing accuracy and system robustness. The expert mechanism's architecture is depicted in Fig. 2.

This approach offers more flexibility than simple rank fusion by allowing dynamic weight adjustment based on task complexity and expert performance. Spceifically, Let $E_1, E_2, \ldots, E_n$ represent the outputs from the traditional experts (e.g., BM25), and E_{LLM} represent the output from the

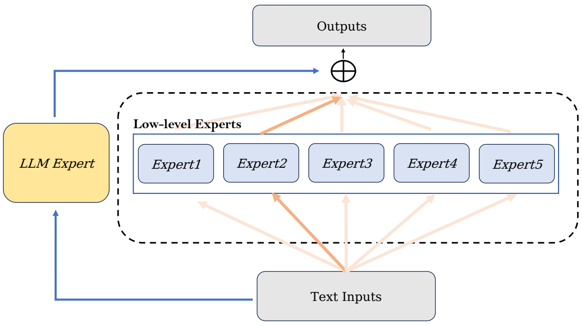

Figure 2: The overall architecture of our Hybrid Expert Mechanism for Refinement.

fine-tuned LLM in Stage 1. Each expert's output is assigned a weight w_i based on its relevance and accuracy. The weighted sum of the expert outputs is calculated as:

$$S = \sum_{i=1}^{n} w_i E_i + w_{LLM} E_{LLM} \quad (1)$$

where w_{LLM} is the weight assigned to the LLM expert.

The output from Stage 1 is considered a high-level expert, and its contribution is combined with the traditional experts' outputs. The final output Y is a weighted combination of all experts' outputs:

$$Y = \frac{\sum_{i=1}^{n} w_i E_i + w_{LLM} E_{LLM}}{\sum_{i=1}^{n} w_i + w_{LLM}} \quad (2)$$

The weights w_i and w_{LLM} can be determined through a training process where the system learns to optimize the combination of experts' outputs for the best performance on a validation set. The combined output Y is then refined through additional processing steps to produce the final result.

We conducted experiments with five retrieval models to serve as our traditional experts: (1) *BM25*(Robertson et al., 1994), a foundational lexical-based model; (2) *DRAGON+*(Lin et al., 2023), a State-of-the-Art (SotA) single-vector dense retriever fine-tuned on the MS MARCO dataset; (3) *SPladev2*(Formal et al., 2021), a SotA neural sparse retriever also fine-tuned on MS MARCO; (4) *ColBERTv2*(Santhanam et al., 2021), a SotA multi-vector dense retriever model, also fine-tuned on MS MARCO. and (5) *Roberta*(Liu et al., 2019), another state-of-the-art method, in our experiments. By integrating the advanced capabilities of the fine-tuned LLM with the robustness of traditional retrieval methods, this hybrid approach leverage the strengths of both to achieve superior performance in regulatory-related retrieval tasks.

2.4 Answer Generation

Drawing inspiration from prior work (Gokhan et al., 2024), we initiate the answer generation process once we have identified 10 relevant passages per query from our passage retrieval system. Transitioning to the post-retrieval phase, we apply a scoring-based filtering strategy. This strategy uses a threshold of 0.25 to identify significant drops in relevance between consecutive passages, ensuring a smooth relevance gradient. Furthermore, we enforce a minimum score threshold of 0.7, which ensures that only the most relevant passages are considered for answer generation.

Armed with these carefully selected passages, we utilize the Qwen2-72B model to generate comprehensive answers. The model is guided by a custom prompt designed to simulate the role of a regulatory compliance assistant. This prompt integrates all critical obligations and best practices from the passages into a cohesive response. The prompt is structured as follows:

System Prompt
You are a regulatory compliance assistant. Provide a detailed answer for the question that fully integrates all the obligations and best practices from the given passages. Ensure your response is cohesive and directly addresses the question. Synthesize the information from all passages into a single, unified answer. Please think step by step.

Table 2: Prompt Design for Regulatory Compliance Assistant.

3 Experiment

In this section, we state the details of the experimental implementation. Finally, we present the corresponding experimental results.

3.1 Implementation Details

We implement our proposed model using the PyTorch framework. Here are the details of the training process: (1) During the parameter-efficient fine-tuning phase, we train the model with 8 × NVIDIA A100 GPUs. The batch size is set at 8 and adopt the AdamW optimizer with a base learning rate of $3e^{-4}$. Furthermore, we apply the lora technique with a rank of 8 and an alpha value of 64 to fine-tune the model parameters effectively. (2) During hybrid expert mechanism for refinement phase, to

Model	Passage-only		Rank fusion		Hybrid Expert	
	R@10	M@10	R@10	M@10	R@10	M@10
BM25	64.2	50.9	64.2	51.0	64.6	51.2
DRAGON+	61.4	46.3	61.9	46.3	61.5	47.7
SPLADE	64.2	49.6	64.1	49.5	63.1	55.0
ColBERTv2	64.5	52.7	64.6	52.7	70.3	56.7
ROBERTA	65.2	51.5	65.8	52.3	71.7	57.1
LLaMa-7B	68.4	55.4	69.0	57.0	72.0	59.3

Table 3: Results of the retrieval task on the test dataset. R@10 and M@10 represent Recall@10 and MAP@10, respectively.

Method	E_s	C_s	OC_s	RePASs
BM25(passage-only)+Qwen2	0.762	0.248	0.227	0.580
BM25(rank fusion)+Qwen2	0.775	0.230	0.244	0.596
BM25(hybrid expert)+Qwen2	0.777	0.234	0.258	0.600

Table 4: Results of the answer generation task using RePASs on the test dataset. E_s, C_s, OC_s, and RePASs represent Entailment, Contradiction, Obligation Coverage, and RePAS score, respectively.

manage computational efficiency and model input constraints, we truncate both queries and passages to a maximum of 512 tokens.

3.2 Metrics

The Passage Retrieval's performance in the RI-RAG task is quantitatively assessed through recall@10, thereby enabling the answer-generation module to focus on refining the output. Furthermore, MAP@10 is implemented as a diagnostic tool to evaluate the precision of the ranking within the top-10 retrieved passages. The Answer Generation's performance is evaluated using the RePASs. This metric ensures the answer-generation module produces accurate and consistent responses by integrating three key components: the entailment score (E_s), the contradiction score (C_s), and Obligation Coverage Score (OC_s). The RePASs is calculated as:

$$RePASs = \frac{E_s - C_s + OC_s + 1}{3} \quad (3)$$

3.3 Results

The outcomes of the two Sub-Challenges are presented in Table 3 and Table 4, where our method demonstrates superior performance over alternative approaches. The fine-tuning of the LLaMa-7B model proves to be more effective than conventional retrieval techniques. When this is augmented with our hybrid expert mechanism, it creates a synergistic effect that harnesses the collective strengths of a variety of models. This includes those enriched with extensive data training and those with the straightforward efficiency of methods like BM25. Our hybrid approach transcends the limitations of individual experts or simple fusion by incorporating a sophisticated process that refines the output. This process is tailored to the subtleties of the task at hand, resulting in a system that is not only more accurate but also better at generalizing from the data. This comprehensive strategy leads to an enhanced overall performance, making it a more robust solution for the challenges presented by the RIRAG task.

4 Conclusion

In summary, this paper delves into the effective utilization of LLMs for regulatory-specific tasks and introduces a hybrid expert mechanism. This innovative approach marries the capabilities of sophisticated LLMs with the tried-and-true methods of traditional retrieval systems. The result is a significant boost in the efficacy of regulatory information retrieval and answer generation processes.

Moving forward, our research will focus on uncovering further optimization techniques for these large models and on broadening their application to a more diverse array of regulatory tasks.

5 Limitations

One limitation we encountered is that it does not explore the performance of the model on a broader range of benchmarks. This restraint may limit the generalizability assessment of the model's applicability to a broader spectrum of downstream tasks, which could be a subject for future work.

6 References

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6.

Thibault Formal, C. Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021. Splade v2: Sparse lexical and expansion model for information retrieval. *ArXiv*, abs/2109.10086.

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.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Patrick 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*, volume 33, pages 9459–9474. Curran Associates, Inc.

Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 2023. How to train your dragon: Diverse augmentation towards generalizable dense retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6385–6400, Singapore. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

OpenAI. 2022. Chatgpt. https://openai.com/blog/. Chatgpt, 2022. 2, 3, 8.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at trec-3. In *Text Retrieval Conference*.

Keshav Santhanam, O. Khattab, Jon Saad-Falcon, Christopher Potts, and Matei A. Zaharia. 2021. Colbertv2: Effective and efficient retrieval via lightweight late interaction. In *North American Chapter of the Association for Computational Linguistics*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

NUST Nova at RIRAG 2025: A Hybrid Framework for Regulatory Information Retrieval and Question Answering

Mariam Babar Khan, Huma Ameer, Seemab Latif, and Mehwish Fatima*
School of Electrical Engineering and Computer Science (SEECS),
National University of Sciences and Technology (NUST),
Islamabad, Pakistan
[mariamk.bscs21seecs,hameer.msds20seecs,
seemab.latif,mehwish.fatima]@seecs.edu.pk

Abstract

NUST Nova participates in RIRAG Shared Task, addressing two critical challenges: Task 1 involves retrieving relevant subsections from regulatory documents based on user queries, while Task 2 focuses on generating concise, contextually accurate answers using the retrieved information. We propose a Hybrid Retrieval Framework that combines graph-based retrieval, vector-based methods, and keyword matching (BM25) to enhance relevance and precision in regulatory QA. Using score-based fusion and iterative refinement, the framework retrieves the top 10 relevant passages, which are then used by an LLM to generate accurate, context-aware answers. After empirical evaluation, we also conduct an error analysis to identify our framework's limitations.

1 Introduction

The Regulatory Information Retrieval and Answer Generation (RIRAG) shared task focuses on advancing Question Answering (QA) in regulatory compliance. Participants develop systems to retrieve relevant information and generate precise answers to complex compliance queries, addressing the critical need for interpreting specialized regulatory language in domains like legal research and policy analysis.

Traditional Information Retrieval (IR) methods like BM25 (Robertson et al., 1994) excel at keyword-based document retrieval but struggle with the nuanced, context-dependent language of regulatory texts (de Andrade and Becker, 2023; Yang et al., 2023). Vector-based retrieval, leveraging document embeddings, shows promising results but faces challenges with domain-specific terminology and maintaining relevance (Monir et al., 2024; Sarmah et al., 2024). Similarly, graph-based retrieval excels in regulatory contexts but suffers from scalability issues and handling ambiguous or incomplete data (Jain et al., 2023; Technology, 2015; Sarmah et al., 2024). These limitations underscore the need for hybrid approaches to enhance precision and scalability in regulatory text retrieval (Sarmah et al., 2024).

To address the challenges of regulatory QA, we propose a Hybrid Retrieval Framework with multi-method scoring to enhance passage retrieval precision. The framework combines three models: (1) Neo4j, which structures queries and passages into a graph for initial relevance extraction, (2) BM25 for keyword matching, and (3) FAISS for ranking passages based on semantic similarity. These models are fused through score-based fusion, refining results by combining BM25 and FAISS outputs with those from Neo4j. This hybrid approach ensures accurate retrieval of the top 10 passages. Finally, the Llama model generates context-aware, regulatory-compliant answers, effectively handling domain-specific terminology and complex relationships.

2 Hybrid Retrieval Framework

Our hybrid retrieval[1] system integrates knowledge-graph and vector-based methods, combining their strengths to enhance accuracy and relevance. A score fusion mechanism merges relevance scores, followed by re-ranking to produce a balanced, high-quality ranked list. An LLM processes the top-ranked passages for context-aware, regulatory-compliant answers. Figure 1 illustrates this integration for efficient results.

2.1 Embeddings Generation

We use LegalBERT (LB) (Chalkidis et al., 2020) to generate dense embeddings for regulatory information retrieval and answer generation. LB provides domain-specific knowledge critical for understanding complex legal content as it is pre-trained on a large corpus of legal and regulatory texts. These

*Corresponding author: mehwish.fatima@seecs.edu.pk

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

Proceedings of the 31st International Conference on Computational Linguistics, pages 73–78
January 19–24, 2025. ©COLING 2025

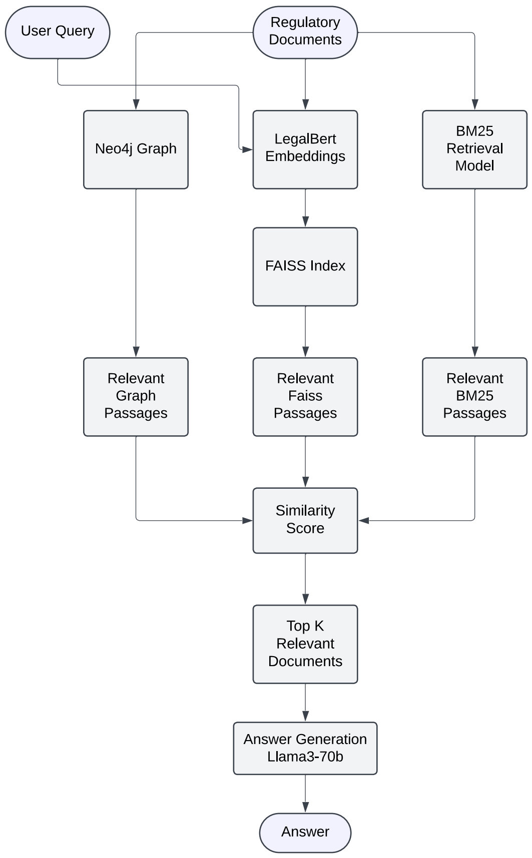

Figure 1: Hybrid Retrieval Framework: three retrieval models work simultaneously and then results are ranked before answer generation

embeddings, optimized for legal semantics, are stored for vector-based retrieval using FAISS(Douze et al., 2024) and enable efficient similarity computations.

2.2 Information Retrieval

For IR, we implement a hybrid approach that combines graph-based retrieval, vector-based retrieval, and traditional BM25.

2.2.1 Graph-Based Retrieval

We use Neo4j Graph Database (NEO4J) (Technology, 2015) to enhance the retrieval of relevant passages based on the structural relationships within the data. We query our NEO4J database for a specific question by retrieving passages that are connected to the query's node through direct relationships in the graph. Using LB, the query text is converted into a numerical vector representation. Relevance is determined by calculating the cosine similarity between query and passage embeddings, with the top-ranked passages returned based on their similarity scores.

2.2.2 Statistical Retrieval

The tokenized dataset is indexed using BM25 (Robertson et al., 1994), a well-established ranking function in information retrieval that evaluates lexical overlap between the query and document passages. BM25 assigns relevance scores based on factors like term frequency, inverse document frequency, and query term saturation, effectively ranking passages by relevance. It ranks document passages based on their relevance to the query, and the top-ranked passages are retrieved for further processing.

2.2.3 Vector-Based Retrieval

For the vector database-based approach, we first generate high-quality vector embeddings for regulatory data using LB. These embeddings are indexed with FAISS for retrieval, where dense query embeddings, also generated by LB, are compared to pre-computed passage embeddings in the FAISS index using cosine similarity. This process efficiently retrieves the top-ranked passages, ensuring semantically accurate and relevant results.

2.3 Fusion and Re-ranking

To combine the results of NEO4J and BM25+FAISS, we use a score-based fusion approach. Initially, passages are retrieved independently by each method, with scores assigned based on their respective retrieval techniques. For graph-based retrieval, similarity scores are computed using cosine similarity between the query and linked passages' embeddings. BM25, a probabilistic model, calculates document relevance based on term frequency and inverse document frequency. In contrast, vector-based retrieval derives scores through approximate nearest-neighbor searches in the embedding space. The results from the mentioned methods are merged, eliminating duplicates and retaining the higher similarity score for overlapping passages. The combined passages are then re-ranked by recalculating their relevance using cosine similarity between the query embedding and the passage embeddings, ensuring that the most relevant passages, as identified by all retrieval methods, are ranked highest.

2.4 Answer Generation

The last step involves generating responses using the Llama3-70b model (LLAMA3) (Dubey et al., 2024). Passages retrieved from the Hybrid Retrieval Framework are concatenated and provided as context to LLAMA3, which generates a coherent and accurate response tailored to the user's query. LLAMA3 synthesizes information from the retrieved passages to produce precise and contextually rich output. By using its pre-trained knowledge and the input passages to generate responses, LLAMA3 maintains the nuance and formal tone required for regulatory language.

Prompt engineering is crucial in our pipeline, ensuring generated responses align with regulatory obligations and avoid contradictions. Clear instructions are provided to cover all key requirements, structure responses, and align with source sentences from retrieved passages, reducing hallucinations and maintaining factual consistency.

Although fine-tuning is not yet implemented, future iterations will focus on adapting LLAMA3 to regulatory documents, enhancing its understanding of domain-specific jargon, hierarchical clauses, and inter-references. This will improve the model's ability to generate precise, compliant answers.

By leveraging structured prompts and relevant passages, LLAMA3 minimizes hallucinations, focusing on the most relevant context for generating accurate, high-quality responses.

3 Experiments

3.1 Dataset

We use the given ObliQA dataset which includes three subsets: the train set contains 22,295 questions, the test set have 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. A comprehensive overview of the shared task and dataset are presented to Appendix C for brevity.

3.2 Baselines and Our Model

We consider BM25 (Gokhan et al., 2024) as our baseline due to its good performance. The top passages are processed by GPT-4-TURBO-1106 (GPT-4) (OpenAI, 2023) with a relevance threshold of 0.7, using a tailored prompt to generate compliance-focused answers that integrate regulatory requirements. We evaluate our final model and a variation of it: (i) NEO4J+BM25+FAISS, and (ii) BM25+FAISS. For answer generation, we use LLAMA3 to generate contextually accurate and coherent responses based on the retrieved passages.

3.3 Evaluation Metrics

We follow the standard metrics by organizers (Gokhan et al., 2024): MAP@10 and RECALL@10 for passage retrieval, and RePaSs (**Re**) for answer generation.

Model	RECALL@10	MAP@10
BM25	0.76	0.62
BM25+FAISS	0.58	0.29
NEO4J+BM25+FAISS	0.79	0.74
NEO4J+BM25+FAISS	0.39	0.23

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

4 Results

Table 1 presents the results of retrieval models for RECALL@10 and MAP@10. The proposed framework NEO4J+BM25+FAISS achieves the highest scores of RECALL@10 = 0.79 and MAP@10 = 0.74 by using Neo4j's graph structure for capturing structural relationships among documents, while BM25 and FAISS ensure precise term matching and semantic alignment. This demonstrates the efficacy of integrating diverse retrieval strategies to address the complexity of regulatory texts.

The BM25 model demonstrates strong performance with RECALL@10 = 0.76 and MAP@10 = 0.62, confirming its reliability in retrieving relevant passages in a regulatory context. Its focus on exact term matching makes it particularly effective for structured legal texts, though it is limited in handling complex semantic relationships.

The BM25 model performs well (RECALL@10 = 0.76, MAP@10 = 0.62), excelling in regulatory contexts with its focus on exact term matching but struggling with semantic complexity. The BM25+FAISS model underperforms (RECALL@10 = 0.58, MAP@10 = 0.29), as FAISS's semantic retrieval weakens precision, highlighting misalignment with BM25 in domain-specific tasks.

4.1 Answer Generation Metrics

Table 2 compares the performance of two baseline methods: BM25+GPT-4 passage-only (PO) and BM25+GPT-4 rank fusion (RF) against two hybrid approaches: NEO4J+BM25+FAISS+LLAMA3 and BM25+FAISS+LLAMA3. The baselines achieve high relevance scores ($\mathbf{E_S}$ = 0.77, 0.77) but decline in contextual accuracy ($\mathbf{C_S}$ = 0.24, 0.24) and open-ended query handling ($\mathbf{OC_S}$ = 0.22, 0.20), resulting in **Re** scores of 0.58 and 0.58. This high-

Models	$\mathbf{E_S}$	$\mathbf{C_S}$	$\mathbf{OC_S}$	$\mathbf{Re}$
BM25(PO)+GPT-4	0.77	0.24	0.22	0.58
BM25(RF)+GPT-4	0.77	0.24	0.20	0.58
BM25+FAISS+LLAMA3	0.31	0.25	0.07	0.37
NEO4J+BM25+FAISS+LLAMA3	0.43	0.36	0.15	0.41
NEO4J+BM25+FAISS+LLAMA3	0.36	0.31	0.11	0.39

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

lights the limitations of keyword-based retrieval for nuanced regulatory queries.

NEO4J+BM25+FAISS+LLAMA3 shows moderate performance ($\mathbf{E_S}$ = 0.43, $\mathbf{C_S}$ = 0.36, $\mathbf{OC_S}$ = 0.15, $\mathbf{Re}$ = 0.41). Its graph-based integration improves semantic retrieval but struggles with open-ended queries. BM25+FAISS+LLAMA3 underperforms, with low relevance ($\mathbf{E_S}$ = 0.31), moderate contextual accuracy ($\mathbf{C_S}$ = 0.25), and poor open-ended query handling ($\mathbf{OC_S}$ = 0.07), yielding a $\mathbf{Re}$ score of 0.37. This highlights that vector-based retrieval alone is inadequate for regulatory QA without structured graph-based methods.

These results show that baseline models excel in relevance but struggle with contextual accuracy and open-ended queries. Hybrid methods improve structured retrieval via graph-based techniques but require optimization to balance relevance and adaptability for regulatory QA.

4.2 Error Analysis

We conduct an in-depth error analysis on 446 unseen questions to identify Hybrid Retrieval Framework's limitations in Appendix A. For this purpose, we apply a multi-step approach to evaluate the performance and quality of the responses generated by the model.

4.2.1 Data Preprocessing

First, we process the dataset by categorizing the questions based on whether an answer was generated or not. We split questions with empty answers and those with generated answers into two groups. we then preprocess data by tokenizing and filtering out stopwords to ensure the format suitable for analysis.

4.2.2 Topic Modeling

To explore further, we apply topic modeling using Latent Dirichlet Allocation (LDA) to identify prevalent themes in both groups of questions. This allow us to analyze the distribution of topics within the questions with empty answers and with generated answers. We evaluate these results to get insights

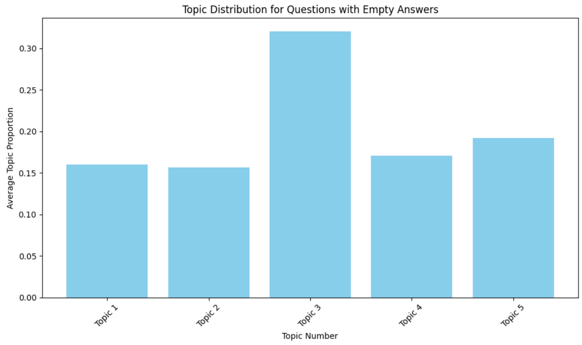

Figure 2: Topic Distribution of Questions with Empty Answers

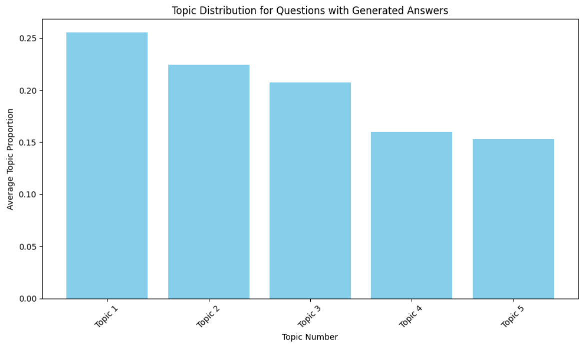

Figure 3: Topic Distribution of Questions with Generated Answers

about quality and relevance of the generated answers. LDA reveals distinct patterns in topic distributions illustrated in Figures 2 and 3. The details of these topics are presented in Appendix 4.2.2.

5 Conclusion

This work presents a hybrid framework combining vector-based, graph-based, and keyword-matching techniques to enhance regulatory information retrieval and answer generation. The approach significantly improves relevance and contextual accuracy, especially in handling domain-specific content. Preliminary results show improvements over baseline methods, with promising retrieval performance. However, answer generation results require refinement, highlighting the need for further enhancement. Future work includes exploring different LLMs or fine-tuning them for regulatory data and incorporating summarization techniques to optimize answer generation and extending graph-based retrieval to operate on entire documents rather than individual passages.

Acknowledgments

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

References

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. Legal-bert: The muppets straight out of law school.

Leonardo de Andrade and Karin Becker. 2023. Bb25hlegalsum: Leveraging bm25 and bert-based clustering for the summarization of legal documents. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 255–263.

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*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Deepali Jain, Malaya Dutta Borah, and Anupam Biswas. 2023. Bayesian optimization based score fusion of linguistic approaches for improving legal document summarization. *Knowledge-Based Systems*, 264:110336.

Solmaz Seyed Monir, Irene Lau, Shubing Yang, and Dongfang Zhao. 2024. Vectorsearch: Enhancing document retrieval with semantic embeddings and optimized search. *arXiv preprint arXiv:2409.17383*.

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

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 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).

Bhaskarjit Sarmah, Dhagash Mehta, Benika Hall, Rohan Rao, Sunil Patel, and Stefano Pasquali. 2024. Hybridrag: Integrating knowledge graphs and vector retrieval augmented generation for efficient information extraction. In *Proceedings of the 5th ACM International Conference on AI in Finance*, pages 608–616.

Inc Technology. 2015. Neo4j, the world's leading graph database. *Neo4j Graph Database*.

Fangkai Yang, Pu Zhao, Zezhong Wang, Lu Wang, Jue Zhang, Mohit Garg, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. 2023. Empower large language model to perform better on industrial domain-specific question answering. *arXiv preprint arXiv:2305.11541*.

A Limitations

The graph-based approach in the framework faces scalability challenges, as managing and querying large, dynamic regulatory datasets can become resource-intensive, leading to slower retrieval times and higher computational costs. Additionally, the reliance on pre-trained models like BM25, FAISS, and Neo4j, while effective, limits adaptability to the nuanced language of the regulatory domain, affecting precision in handling domain-specific variations. Analysis of questions with empty answers reveals the system's strength in addressing specialized queries, with topics such as "adgm", "compliance", and "regulations" highlighting its focus on financial and regulatory concepts. However, for broader or less specific inquiries, the system struggles to maintain relevance, as indicated by topics like "customer" and "business". This highlights a gap in handling ambiguous or general questions, suggesting the need for enhanced contextual interpretation to improve performance across diverse query types.

B Training Considerations

Our framework avoids custom training, using pretrained retrieval techniques for efficiency. This eliminates the need for resource-intensive model training while maintaining strong relevance for regulatory QA tasks.

C Task and Data

The RIRAG shared task comprises two key components: *Task 1: Regulatory Information Retrieval* focuses on retrieving relevant passages from complex, domain-specific regulatory documents in response to user queries, and *Task 2: Regulatory Answer Generation* involves producing concise, accurate answers based on the retrieved passages. Together, these tasks aim to advance the development of models that improve the accuracy and reliability of systems addressing complex regulatory queries.

The ObliQA dataset (Gokhan et al., 2024) advances Regulatory NLP (REGNLP) by providing 40 structured regulatory documents from Abu Dhabi Global Markets (ADGM), governing financial services in UAE free zones. With subsections, numbered clauses, and cross-references, it is well-suited for compliance applications. Converted to JSON format, the dataset is validated using the DEBERTA-V3-XSMALL model (He et al., 2021) across three classes: Entailment, Contradiction, and Neutral.

D Error Analysis

Topic modeling on questions with empty answers revealed distinct themes. **Topic 1** encompassed terms like "risk", "person", "authorised", "adgm", and "management", reflecting a focus on risk and authorization processes in the ADGM context. **Topic 2** highlighted words such as "provide", "could", "specific", "risk", and "requirements", indicating queries related to precise regulatory risks and compliance criteria. **Topic 3** emphasized "virtual", "assets", "specific", "adgm", and "requirements", underscoring questions about virtual asset regulations. Similarly, **Topic 4** involved "could", "requirements", "guidance", "risk", and "adgm", pointing to inquiries about regulatory guidance. Lastly, **Topic 5** featured terms like "regulator", "person", "rule", "adgm", and "reporting", focusing on reporting standards and regulatory rules. These themes provide insights into gaps in the system's ability to generate answers and highlight areas for enhancement.

Topic modeling on questions with generated answers revealed five distinct themes. **Topic 1** was characterized by terms such as "compliance", "reporting", "virtual", "must", and "adgm", indicating a focus on regulatory compliance and mandatory reporting requirements. **Topic 2** featured terms like "provide", "information", "customer", "business", and "could", suggesting queries related to customer or business-specific information needs. **Topic 3** emphasized "financial", "risk", "must", "person", and "authorised", highlighting themes around financial risk and regulatory authorizations. **Topic 4** included terms such as "financial", "treatment", "standards", "per", and "could", reflecting inquiries about financial treatment and adherence to standards. Lastly, **Topic 5** was defined by terms like "adgm", "risk", "reporting", "person", and "regulations", focusing on risk management and regulatory reporting within the context of the Abu Dhabi Global Market (ADGM). These topics collectively provide insights into the nature of questions for which the system successfully generated answers.

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

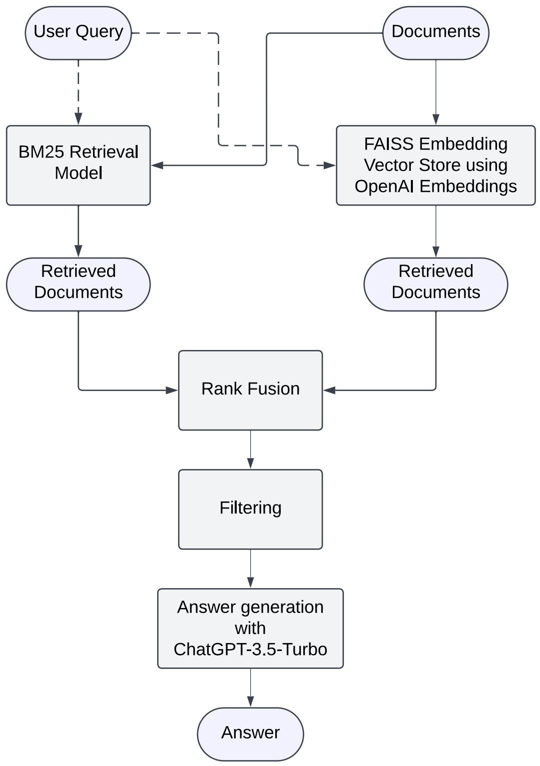

Figure 1: Architecture Diagram of FusionRAG

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 trade-off 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.

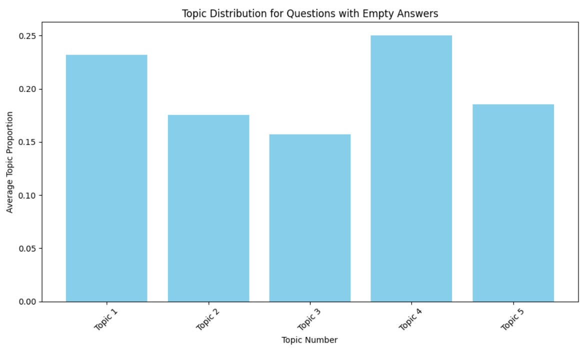

Figure 2: Topic distribution for questions with empty answers.

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*.

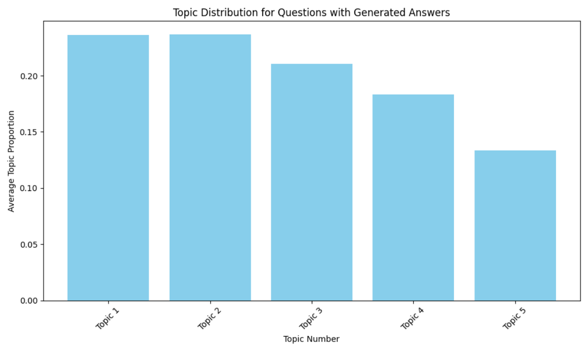

Figure 3: Topic distribution for questions with generated answers.

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

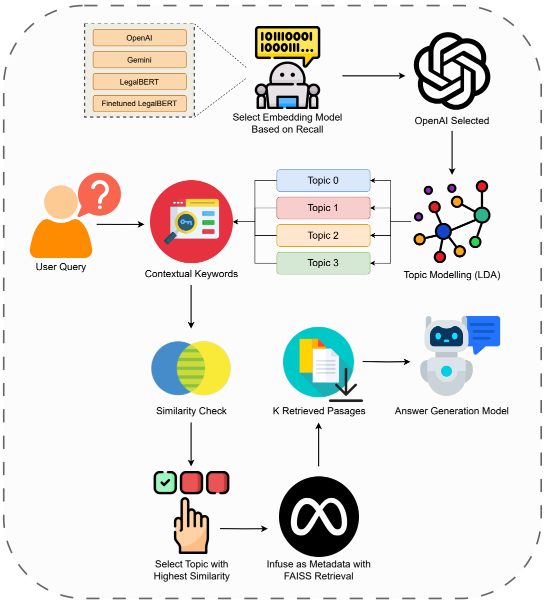

Figure 1: Progressive Retrieval Pipeline (ProReg)

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.

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.

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.

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

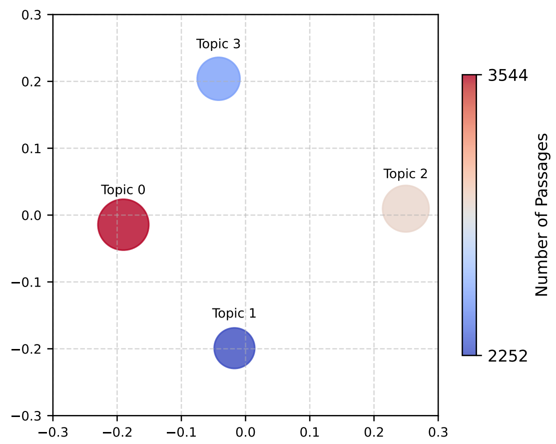

Figure 2: LDA Based Topic Modeling

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.

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 $\mathcal{T} = \{T_1, T_2, \ldots, T_k\}$;
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_{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:

$$S(T_i, q) = \frac{{}_{k \in \mathcal{K}_{T_i} \cap \mathcal{K}_q} w_k^{(T_i)} \cdot w_k^{(q)}}{\overline{{}_{k \in \mathcal{K}_{T_i}} \left(w_k^{(T_i)}\right)^2} \cdot \overline{{}_{k \in \mathcal{K}_q} \left(w_k^{(q)}\right)^2}}$$

  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 $\mathcal{C}_{\text{relevant}}$

Enhancing Regulatory Compliance Through Automated Retrieval, Reranking, and Answer Generation

Kübranur Umar*
TOBB University of Economics and Technology
kumar@etu.edu.tr

Hakan Doğan*
TOBB University of Economics and Technology
hakandogan@etu.edu.tr

Onur Özcan*
TOBB University of Economics and Technology
onurozcan@etu.edu.tr

İsmail Karakaya
HAVELSAN Inc.
ikarakaya@havelsan.com.tr

Alper Karamanlıoğlu
HAVELSAN Inc.
alperk@havelsan.com.tr

Berkan Demirel
HAVELSAN Inc.
bdemirel@havelsan.com.tr

Abstract

This paper explains a Retrieval-Augmented Generation (RAG) pipeline that optimizes regularity compliance using a combination of embedding models (*i.e.* bge-m3, jina-embeddings-v3, e5-large-v2) with reranker (*i.e.* bge-reranker-v2-m3). To efficiently process long context passages, we introduce *context aware chunking* method. By using the RePASS metric, we ensure comprehensive coverage of obligations and minimizes contradictions, thereby setting a new benchmark for RAG-based regulatory compliance systems. The experimental results show that our best configuration achieves a score of **0.79** in Recall@10 and **0.66** in MAP@10 with LLaMA-3.1-8B model for answer generation.

1 Introduction

Regulatory documents are critical components for many industries including finance, healthcare and insurance, to comply with standards and laws. These documents are characterized by complex legal terminology, hierarchical structures, and frequent updates. Therefore, this creates difficulties for interpretation and implementation. These incompatibilities lead to negative outcomes such as significant financial penalties, loss of reputation, and operational disruptions.

The complexity of regulatory documents to put forward the necessity for advanced systems capable of efficient information retrieval and synthesis. Retrieval-Augmented Generation (RAG) systems offers a promising solution for retrieval mechanism and answer generation.

Previous research in Regulatory Natural Language Processing (RegNLP) discovered the potential of machine learning for automating regulatory compliance, but some difficulties still exist:

1. High-precision retrieval of relevant passages from large regulatory corpora is challenging.
2. Ranking and synthesizing retrieved passages to ensure completeness and scope of obligation is another challenge.
3. Efficient processing of long contextual queries where relevant information may span multiple sections of a document, is another major challenge.

In this study, to address these challenges, we propose an optimized RAG pipeline for advanced-level ranking and improved generative performance, using a context-aware chunking strategy combined with *"bge-m3 + hybrid search"*, and *"bge-reranker-v2"*. Our contributions are as follows:

- Introducing a chunk-based approach for processing long regulatory contexts effectively.
- Evaluation of multiple retrieval and re-ranking models for regulatory QA tasks using the RePASS metric.

2 Related Work

The significant progress on RegNLP mostly about complexities of regulatory texts. The structured data extraction is focused by the previous studies. In this context, Lau et al. (2005) focus on XML-based frameworks in order to extract information from accessibility regulations. Also, Kiyavitskaya et al. (2008) propose the Cerno framework to focus on automation of rights and obligation extraction

*These authors contributed equally to this work

Proceedings of the 31st International Conference on Computational Linguistics, pages 91–96
January 19–24, 2025. ©COLING 2025

from legal texts. However, these works are insufficient in scalability and adaptability.

Thanks to the advent of deep learning, considerable improvements achieved in RegNLP. In this context, Chalkidis et al. (2018) introduce a hierarchical BiLSTM model in order to extract obligations from legal contracts, and this study outperforms the previous methods that relies on manual features. In addition, Nair et al. (2018) implement deep learning pipelines in the work of annotating global trade regulations. This method enables enhanced compliance workflows in the field of RegNLP. Similar to these works, Chalkidis et al. (2021) leverage BERT-based models to handle complex queries in EU/UK legislative texts. This method shows how transformer architecture is effective in processing long documents. Abualhaija et al. (2022) extend this method with BERT for automated question-answering (QA) systems targeting GDPR-related texts. Thanks to this work, a considerable success has achieved in passage retrieval tasks.

Gokhan et al. (2024) provide a baseline framework for regulatory QA tasks by introducing the ObliQA dataset[1] curated to address multi-passage queries. This dataset is as collection of over 27,000 QA pairs derived from Abu Dhabi Global Markets[2] regulations. Additionally, this study introduces Regulatory Passage Answer Stability Score (RePASS), a novel evaluation metric designed to measure the accuracy and consistency of generated answers in regulatory contexts. They combine sparse and dense retrieval methods (*e.g.*, BM25 and BGE models) with a generative approach to synthesize answers from retrieved passages. Despite the contribution of this work, they challenged in handling complex or lengthy queries, and the generative model exhibited limitations in contextual comprehension and obligation coverage.

RegNLP applications accelerated by the recent advancements in synthetic data generation. An example of these upgrades is QA dataset for roundtrip validation in Alberti et al. (2019). Also, Maatouk et al. (2023) propose zero-shot learning method for neural passage retrieval.

The integration of retrieval and generative models enables advanced QA methodologies in RAG systems. Lewis et al. (2020) formalize RAG as a framework that enriches generative models with retrieved knowledge. The retrieval efficiency and response quality is improved by Self-RAG (Asai et al. (2023)) and PipeRAG (Jiang et al. (2024)) systems having limited adaptation to regulatory texts.

Our study addresses challenges in retrieval precision and generative accuracy for regulatory QA. We introduce a robust RAG pipeline incorporating hybrid retrieval using dense models, advance re-ranking and context-aware chunking to manage long regulatory documents. This system achieves a Recall@10 of **0.79** and MAP@10 of **0.66**, establishing a new standard for regulatory question answering.

3 Methodology

The long passages in regulatory documents affect the performance of generative models, since processing extended contexts efficiently is a challenge for these models. In order to handle this challenge, we segment long passages into smaller chunks, then filter and re-rank to optimize the input for generative models. In the next sections, we describe the proposed methodology by explaining the retrieval pipeline, long-context processing techniques, and answer generation system. The demonstration of our pipeline is shown in Figure 1.

3.1 Retrieval Pipeline

Combination of retrieval and re-ranking models in the retrieval pipeline maximizes the recall and precision. This system ensures that the most relevant passages are prioritized for downstream processing.

3.1.1 Passage Retrieval

In the first retrieval stage, we experiment with multiple dense retrieval models, including **bge-m3**, **jina-embeddings-v3**, and **e5-large**. According to the obtained results, the **bge-m3** model outperforms other models and achieves Recall@10 of **0.74**. The results are detailed in Table 1. This results show that the model is suitable for regulatory texts, thanks to its ability to effectively capture semantic nuances.

In order to improve retrieval performance, a hybrid search mechanism is implemented combining dense (vector-based) and lexical retrieval methods. We achieve the best recall by tuning the hybrid search parameter to **0.3**. Top-50 passages are retrieved for each query that serves as input for the re-ranker model.

[1] https://github.com/RegNLP/ObliQADataset
[2] https://www.adgm.com/

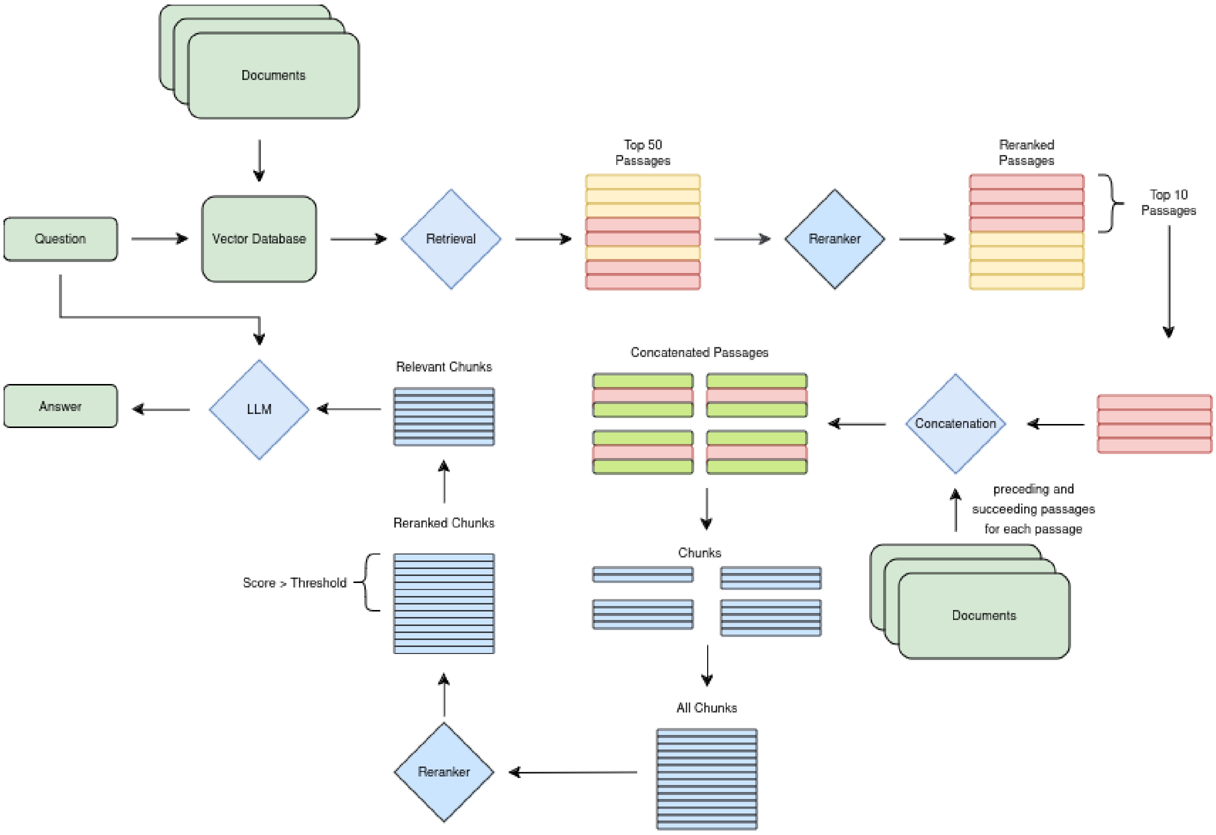

Figure 1: Pipeline Diagram: Retrieval, Reranking, and Generative Model Integration.

	w=0.5		w=0.3	
Model	**Recall@10**	**MAP@10**	**Recall@10**	**MAP@10**
jina-embeddings-v3	0.71	0.60	0.76	0.62
e5-large-v2	0.69	0.59	0.76	0.62
bge-m3	0.74	0.61	0.76	0.62
bge-m3+bge-reranker-v2-m3	0.77	0.65	**0.79**	**0.66**

Table 1: Retrieval and Reranking Performance.

3.1.2 Passage Re-ranking

We evaluate the **bge-reranker-v3-m3** model to improve the ranking of retrieved passages. We achieve the highest performance, with a **0.79** Recall@10, by the combination of **bge-m3** and **bge-reranker-v2**, when the hybrid search hyperparameter is set to **0.3**. In contrast, when the hybrid parameter is set to **0.5**, the Recall@10 value is **0.77**. These results indicate the importance of hyperparameter optimization in achieving high retrieval performance.

The re-ranker assigns points to retrieved passages and prioritizes the scope of obligation and relevance. The top 10 highest-scoring passages, according to their scores, are selected for further processing, which significantly improves the quality of inputs to the generative model.

3.2 Long Context Processing

Regulatory queries usually require synthesizing information span over more than one section. To handle this difficulty, we use a strategy that contains context expansion, chunking, and chunk filtering and re-ranking. This strategy is detailed below:

1. **Context Expansion**: Retrieved passages are enriched by their preceding and succeeding sections. This additional context improves the system's ability to address cross-referenced information and capture narrow regulatory obligations.

2. **Chunking**: Expanded passages are divided into smaller chunks in accordance with the input limitations of generative models *e.g.* **LLaMA-3.1-8B-Instruct**, with a maximum

Method	Es	Cs	OCs	RePASs
bge-m3+bge-reranker-v2-m3+LLaMA-3.1-8B-Instruct	0.39	0.30	0.12	0.41

Table 2: RePAS Scores

length of 1024 tokens per chunk and a stride value of 100 tokens. This operation proposes efficient processing while protecting critical regulatory information.

3. **Chunk Filtering and Re-ranking**: Each 1024-token segment provided as input to the model is processed through the **bge-reranker-v2-m3** model to enhance its performance and efficacy. The re-ranking process prioritizes chunks that are both relevant and contextually comprehensive, resulting in improved generative performance. The reranker model filters out less relevant chunks. This process reduces noise in the input data and ensures the generative model focuses on the most critical regulatory information. This streamlined input allows the model to generate more precise, contextually aligned, and reliable outputs, ultimately enhancing the accuracy and utility of the system for regulatory question-answering tasks.

   Re-ranker scores chunks for relevance and contextual completeness. Chunks that exceed a certain threshold are given as input to the model. In order to select high-quality chunks, a threshold score of 0.7 is applied. The threshold is reduced incrementally by 0.1 until at least one chunk meets the criteria because of some cases where all chunks score below this threshold. This process ensures that there are always input data for the generative model to process. By this way, the pipeline ensures that the generative model processes only the most relevant and high-quality parts, reducing noise and improving response accuracy.

3.3 Answer Generation

Filtered parts are given to the **LLaMA-3.1-8B-Instruct** model along with the query, and responses are generated. The generative model is used with one-shot prompt showed in Figure A.1 of Appendix A. The model is configured with the parameter `max_new_tokens` set to **512**. This allows the model to produce short but comprehensive answers. By presenting only the most relevant parts to the generative model, we ensure that its output is context-appropriate and compliant with regulatory requirements.

To evaluate the system performance, we use the **ObliQA dataset** which is introduced by Gokhan et al. (2024). This dataset consists of 27.869 QA pairs collected from financial regulations and provides a robust benchmark for regulatory QA systems. The evaluation is performed using the **RePASS metric** proposed by Gokhan et al. (2024). This metric evaluates obligation coverage, contradiction avoidance, and overall entailment. By using these tools, we ensure that it is compatible with the standards set for regulatory QA. The evaluation results, including RePASS values, are presented in detail in Table 2.

This methodology systematically addresses regulatory QA challenges by combining advanced retrieval techniques, efficient long-context processing capabilities, and careful filtering of inputs for generative models. By optimizing each stage of the pipeline, we demonstrate significant improvements in both retrieval accuracy and response quality.

4 Conclusion

This paper proposes RAG pipeline for regulatory compliance tasks by integrating hybrid search, advanced re-ranking and context-aware chunking strategies. RAG performance improved precision and recall significantly by using **bge-m3** model for hybrid search and **bge-reranker-v2** model for re-ranker. According to the experimental results, we achieve a score of **0.79** at Recall@10 and **0.66** at MAP@10. Introducing the chunk-based processing approach enhanced the **LLaMA-3.1-8B-Instruct** model's generative capabilities and enabled more effective processing of long-context regulatory documents.

References

Sallam Abualhaija, Chetan Arora, Amin Sleimi, and Lionel C Briand. 2022. Automated question answering for improved understanding of compliance requirements: A multi-document study. In *2022 IEEE 30th international requirements engineering conference (RE)*, pages 39–50. IEEE.

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic qa corpora generation with roundtrip consistency. *arXiv preprint arXiv:1906.05416*.

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

Ilias Chalkidis, Ion Androutsopoulos, and Achilleas Michos. 2018. Obligation and prohibition extraction using hierarchical rnns. *arXiv preprint arXiv:1805.03871*.

Ilias Chalkidis, Manos Fergadiotis, Nikolaos Manginas, Eva Katakalou, and Prodromos Malakasiotis. 2021. Regulatory compliance through doc2doc information retrieval: A case study in eu/uk legislation where text similarity has limitations. *arXiv preprint arXiv:2101.10726*.

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*.

Wenqi Jiang, Shuai Zhang, Boran Han, Jie Wang, Bernie Wang, and Tim Kraska. 2024. Piperag: Fast retrieval-augmented generation via algorithm-system co-design. *arXiv preprint arXiv:2403.05676*.

Nadzeya Kiyavitskaya, Nicola Zeni, Travis D Breaux, Annie I Antón, James R Cordy, Luisa Mich, and John Mylopoulos. 2008. Automating the extraction of rights and obligations for regulatory compliance. In *Conceptual Modeling-ER 2008: 27th International Conference on Conceptual Modeling, Barcelona, Spain, October 20-24, 2008. Proceedings 27*, pages 154–168. Springer.

Gloria T Lau, Kincho H Law, and Gio Wiederhold. 2005. Legal information retrieval and application to e-rulemaking. In *Proceedings of the 10th international conference on Artificial intelligence and law*, pages 146–154.

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.

Ali Maatouk, Fadhel Ayed, Nicola Piovesan, Antonio De Domenico, Merouane Debbah, and Zhi-Quan Luo. 2023. Teleqna: A benchmark dataset to assess large language models telecommunications knowledge. *arXiv preprint arXiv:2310.15051*.

Rahul Nair, Killian Levacher, and Martin Stephenson. 2018. Towards automated extraction of business constraints from unstructured regulatory text. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 157–160.

A Appendix

One-Shot Prompt

You are a knowledgeable assistant specialized in regulatory documents. Your task is to generate precise answers to questions based on the given context.

Example:

Question: *What are the specific approval requirements and criteria set forth by the ADGM for a Shari'a Supervisory Board's endorsement of a Public Fund's Constitution and Prospectus?*
Context :

- *The Fund Manager of a Domestic Fund that is a Public Fund must ensure that its Fund's Constitution and Prospectus are, and remain, approved by the Fund's Shari'a Supervisory Board.*
- *The Fund Manager of an Exempt Fund must ensure that the Fund's Constitution and Prospectus are, and remain, approved by the Fund Manager's Shari'a Supervisory Board.*

Answer: *The ADGM requires that the Constitution and Prospectus of a Public Fund be approved by a Shari'a Supervisory Board. For Domestic Public Funds, this approval must come from the Fund's Shari'a Supervisory Board, whereas for Exempt Funds, it is the Fund Manager's Shari'a Supervisory Board that must provide and maintain this approval.*

Now, answer the following question based on the given context:

Question: Question Text
Context : Relative Chunks
Answer:

Figure A.1: One-Shot Prompt Representation.

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 **L** is constructed as follows:

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

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.

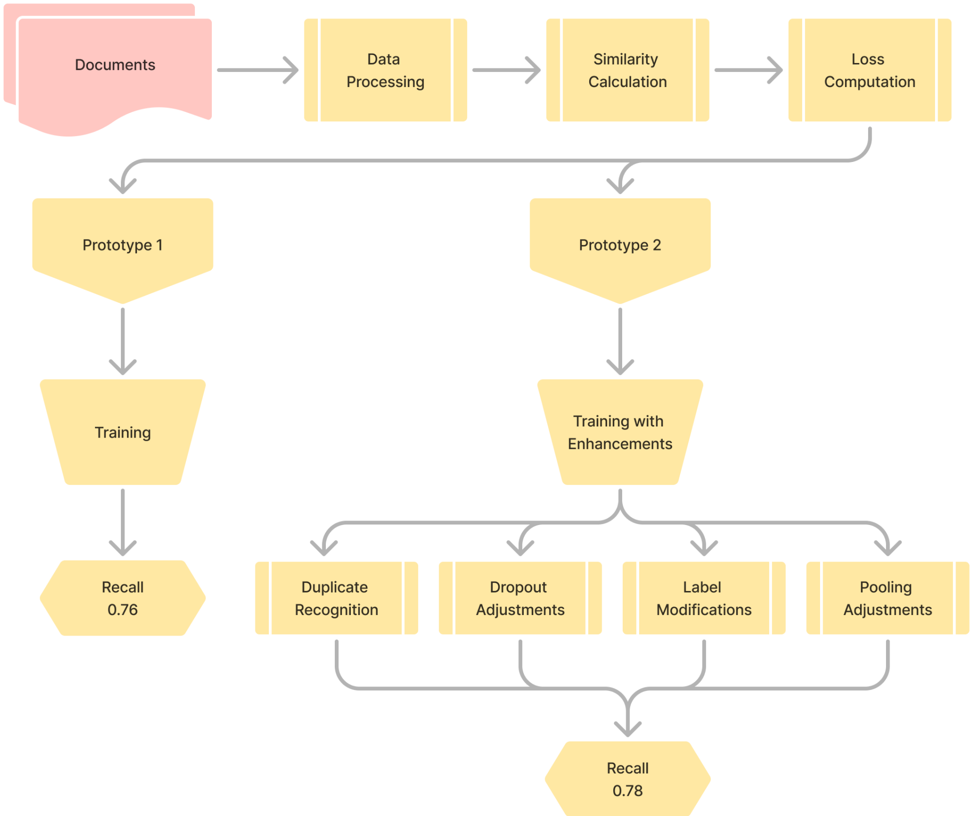

Figure 1: Proposed framework.

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 **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.

Regulatory Question-Answering Using Generative AI

Devin Quinn[1]*, Sumit Pai[2]*, Nirmala Pudota[2]*, Iman Yousfi[1], Sanmitra Bhattacharya[1]
[1]Deloitte & Touche LLP, United States
[2]Deloitte & Touche Assurance & Enterprise Risk Services India Private Limited, India

Abstract

Although retrieval augmented generation (RAG) has proven to be an effective approach for creating question-answering systems on a corpus of documents, there is a need to improve the performance of these systems, especially in the regulatory domain where clear and accurate answers are required. This paper outlines the methodology used in our submission to the Regulatory Information Retrieval and Answer Generation (RIRAG) shared task at the Regulatory Natural Language Processing Workshop (RegNLP 2025). The goal is to improve document retrieval (Shared Task 1) and answer generation (Shared Task 2). Our pipeline is constructed as a two-step process for Shared Task 1. In the first step, we utilize a text-embedding-ada-002-based retriever, followed by a RankGPT-based re-ranker. The ranked results of Task 1 are then used to generate responses to user queries in Shared Task 2 through a prompt-based approach using GPT-4o[1]. For Shared Task 1, we achieved a recall rate of 75%, and with the prompts we developed, we were able to generate coherent answers for Shared Task 2.

1 Introduction

Regulations are official rules and directives established and maintained by authoritative bodies, such as government or regulatory agencies, to ensure compliance with legal standards. They are crucial for maintaining order, protecting public interests, and fostering fair practices across various industries. Due to the extensive range of regulations and the intricate nature of the language used in the regulatory content, comprehending these guidelines can be challenging for both the general public and regulatory professionals. Failure to adhere to regulations can result in legal and financial consequences, adversely impacting an organization's reputation and operations.

The RIRAG shared task (Gokhan et al., 2024) aims to improve the efficiency and accuracy of compliance-related tasks within the regulatory domain by encouraging the development of advanced Information Retrieval (IR) and answer generation techniques. When presented with a regulatory question, the main objective is to extract relevant passages from a vast collection of regulatory documents from Abu Dhabi Global Markets (ADGM)[2], which oversees financial services in the UAE's free economic zones. These extracted passages are then used to generate coherent and contextually accurate responses to the queries. The details of this dataset are described in Section 3.

In our submission, we address both shared tasks: Passage Retrieval (Subtask 1) and Answer Generation (Subtask 2). Our system design is presented in Section 4. Evaluation results on the development and test set is shown in Section 5. Finally, we conclude and discuss the next steps in Section 6.

2 Related Work

Early efforts in RegNLP concentrated on pattern matching, rule-based and semantic relation extraction methods. However, devising these patterns and rules can be quite difficult due to the complex nature and style of regulatory texts. Traditional information retrieval methods such as Best Matching-25 (BM25) and Term Frequency-Inverse Document Frequency (TF-IDF) have been widely used for regulatory information retrieval (Rosa et al., 2021)(Lau et al., 2003). But, these methods often struggle with shifts in word distribution and fail to adequately capture semantic similarity between words. Regulatory information retrieval using modern machine learning approaches (Ash and Chen, 2017; Tang et al., 2016; Collarana et al., 2018) such as word/document embeddings, Recurrent neural networks (RNN), and Long Short Term Memory

*These authors contributed equally to this work

[1]GPT-4o was selected based on performance as demonstrated on the HELM leaderboard

[2]https://www.adgm.com

Proceedings of the 31st International Conference on Computational Linguistics, pages 102–106
January 19–24, 2025. ©COLING 2025

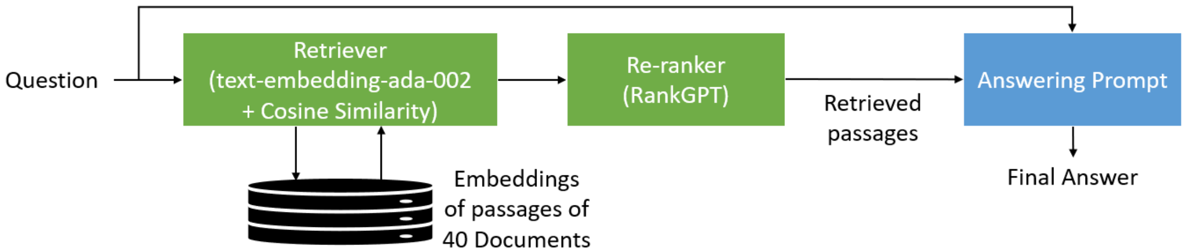

Figure 1: System Architecture

networks (LSTM) are good at modeling language and recognizing semantic similarities among various words and passages. However, they fall short in capturing long-range dependencies. Transformer-based approaches have demonstrated notable advancements in retrieval performance, as highlighted by several studies relevant to regulatory retrieval (Louis and Spanakis, 2021; Schumann et al., 2022) (Su et al., 2024). However, they still need significant amount of annotated datasets for fine-tuning.

With the advent of Large Language Models (LLMs), we can now accomplish a diverse range of NLP-related tasks without requiring task-specific fine-tuning. Ranking is one such area where these models particularly excel. RankGPT (Sun et al., 2023) proposes a list-wise ranking methodology (Ma et al., 2023; Pradeep et al., 2023) that addresses the issue of the LLM's context length by employing a sliding-window technique. Passages are ranked within each window, which then shifts incrementally to cover the entire list, ensuring overall re-ranking while staying within the LLM's context length limitation. For the task of answer-generation in regulatory domain, LSTM and transformer-based architectures have demonstrated significant results as highlighted by Collarana et al. 2018 (Zhong et al., 2020), but they require domain-specific labeled data for training. Recently, LLMs, through prompt engineering (Liu et al., 2021; Reynolds and McDonell, 2021) and RAG (Lewis et al., 2021) have shown promising results in generating coherent and grounded answers in the provided context.

In our paper, we use text-embedding-ada-002 from OpenAI for the retrieval followed by RankGPT for re-ranking, the details of which are elaborated in Section 4. The retrieved passages are then used as context within an engineered prompt to generate answers to the user query using GPT-4o.

3 Dataset

The Obligation-Based Question Answering (ObliQA) dataset (Gokhan et al., 2024), was specifically developed to support research in regulatory compliance. It includes question-answer pairs derived from passages in regulatory documents provided by the ADGM financial authority. These passages were selected individually or identified through topic-based clustering. Question-answer pairs were generated utilizing the GPT-4 model. To maintain precision and relevance, the generated questions were meticulously filtered for strong semantic alignment with the corresponding passages. Based on the number of passages used to generate the answers, the dataset was categorized into groups, where each group contains different combinations of 1 to 6 input passages per question. The data was subsequently divided into training, development, and test sets, containing 22k, 2.7k, and 2.7k samples, respectively.

4 System Design

The combined architecture of our system for both subtasks is shown in Figure 1 and detailed in the following sub-sections.

4.1 Subtask 1: Passage Retrieval

The components highlighted in green correspond to Subtask 1. We use text-embedding-ada-2 to embed the passages of ObliQA dataset and a standard vector database with a cosine similarity retriever. When a query is presented, we embed the query with the same embedding model and compute cosine similarity between the query embedding and the passage embeddings to retrieve the 30 most semantically similar passages. These initially retrieved passages are then input into RankGPT, which functions as a re-ranker as explained in Section 2, to reorder the passages and return its top 10. The final rankings reported for Subtask 1 were derived from this process.

As we developed our approach, we experimented with several different embedding models/techniques for RAG. Due to resource and time constraints, we limit our exploration to a few relatively small models, but of various sizes (ranging from 100M params to 8b params) and embedding dimension sizes (ranging from 768 to 4096). We evaluate base and fine-tuned versions of all-mpnet-base-v2, legal-bert-base-uncased, and Qwen2.5-1.5B-Instruct on a small subset of test data, but ultimately choose text-embedding-ada-2 as it showed the best performance metrics on it.

4.2 Subtask 2: Answer Generation

We design and iteratively improve a prompt that optimizes RePaSs. This prompt incorporates the user query and the passages retrieved from Subtask 1, generating contextually grounded answers (the blue component in Figure 1). This prompt is outlined in Table 3 and is used with GPT-4o to generate the relevant answers.

5 Results

To evaluate the effectiveness of our methodology, we first establish a baseline retrieval using text-embedding-ada-2 for comparison which only returns the initial top-30 results without any re-ranking. For an initial qualitative analysis, we use a small subset (10%) of the test data. The baseline system achieved 70% recall@10, while re-ranking using RankGPT demonstrated a 5% improvement over this. An example of this improvement is shown in Table 4, where RankGPT successfully re-ranked a ground-truth reference passage that was initially not within the top 10, which the baseline retriever missed. This passage provided critical context for a comprehensive answer, which our method captured accurately, unlike the baseline. Encouraged by these initial qualitative results, we proceed to conduct evaluations on the full datasets.

The performance of our method on the full development and test sets for Subtask 1 is summarized in Table 1. Our approach gets a good recall and mean average precision (MAP). A group-wise and passage-wise analysis reveals that model performance diminishes as complexity rises. Specifically, recall@10 and MAP@10 scores are high for the retrieval of single passages; however, these metrics decline as the number of passages to be retrieved increases. This trend is consistent across different groups, indicating that the model's ability to effectively retrieve and rank passages declines as the quantity of relevant passages grows.

On Subtask 2, we evaluate the RePASs metric as described in Gokhan et al. 2024. Our prompts achieve a high entailment score (E_S), indicating that the answers are well-supported by the source passages. However, performance on obligation coverage (OC_S) is comparatively lower. The Overall Composite Score remains consistent across both datasets and is relatively good, as shown in Table 2 (comparable to the performance of the best models (Gokhan et al., 2024) on evaluation dataset).

6 Conclusion and Future Work

Our retrieval followed by re-ranking methodology demonstrates consistent and relatively good performance on both sets. However, as complexity increases, the effectiveness declines. We intend to investigate contrastive fine-tuning of retrievers to enhance retrieval capabilities and implement corrective-RAG for better contextual understanding, thereby delivering more relevant responses. Similarly, for answer generation, we observe that there is some degree of contradiction in answers compared to source passages as indicated by relatively high value for C_S. We aim to explore mechanisms to detect and resolve these contradictions and improve the obligation coverage (OC_S) by updating the answer generation prompt or potentially using a secondary prompt for refinement.

	Development		**Test**	
Subset	**R@10**	**M@10**	**R@10**	**M@10**
Full	75.7	60.3	75.3	59.7
G1	98.4	43.1	99.4	43.0
G2	71.0	23.1	72.7	25.8
G3	72.1	25.3	69.8	25.5
G4	60.1	23.9	58.7	23.1
G10	55.2	19.3	55.0	18.5
P1	84.0	32.8	83.9	33.0
P2	53.8	20.6	52.8	19.7
P3	38.8	14.5	35.2	15.3
P4	21.7	9.5	24.3	8.7
P5	36.7	19.0	26.7	12.0
P6	16.7	2.1	16.7	8.0

Table 1: Results of our approach on the development and test sets. Recall @10 and MAP@10 are reported on the Full dataset and on different groups and passage retrievals.

Type	E_S	C_S	OC_S	RePASs
Development	83.1	28.8	18.6	57.6
Test	82.7	28.1	19.4	58.0

Table 2: Performance of RePASs on Development and Test sets.

You are an editor of a regulatory magazine.
You are given a regulatory question and a set of passages as premise.
Your task is to understand the question and answer based on the provided passages.
——————————————-
Here is the question: <question>
——————————————-
Here are the passages: <passages>
——————————————-
Your task is to carefully read the question and generate an answer based on the provided passages.

While generating the answer from the passages, you must strictly adhere to the following instructions:
1. The answer must not contain any statements that contradict the information in the passages.
2. The answer must encompass all the obligations present in the passages, ensuring that all critical regulatory obligations are reflected.
3. Every sentence in the answer must be substantiated by a corresponding sentence in the source passage.
4. Do not amalgamate information from multiple passages into a single sentence. Instead, create distinct sentences for each piece of information.
5. Provide a detailed answer, ensuring that all relevant information from the passages is included.

Generate an answer for the above question by strictly following these instructions, using the provided passages as the premise.
Output only the answer and nothing else. There is no need to justify or provide citations to the passages.
Simply generate the answer for the provided question based on the passages.
GENERATED ANSWER:

Table 3: Answer Generation Prompt for Subtask 2

Question:

With respect to Financial Instruments and Regulated Activities documentation under point (b),
could you please detail the level of granularity required in the records for each transaction or advice provided to a client?

(Original) Answer without re-ranking:

An authorised person must, pursuant to rule 6.8.2(b), make a record of the following in respect of each transaction:
the identity and account number of the client for whom the transaction was executed, or an indication that the transaction was for its own account;
the name of the counterparty;
the date and time where the transaction was executed;
the identity of the employee executing the transaction;
the investment, including the number of instruments or their value and price;
and whether the transaction was a purchase or a sale, long, short, buyer, seller or other relevant position.

(New) Answer with re-ranked passages:

An authorised person must, pursuant to rule 6.8.2(b), make a record of the following in respect of each transaction:
the identity and account number of the client for whom the transaction was executed, or an indication that the transaction was for its own account;
the name of the counterparty;
the date and time where the transaction was executed;
the identity of the employee executing the transaction;
the investment, including the number of instruments or their value and price;
and whether the transaction was a purchase or a sale, long, short, buyer, seller or other relevant position.
Additionally, an authorised person must, for a minimum of six years, maintain sufficient records in relation to each activity and function of the authorised person
including any financial instruments provided to or regulated activities carried out for the benefit of a client and each advice or recommendation made to a client.

Ground truth context passage(s) missed before re-ranking:

An authorised person must, for a minimum of six years, maintain sufficient records in relation to each activity and function of the authorised person. these must include, where applicable, the following:
(a) any marketing material issued by, or on behalf of, the authorised person;
(b) any financial instruments provided to or regulated activities carried out for the benefit of a client and each advice or recommendation made to a client;
(c) documents regarding client classification under chapter 2;
(d) a record of each client agreement including any subsequent amendments to it as agreed with the client;
(e) records relating to the suitability assessment undertaken by the authorised person to demonstrate compliance with these rules;
(f) records to demonstrate compliance with the requirements relating to inducements, including any disclosure made to clients under that rule
and if any goods and services are received by the authorised person under a soft dollar agreement, the details relating to those agreements;
(g) financial promotions under schedule 2 of fsmr; and
(h) any other disclosures made to clients.

Table 4: Example of an input question and generated answers with/without re-ranking. In this example, RankGPT correctly re-ranked a reference passage missed by the baseline retriever, and subsequently the generated answer captures this necessary information while the original answer did not.

References

Elliott Ash and Daniel L Chen. 2017. Judge embeddings: Toward vector representations of legal belief. Technical report, Technical report.

Diego Collarana, Timm Heuss, Jens Lehmann, Ioanna Lytra, Gaurav Maheshwari, Rostislav Nedelchev, Thorsten Schmidt, and Priyansh Trivedi. 2018. A question answering system on regulatory documents. In *Legal knowledge and information systems*, pages 41–50. IOS Press.

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

Gloria T. Lau, Kincho H. Law, and Gio Wiederhold. 2003. Similarity analysis on government regulations. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, page 711–716, New York, NY, USA. Association for Computing Machinery.

Patrick 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. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing.

Antoine Louis and Gerasimos Spanakis. 2021. A statutory article retrieval dataset in french. *arXiv preprint arXiv:2108.11792*.

Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. 2023. Zero-shot listwise document reranking with a large language model.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023. Rankvicuna: Zero-shot listwise document reranking with open-source large language models.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm.

Guilherme Moraes Rosa, Ruan Chaves Rodrigues, Roberto Lotufo, and Rodrigo Nogueira. 2021. Yes, bm25 is a strong baseline for legal case retrieval.

Gerrit Schumann, Katharina Meyer, and Jorge Marx Gomez. 2022. Query-based retrieval of german regulatory documents for internal auditing purposes. In *2022 5th International Conference on Data Science and Information Technology (DSIT)*, pages 01–10. IEEE.

Weihang Su, Qingyao Ai, Yueyue Wu, Yixiao Ma, Haitao Li, Yiqun Liu, Zhijing Wu, and Min Zhang. 2024. Caseformer: Pre-training for legal case retrieval based on inter-case distinctions.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is chatgpt good at search? investigating large language models as re-ranking agents.

Guoyu Tang, Honglei Guo, Zhili Guo, and Song Xu. 2016. Matching law cases and reference law provision with a neural attention model. *IBM China Research, Beijing*.

Botao Zhong, Wanlei He, Ziwei Huang, Peter E.D. Love, Junqing Tang, and Hanbin Luo. 2020. A building regulation question answering system: A deep learning methodology. *Advanced Engineering Informatics*, 46:101195.

RIRAG: A Bi-Directional Retrieval-Enhanced Framework for Financial Legal QA in ObliQA Shared Task

Xinyan Zhang[1], Xiaobeng Feng*[2], Xiujuan Xu[1], Zhiliang Zheng[3], Kai wu[3],

[1]School of Software Technology,Dalian University of Technology,Dalian,China,
[2]Shanghai University of International Business and Economics,Shanghai,China
,Institute of Change Management and Artificial Intelligence,
[3]School of Computer Science and Technology,Hangzhou Dianzi University,Hangzhou,China,
Correspondence: fxb@suibe.edu.cn

Abstract

In professional financial-legal consulting services, accurately and efficiently retrieving and answering legal questions is crucial. Although some breakthroughs have been made in information retrieval and answer generation, few frameworks have successfully integrated these tasks. Therefore, we propose RIRAG (Retrieval-In-the-loop Response and Answer Generation), a bi-directional retrieval-enhanced framework for financial-legal question answering in ObliQA Shared Task. The system introduces BDD-FinLegal, which means Bi-Directional Dynamic finance-legal, a novel retrieval mechanism specifically designed for financial-legal documents, combining traditional retrieval algorithms with modern neural network methods. Legal answer generation is implemented through large language models retrained on expert-annotated datasets. Our method significantly improves the professionalism and interpretability of the answers while maintaining high retrieval accuracy. Experiments on the ADGM dataset show that the system achieved a significant improvement in the Recall@10 evaluation metric and was recognized by financial legal experts for the accuracy and professionalism of the answer generation. This study provides new ideas for building efficient and reliable question-answering systems in the financial-legal domain. The code of our system is available at https://github.com/Mira-dahu/RIRAG

1 Introduction

Financial-legal question answering systems have emerged as crucial tools for improving access to specialized legal information and services in the financial sector. The complexity of financial-legal documents, combined with the need for accurate and context-aware responses, presents unique challenges in natural language processing. This paper introduces RIRAG, a hybrid system that combines our novel BDD-FinLegal retrieval mechanism, cross-encoding, and advanced language models specifically trained for financial-legal domain question answering.

Recent advances in large language models have revolutionized question answering systems, yet their application in the financial-legal domain remains challenging due to the need for precise citation and adherence to financial regulatory frameworks. Previous approaches have either focused solely on retrieval accuracy or generation quality, often failing to maintain a balance between both aspects. Therefore, we have constructed a completely new system and employed innovative models to address the aforementioned issues. In brief, the contributions of our work are as follows:

- Innovative search mechanism: proposes the BDD-FinLegal dynamic search architecture, which intelligently adjusts traditional and dense embedding methods through query features to achieve more accurate legal document retrieval

- Semantically precise reordering technology: designs a specialized cross-encoder reordering mechanism to significantly improve the relevance and accuracy of legal document retrieval

- Answer generation framework adapted across legal systems: constructs a dual model approach of localization and globalization; achieves comprehensive coverage of legal knowledge in different jurisdictions; and ensures the traceability and professionalism of answers based on expert-annotated datasets

The rest of this paper is structured as follows: Section 2 provides a comprehensive review of existing research on question answering and retrieval systems, identifying key challenges in the domain. Section 3 details our methodology and system

Proceedings of the 31st International Conference on Computational Linguistics, pages 107–113
January 19–24, 2025. ©COLING 2025

architecture, including the novel BDD-FinLegal mechanism. Section 4 presents the experimental results and analysis. Section 5 discusses the implications and limitations, and Section 6 concludes.

2 Related Work

2.1 Legal Question Answering Systems

Recent advances in natural language processing have yielded sophisticated solutions, moving beyond traditional rule-based systems and keyword matching (Ashley, 2017). Some researchers approach legal QA by utilizing ontologies and knowledge graphs, framing it as an information retrieval challenge (Sovrano et al., 2024). While information retrieval (IR) techniques remain dominant for handling legpal documents and queries (Martinez-Gil, 2023), utilizing large language models represents a promising yet underexplored domain in legal technology.

2.2 Information Retrieval Methods

Dense retrieval has become pivotal in IR with deep neural networks (Luo et al., 2024), demonstrating advantages through continuous vector representations thapt capture semantic relationshipsp. Notable works like DPR (Karpukhin et al., 2020) and ANCE (Xiong et al., 2020) have shown strong performance in open-domain QA tasks. However, applying these methods to legal domains presents unique challenges with terminology, document structure, and citation relationships.

Cross-encopder models have proven effective in reranking initial retrieval results (Nogueira and Cho, 2020), with recent architectures including encoder-decoder and decoder-only models (Déjean et al., 2024). Legal-specific approaches emphasize citation-aware reranking, precedent-based scoring, and hierarchical document structures.

2.3 Hybprid System

Hybrid systems combining multiple components (Zhang et al., 2021) typically employ broad retrieval followed by precise reranking and contextual answer generation. However, current methods often lack context sensitivity and rely heavily on single evaluators familiar with policy corpora (Kalra et al., 2024). Our work builds upon these approaches by introducing novel components specifically designed for legal question answering challenges.

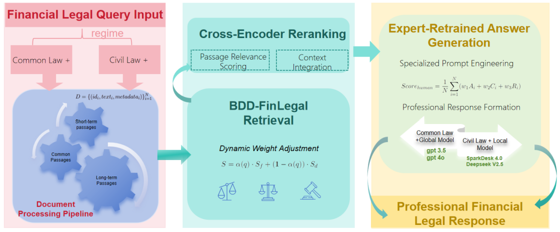

Figure 1: **: An illustration of our system for retrieval-Generation in Legal Question Answering.**

3 Methodology

This section details the architecture and implementation of our RIRAG system, comprising three main components: BDD-FinLegal Retrieval, Cross-Encoder Reranking, and Expert-Retrained Answer Generation.

3.1 System Architecture

The RIRAG system employs a modular architecture designed to handle complex financial legal queries effectively. The system workflow consists of three primary stages:

1. Initial retrieval using the novel BDD-FinLegal approach
2. Reranking of retrieved passages using a specialized cross-encoder
3. Context-aware answer generation leveraging retrained financial-legal expertise

3.2 BDD-FinLegal Retrieval Mechanism

Our novel BDD-FinLegal retrieval mechanism is specifically designed for financial-legal document retrieval:

3.2.1 Dynamic Weight Adjustment

The system implements a sophisticated adaptive weighting scheme:

$$S = \alpha(q) \cdot S_f + (1 - \alpha(q)) \cdot S_d \quad (1)$$

where S_f represents the traditional retrieval score and S_d represents the dense retrieval score.

3.2.2 Adaptive Weighting Scheme

The weight $\alpha(q)$ is dynamically adjusted based on query characteristics:

$$\alpha(q) = \begin{cases} 0.7 & \text{if } |q| < 5 \\ 0.5 & \text{if } 5 \leq |q| < 10 \\ 0.3 & \text{otherwise} \end{cases} \quad (2)$$

3.3 Document Processing Pipeline

The system implements a robust document processing pipeline:

$$D = \{(id_i, text_i, metadata_i)\}_{i=1}^{N} \quad (3)$$

where each document contains:

- Unique identifier (`DocumentID`)
- Passage text
- Passage metadata including `PassageID`

3.4 Expert-Retrained Answer Generation

The answer generation component employs a structured approach with financial legal expertise:

3.4.1 Specialized Prompt Engineering

We implement a domain-specific prompt template as follows: "System: Professional ADGM financial-legal advisor. Guidelines: 1. Base answers on provided financial regulations. 2. Cite specific legal provisions. 3. Use professional financial-legal terminology. 4. Ensure logical completeness. 5. State when information is unavailable."

3.4.2 Context Integration

Retrieved passages are integrated using:

$$C = \sum_{i=1}^{k} w_i \cdot P_i \quad (4)$$

where w_i represents the relevance score and P_i represents the i-th passage.

4 Experiments and Results

This section presents our experimental setup, evaluation metrics, and comparative analysis of different retrieval and generation approaches.

4.1 Experimental Setup

We conducted experiments on the ObliQA dataset[1](Gokhan et al., 2024)

The legal documents included in this dataset cover a range from specific national natural resource assets to current virtual products or services. To address the differences in legal systems across various jurisdictions, we selected two categories of large language models for experimentation: local models and global models. The local models are optimized for Civil Law, while the global models aim to capture the legal principles and applicability of Common Law.

[1]https://github.com/RegNLP/ObliQADataset/tree/main

4.2 Retrieval Performance Analysis

We compared different retrieval approaches. See Table 1 for the comparison results. We can derive several key insights from the outcomes of our results, our BDD-FinLegal method significantly outperforms traditional approaches across all metrics, achieving a 13.8% improvement in Recall@10 compared to the closest baseline.

Method	**R@10**	**MRR**	**N@10**
TF-IDF	0.456	0.312	0.378
BM25	0.583	0.425	0.491
Dense Retrieval	0.621	0.467	0.535
BDD-FinLegal (Ours)	**0.759**	**0.667**	**0.755**

Table 1: Comparison of Different Retrieval Methods

4.3 Answer Generation Evaluation

Our system was evaluated on the ObliQA Datasets. Table 2 in the appendix shows the results, where Expert-Retrained achieved an overall RePASs score of 0.472, demonstrating the framework's capability in handling financial legal queries. We evaluated multiple language models for answer generation. See Table 3 in the appendix for the evaluation results. From Table 3, we can observe that our Expert-Retrained model demonstrates substantial improvements in both generation quality and accuracy.

4.4 Ablation Studies

We conducted ablation studies to analyze the contribution of each component. The performance difference is calculated as follows. P is the abbreviation of performance:

$$\Delta P = P_f - P_a \quad (5)$$

The ablation study results in Table 4 in Appendix part demonstrate the crucial role of each component in our system's performance. Notably, the removal of the BDD-FinLegal mechanism resulted in the largest performance drop (-8.4%), highlighting its importance in the overall framework.

4.5 Human Expert Evaluation

Legal experts evaluated system outputs based on professional accuracy, citation completeness, and response coherence. Obtain more subjective and nuanced assessment results to help validate the accuracy of automated assessment methods.The score is calculated as follows:

$$Score_{human} = \frac{1}{N}\sum_{i=1}^{N}(w_1 A_i + w_2 C_i + w_3 R_i) \quad (6)$$

We evaluate using a subset of 40 comprehensive legal documents from the ObliQA dataset, ensuring balanced coverage of domestic-specific regulations and international financial service frameworks. The 40 legal documents are comprehensive enough, ranging from domestic-specific natural resource assets to current virtual products or services. For representative questions randomly selected from the ObliQA dataset, we conduct similarity comparison experiments using Chinese SparkDesk and Deepseek. See Appendix Figure 2 for the results of the experiment.

Where A_i is the accuracy score, C_i is the citation score, R_i is the response coherence score, and w_1, w_2, w_3 are the respective weights. The citation completeness metric directly corresponds to the obligation coverage measure used in RePASs evaluation, providing complementary human validation of our automated metrics.

Detailed evaluation results comparing Global and Local models across different legal systems are presented in Table 5 in the Appendix. The comparison particularly highlights significant differences in handling jurisdiction-specific questions, especially in cases involving financial market infrastructure and liquidation scenarios

5 Discussion

5.1 Key Insights and Implications

Our research provides several significant insights into financial-legal question answering systems:

- The proposed BDD-FinLegal retrieval mechanism demonstrates the effectiveness of dynamically adjusting retrieval strategies based on query characteristics. This approach addresses the inherent variability in financial-legal queries.
- The cross-encoder reranking mechanism significantly enhances the relevance and precision of retrieved passages, a critical aspect in legal document retrieval.
- Expert-retrained language models show substantial improvements in generating contextually accurate and professionally formatted legal responses.

5.2 Limitations

Despite the promising results, our research has several limitations:

The current system faces challenges in data representativeness and potential bias, with restricted generalizability across different legal jurisdictions. Ethical concerns include inherent biases in expert-annotated datasets and the need for robust privacy protection. The dynamic weighting mechanism, while effective, relies on a simple heuristic that requires more sophisticated adaptive strategies. Additionally, the substantial computational resources needed for training and inference may impede widespread deployment.

5.3 Future Research Directions

Future work could focus on:

- Expanding the approach to multi-lingual and cross-jurisdictional legal question answering systems.
- Developing more nuanced adaptive retrieval mechanisms that consider semantic complexity beyond the existing mechanisms..
- Investigating continual learning approaches to keep the system updated with evolving legal frameworks.

6 Conclusion

In this paper, we introduced RIRAG, a bi-directional retrieval-enhanced framework for financial-legal question answering in ObliQA Shared Task. Our key contributions are a dynamic BDD-FinLegal retrieval mechanism adapting strategies based on query characteristics, a specialized cross-encoder reranking approach enhancing passage relevance, and an expert-retrained answer generation framework maintaining high professional standards. Experimental results on the ADGM financial-legal dataset showed significant improvements in retrieval accuracy, answer quality, and expert evaluation metrics, with a Recall@10 of 0.759 and an expert evaluation score of 0.834, outperforming existing approaches.

Our work provides a promising direction for developing more accurate, interpretable, and reliable question-answering systems in the financial-legal domain. By combining advanced retrieval techniques, neural reranking, and domain-specific language models, we have addressed critical challenges in legal information access.

References

Kevin D Ashley. 2017. Artificial intelligence and legal analytics: new tools for law practice in the digital age.

Hervé Déjean, Stéphane Clinchant, and Thibault Formal. 2024. A thorough comparison of cross-encoders and llms for reranking splade. *arXiv preprint arXiv:2403.10407*.

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.

Rishi Kalra, Zekun Wu, Ayesha Gulley, Airlie Hilliard, Xin Guan, Adriano Koshiyama, and Philip Treleaven. 2024. Hypa-rag: A hybrid parameter adaptive retrieval-augmented generation system for ai legal and policy applications. *arXiv preprint arXiv:2409.09046*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense passage retrieval for open-domain question answering. *Preprint*, arXiv:2004.04906.

Kun Luo, Minghao Qin, Zheng Liu, Shitao Xiao, Jun Zhao, and Kang Liu. 2024. Large language models as foundations for next-gen dense retrieval: A comprehensive empirical assessment. *arXiv preprint arXiv:2408.12194*.

Jorge Martinez-Gil. 2023. A survey on legal question–answering systems. *Computer Science Review*, 48:100552.

Rodrigo Nogueira and Kyunghyun Cho. 2020. Passage re-ranking with bert. *Preprint*, arXiv:1901.04085.

Francesco Sovrano, Monica Palmirani, Salvatore Sapienza, and Vittoria Pistone. 2024. Discolqa: zero-shot discourse-based legal question answering on european legislation. *Artificial Intelligence and Law*, pages 1–37.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *Preprint*, arXiv:2007.00808.

Han Zhang, Hongwei Shen, Yiming Qiu, Yunjiang Jiang, Songlin Wang, Sulong Xu, Yun Xiao, Bo Long, and Wen-Yun Yang. 2021. Joint learning of deep retrieval model and product quantization based embedding index. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 1718–1722. ACM.

Method	**E_s**	**C_s**	**OC_s**	**RePASs**
BDD-FinLegal+Deepseek	0.418	0.389	0.387	0.472

Table 2: Results of the answer generation task using RePASs on the evaluation dataset. E_s, C_s, OC_s, and RePASs represent Entailment, Contradiction, Obligation Coverage and RePAS score, respectively.

A Appendix

A.1 Experiment Details

In this section, we will show you some detailed result of experiment.

Model	**S_c**	**L_c**	**A_c**
Base LLM	0.412	0.385	0.723
Fine-tuned LLM	0.445	0.401	0.756
Expert-Retrained (Ours)	**0.502**	**0.458**	**0.834**

Table 3: Comparison of Answer Generation Models

Component	Performance	Relative
Full System	0.834	-
w/o BDD-FinLegal	0.750	-8.4%
w/o Cross-encoder	0.777	-5.7%
w/o Regime-judegment	0.760	-7.4%
w/o Expert-Retrained	0.765	-6.9%

Table 4: Ablation Study Results

A.2 Web interface display

This is a simple web page that we designed for our hybrid-system.

[2]Source: https://adgmen.thomsonreuters.com/rulebook/fund-rules-funds-ver08040723

Q_{Index}	ModelName	Es	Cs	Ocs	RePASs	Maxs	Mins	AverageScore	AnswerSimilarity
1.0	GPT-4o	0.9615	0.3122	0.3333	0.6609	0.9615	0.3122	0.5670	0.9785
1.0	DeepSeek	0.8875	0.3030	0.1111	0.5652	0.8875	0.1111	0.4667	0.9785
2.0	GPT-4o	0.4624	0.2153	0.3333	0.5268	0.5268	0.2153	0.3845	0.9835
2.0	DeepSeek	0.1712	0.2850	0.3333	0.4065	0.4065	0.1712	0.2990	0.9835
3.0	GPT-4o	0.0536	0.2196	0.0000	0.2780	0.2780	0.0000	0.1378	0.9828
3.0	DeepSeek	0.0737	0.0798	0.3000	0.4313	0.4313	0.0737	0.2212	0.9828
4.0	GPT-4o	0.4357	0.3469	0.8000	0.6296	0.8000	0.3469	0.5530	0.9743
4.0	DeepSeek	0.4963	0.2996	0.4000	0.5322	0.5322	0.2996	0.4320	0.9743
5.0	GPT-4o	0.3253	0.2977	0.3000	0.4425	0.4425	0.2977	0.3414	0.9867
5.0	DeepSeek	0.2791	0.2335	0.2000	0.4152	0.4152	0.2000	0.2820	0.9867
6.0	GPT-4o	0.3572	0.3164	0.4000	0.4803	0.4803	0.3164	0.3885	0.9862
6.0	DeepSeek	0.4046	0.2657	0.7000	0.6130	0.7000	0.2657	0.4958	0.9862
7.0	GPT-4o	0.2427	0.2242	0.1667	0.3951	0.3951	0.1667	0.2572	0.9871
7.0	DeepSeek	0.2018	0.2368	0.0000	0.3217	0.3217	0.0000	0.1901	0.9871
8.0	GPT-4o	0.4495	0.7150	0.3333	0.3560	0.7150	0.3333	0.4634	0.9862
8.0	DeepSeek	0.4988	0.7244	0.4444	0.4063	0.7244	0.4063	0.5185	0.9862
9.0	GPT-4o	0.9806	0.0817	0.6000	0.8330	0.9806	0.0817	0.6238	0.8194
9.0	DeepSeek	0.2440	0.4807	0.0000	0.2544	0.4807	0.0000	0.2448	0.8194
10.0	GPT-4o	0.9886	0.0064	0.8889	0.9570	0.9886	0.0064	0.7102	0.7820
10.0	DeepSeek	0.4230	0.3672	0.3333	0.4630	0.4630	0.3333	0.3967	0.7820

Table 5: Comparative Analysis of Global and Local Models on Legal System-Specific Questions. This table presents a detailed comparison between GPT-4o (representing Global Model + Common Law approach) and DeepSeek (representing Local Model + Civil Law approach) across 10 representative questions from the ObliQA dataset. The evaluation metrics include: Es (Embedding Similarity Score), Cs (Citation Score measuring accurate legal reference usage), Ocs (Obligation Coverage Score), RePASs (Response Professional Accuracy Score), and AnswerSimilarity (similarity score between model outputs). Notable observations: 1) Question 1 demonstrates a clear divergence between Common Law and Civil Law approaches, with GPT-4o showing higher scores across most metrics (Es: 0.96 vs 0.89), reflecting different legal interpretations between the two systems. 2) Questions 9 and 10, which deal with clearing house operations during financial crises, show significant performance gaps. GPT-4o achieves notably higher scores (Es: 0.98, Ocs: 0.89 for Q10) compared to DeepSeek (Es: 0.42, Ocs: 0.33), indicating stronger capabilities in handling complex financial infrastructure scenarios. 3) The overall trend suggests that while both models perform competently, the Global Model (GPT-4o) generally demonstrates more consistent performance across diverse legal contexts, particularly in scenarios requiring cross-jurisdictional understanding. The evaluation was conducted using a subset of 40 comprehensive legal documents, ensuring balanced coverage of both domestic-specific regulations and international financial services frameworks.

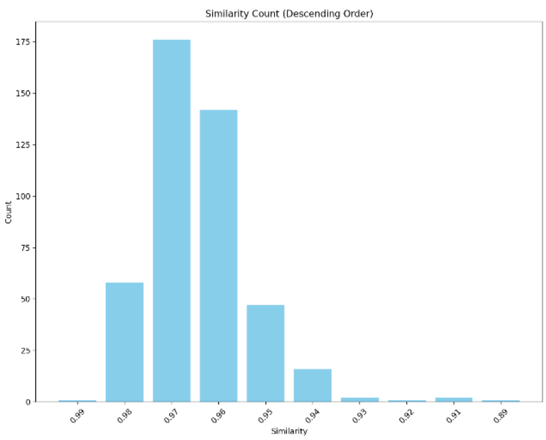

Figure 2: Experimental results of document similarity across legal fields (SparkDesk & Deepseek). We identified the lowest 5 scores corresponding as follows. The lowest scored two queries on virtual assets, followed by the queries on identification of contravention, lastly on definition of a term. The difference might be a results of different jurisdiction applies different legal regimes.[2]

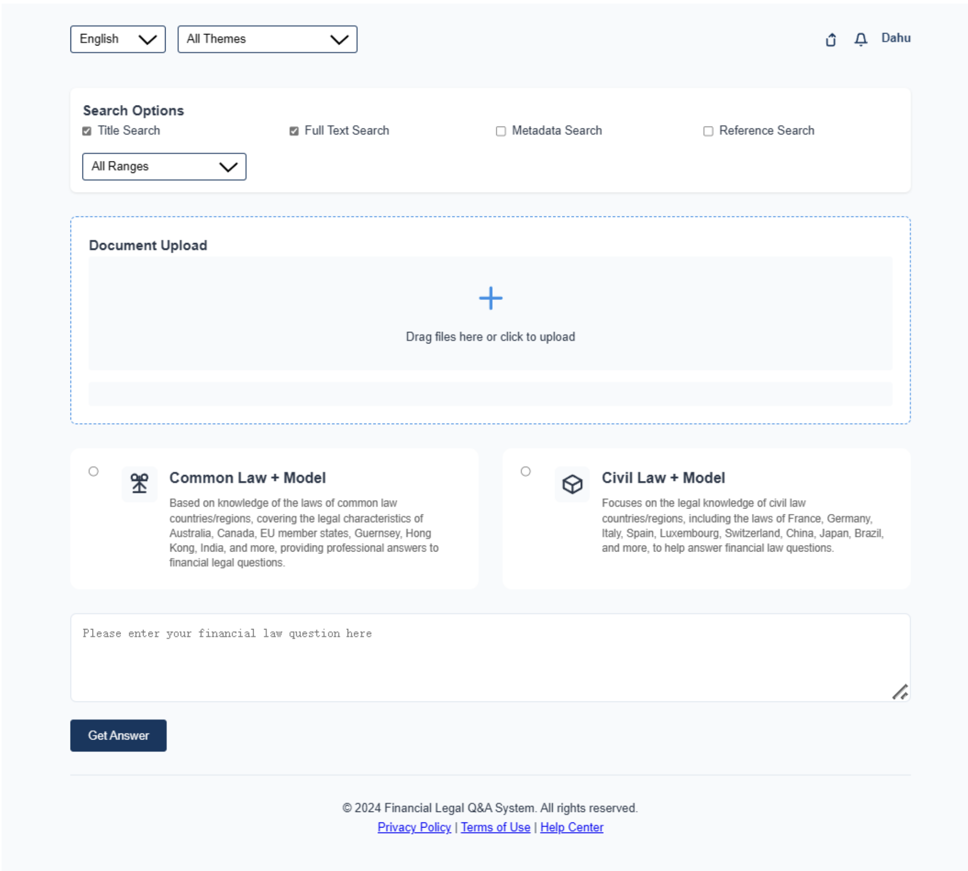

Figure 3: The web interface of the financial law QA system. The interface provides language and topic selection, multiple search options, a document upload function, and the choice of a model based on the common law system and the civil law system. Users can enter questions to get answers. The bottom contains copyright notices and related links.

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. ©COLING 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	70.09
	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 RegNLP 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

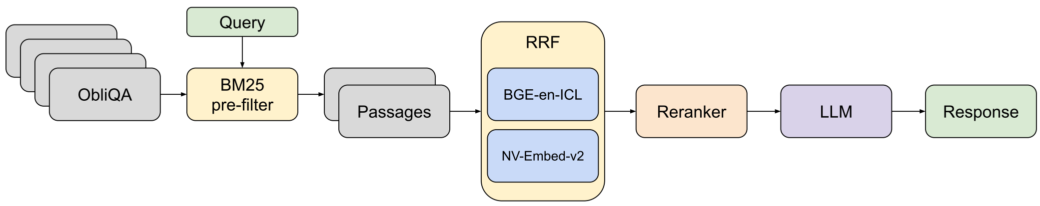

Figure 1: Retrieval and Generation Pipeline

Author Index