Empirical Analysis of Text Segmentation for Statute-Level Retrieval in Regulatory RAG Pipelines

1st Beomseok Kim
Dept. of Medical IT
Inje University
Gimhae, South Korea
qjatjr9958@oasis.inje.ac.kr

2nd Hoansuk Choi
Dept. of Game Engineering
Kyungnam University
Changwon, South Korea
chs2024@kyungnam.ac.kr

3rd Namhyun Yoo
Dept. of Computer Engineering
Inje University
Gimhae, South Korea
hyun43@kyungnam.ac.kr

4th Jinhong Yang
Dept. of Medical IT
Inje University
Gimhae, South Korea
jinhong@inje.ac.kr

***Abstract*—Retrieval-Augmented Generation (RAG) is increasingly adopted in the legal domain to mitigate hallucinations and ensure regulatory compliance. However, standard fixed-size chunking strategies often disrupt the logical hierarchy of legal texts, leading to context fragmentation. This paper investigates the impact of chunking strategies on retrieval performance by comparing fixed-size chunking (256, 512, and 1024 tokens) with a structure-aware chunking method based on the inherent hierarchy of legal articles (Article–Paragraph–Subparagraph). We construct a synthetic dataset using the Personal Information Protection Act (PIPA) of South Korea and evaluate retrieval performance using Hit@k and Mean Reciprocal Rank (MRR) under both lexical (BM25) and semantic (dense) retrieval settings. Experimental results reveal a performance inversion: smaller fixed chunks (256 tokens) benefit keyword-based search due to length bias, while the proposed structure-aware method achieves retrieval performance comparable to large-context strategies in semantic retrieval (MRR 0.737 vs. 0.736, i.e., a numerically small difference) while preserving complete legal provisions. Furthermore, we detail the specific experimental environment, including the `klue/roberta-base` model configuration, to ensure reproducibility. Although these results are obtained in a single-statute, retrieval-only setting, they provide an initial empirical basis for designing retrieval components in statute-level RAG pipelines. Overall, our findings suggest that structure-aware chunking offers a promising trade-off between retrieval precision and semantic completeness, helping minimize the risk of fragmented legal contexts in statute-level RAG systems.**

***Index Terms*—Retrieval-Augmented Generation, Legal AI, Text Chunking, Structure-Aware, BM25, Dense Retrieval**

I. Introduction

Large Language Models (LLMs) are transforming domains requiring specialized knowledge, such as law, medicine, and finance [1]. In particular, Retrieval-Augmented Generation (RAG), which grounds model responses in external knowledge bases, has become a core technology for Regulatory Compliance systems by mitigating hallucinations and incorporating up-to-date legal statutes [2].

However, the impact of *text segmentation strategies (chunking)* strategies—a critical component that determines RAG performance—has been relatively overlooked. **Widely adopted frameworks, such as LangChain and LlamaIndex, predominantly rely on fixed-size sliding window strategies by default** [3]. While recent LLMs support extended context windows, relying solely on long context input is inefficient due to the **"Lost in the Middle"** phenomenon, where models fail to retrieve information located in the middle of long contexts [4]. Therefore, optimizing retrieval units remains a crucial challenge.

In legal texts, mechanical segmentation poses severe risks. Legal statutes possess strict hierarchical structures (Article-Paragraph-Subparagraph) and frequent **cross-references** (e.g., "subject to Article X") [5]. Fixed-size chunking often leads to **Context Fragmentation**, where the 'condition' of a law and the 'penalty' for violating it, or the referenced clauses, are separated into different chunks. This physical separation prevents the LLM from accessing the complete legal logic required for accurate reasoning.

This study investigates the fundamental question: *"Is the conventional fixed-size chunking optimal for legal RAG systems?"* We focus on the **Personal Information Protection Act (PIPA)** of South Korea, a high-stakes domain where compliance violations can lead to severe financial penalties. Using this data set, we quantitatively compare the retrieval performance of fixed-size chunking against a proposed **Structure-Aware Chunking** strategy.

Specifically, this paper makes the following contributions:

- We propose a structure-aware chunking algorithm tailored to the hierarchical nature of statutory law.
- We construct a synthetic QA dataset on the Korean Personal Information Protection Act (PIPA) with explicit article-level ground truth mappings.
- We empirically reveal (i) a consistent performance inversion between keyword-based and semantic retrieval under different chunk sizes, and (ii) a saturation effect beyond 512 tokens, offering practical guidance for configuring retrieval components in statute-level RAG systems.

II. Related Work

A. Retrieval-Augmented Generation in the Legal Domain

Retrieval-Augmented Generation (RAG) has emerged as a promising approach to enhance the factual accuracy and domain adaptability of LLMs [6]. In the legal domain, where hallucination can lead to severe consequences, RAG systems are widely adopted to ground model responses in authoritative statutes and precedents [7]. However, previous

studies have predominantly focused on *Case Law* retrieval or fine-tuning LLMs for legal reasoning. In contrast, *Statutory Law*—characterized by rigorous hierarchical structures (e.g., Articles and Paragraphs)—requires a distinct retrieval approach [8]. Unlike unstructured legal narratives, statutes possess high logical density where a single missing clause can invert the legal interpretation. Consequently, the optimization of the retrieval component, specifically preserving the structural integrity of statutes, remains underexplored.

B. Advanced Text Segmentation Strategies

Effective indexing requires segmenting long documents into manageable units. Standard RAG frameworks typically employ **Fixed-size Chunking** due to its simplicity and computational efficiency [9]. However, this mechanical splitting often severs the semantic link between a condition and its legal effect, leading to *Context Fragmentation*. Recently, *Semantic Chunking*, which utilizes embedding similarities to identify topic boundaries, has been proposed to mitigate this issue [10]. Yet, in the legal domain, explicit structural boundaries (e.g., Article numbers) serve as stronger semantic delimiters than latent embedding shifts. Despite the importance of structural preservation, there has been little empirical work that systematically compares fixed-size and structure-aware chunking strategies [11]. Our work addresses this gap by validating the efficacy of structure-aware segmentation, particularly within a non-English regulatory corpus where morphological complexity exacerbates the challenge.

III. Methodology

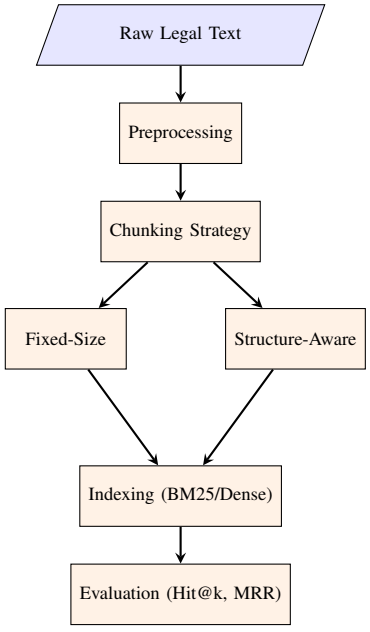

Fig. 1. Overall experimental workflow: from raw legal text processing to performance evaluation comparing fixed-size and structure-aware chunking strategies.

A. Dataset Construction

We utilized the *Personal Information Protection Act* (PIPA) of South Korea as our primary corpus. PIPA exhibits high structural complexity with frequent cross-references between definitions, obligations, and penalties, making it an ideal testbed for evaluating context preservation capabilities in statute-level RAG systems.

To ensure objective evaluation, we generated a **synthetic QA dataset** using rule-based templates. This methodology creates a deterministic mapping between queries and ground truth articles, eliminating the ambiguity or hallucination often introduced by LLM-generated queries. The detailed generation process is presented in **Algorithm 1**. The final dataset consists of **113 articles** and **226 QA pairs** (two queries per article).

It is important to note that our synthetic QA pairs primarily target *article-level identification* rather than free-form legal question answering. This design choice allows us to obtain a clean and deterministic mapping between queries and ground truth articles, which is suitable for isolating the effect of chunking strategies on retrieval behavior. However, it does not fully capture the variety and ambiguity of real-world legal queries (e.g., implicit conditions, multi-hop reasoning across multiple statutes). Therefore, the absolute performance numbers should be interpreted as a controlled proxy for retrieval behavior rather than as an end-to-end legal QA benchmark.

Algorithm 1 Synthetic QA Generation

Require: Articles $\{A_1, ..., A_n\}$
Ensure: QA Dataset D_{qa}
$D_{qa} \leftarrow \emptyset$
for each Article A_i **do**
  $Q_{id} \leftarrow$ "What is the content of Article " + A_i.id + "?"
  $Q_{content} \leftarrow$ "Which article mentions: " + A_i.text[: 50] + "?"
  Add $(Q_{id}, A_i.\text{id})$ to D_{qa}
  Add $(Q_{content}, A_i.\text{id})$ to D_{qa}
end for
return D_{qa}

TABLE I
STATISTICS OF THE PIPA DATASET

Metric	Value
Total Number of Articles	113
Total Tokens (Full Text)	52,000+
Average Tokens per Article	462.6
Min / Max Tokens (per article)	38 / 2,930
Number of QA Pairs	226

B. Chunking Strategies

We compared four chunking conditions to analyze the trade-off between context size and retrieval precision:

- **Fixed-size Chunking (S1, S2, S3):** A sliding-window approach with fixed lengths of 256, 512, and 1,024

tokens. To mitigate boundary effects, we set the **overlap ratio** to approximately 12.5% of each chunk size (i.e., 32, 64, and 128 tokens, respectively).

- **Structure-Aware Chunking (Proposed):** Segmentation based on the *Article* unit to preserve the semantic completeness of legal provisions. When an article exceeds a predefined maximum length L_{max}, it is further split into paragraph level chunks. The detailed process is described in **Algorithm 2**.

Algorithm 2 Structure-Aware Chunking Process

Require: Legal Document D, MaxToken L_{max}
Ensure: Set of Chunks C
$C \leftarrow \emptyset$
Parse D into Articles $\{A_1, A_2, ..., A_n\}$
for each Article A_i in D **do**
  $T \leftarrow$ Tokenize(A_i)
  if Length$(T) \leq L_{max}$ **then**
    Add T to C
  else
    Split A_i into Paragraphs $\{P_1, ..., P_m\}$
    for each Paragraph P_j **do**
      Add Tokenize(P_j) to C
    end for
  end if
end for
return C

In our experiments, we set $L_{\max}$ to 4,096 tokens, which is significantly larger than the maximum article length in the PIPA corpus (2,930 tokens; see Table I). As a result, no article was further split into paragraph-level chunks, leading to a split rate of 0.0% for the Structure-Aware strategy in Table IV. Consequently, on this corpus Structure-Aware Chunking effectively reduces to article-level segmentation (i.e., each article corresponds to a single chunk), while the paragraph-level splitting logic in Algorithm 2 is included for generality when applying the method to longer or more heterogeneous statutes.

C. Retrieval Models and Mathematical Formulation

We evaluated performance using two distinct models to capture both lexical and semantic relevance.

1) BM25 (Lexical Search): We used BM25Okapi, a probabilistic retrieval model. The score for a document D given a query Q containing terms $q_1, ..., q_n$ is calculated as:

$$\text{Score}(D, Q) = \sum_{i=1}^{n} IDF(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)} \quad (1)$$

where $f(q_i, D)$ is the term frequency of q_i in D, $|D|$ is the document length, and avgdl is the average document length in the corpus. We set $k_1 = 1.2$ and $b = 0.75$. This model inherently biases towards shorter documents with higher term density, which is particularly relevant when comparing fixed-size chunks of different lengths.

2) Dense Retrieval (Semantic Search): For semantic retrieval, we encoded both queries and chunks into dense vector representations. The relevance score is defined by the cosine similarity between the query embedding vector $\mathbf{v}_Q$ and the document embedding vector $\mathbf{v}_D$:

$$\text{Sim}(Q, D) = \cos(\theta) = \frac{\mathbf{v}_Q \cdot \mathbf{v}_D}{\|\mathbf{v}_Q\| \, \|\mathbf{v}_D\|} \quad (2)$$

This approach captures the contextual meaning of the text, making it sensitive to semantic fragmentation caused by inappropriate chunk boundaries. In our setting, all chunks produced by the different strategies were encoded and ranked under identical model and hyperparameter configurations, allowing a controlled comparison of chunking effects on retrieval performance.

IV. Experimental Setup

To ensure reproducibility and transparency, we detail our computing environment and the specific configurations of the models used in the experiments.

A. Computing Environment

All experiments were conducted on a cloud-based environment (Google Colab) utilizing GPU acceleration to handle the computational load of dense vector generation.

- **Platform:** Google Colab
- **GPU:** NVIDIA Tesla T4 (15GB VRAM)
- **CPU:** Intel Xeon Processor (2.20GHz, 2 cores)
- **RAM:** 13GB System RAM
- **Software Stack:** Python 3.12, PyTorch 2.9.0, Transformers 4.57.2

B. Model Configuration

1) Tokenizer for Chunking: To maintain consistency with commercial LLM services (e.g., GPT-4) and ensure accurate context window calculations, we employed the `cl100k_base` tokenizer provided by the `tiktoken` library for all chunking strategies.

2) Retrieval Models:

- **Sparse Retriever (BM25):** We utilized the `rank_bm25` library. For this baseline model, we applied simple whitespace tokenization to evaluate pure keyword matching performance without morphological analysis bias.
- **Dense Retriever:** We employed the `jhgan/ko-sroberta-multitask` model, available on HuggingFace. This model is fine-tuned on the `klue/roberta-base` architecture, which is specifically for Korean semantic textual similarity (STS) tasks [12] [13].
- **Pooling Strategy:** We generated 768-dimensional dense vectors by applying **mean-pooling** to the output hidden states. [14].

V. Experiments and Results

A. Quantitative Analysis

Table II and Table III present the retrieval performance in BM25 and dense retrieval environments, respectively, computed over all 226 queries in our synthetic PIPA dataset. Additionally, Table IV compares the structural characteristics of the generated chunks under each strategy.

In the BM25 setting (Table II), the shortest unit, **Fixed-256**, achieved the highest MRR (0.554) and a competitive Hit@1 (0.527), while exhibiting slightly lower Hit@5 than the larger chunk sizes. This pattern is consistent with the well-known length bias of BM25: shorter documents with higher term density tend to receive higher scores when exact keyword overlaps are present. Structure-Aware Chunking, on the other hand, achieved the best Hit@5 (0.558), indicating that article-level units can still provide robust top-k coverage in lexical retrieval despite being longer on average.

In the dense retrieval setting (Table III), performance generally improved as the context window increased from 256 to 1,024 tokens. **Structure-Aware Chunking** matched the best Hit@1 and Hit@5 scores (0.681 and 0.841, respectively) and achieved the highest MRR (0.737), slightly outperforming Fixed-1,024 (MRR 0.736). Although the absolute gaps in MRR between the best-performing strategies are numerically small (at most $\Delta\text{MRR} = 0.008$ across configurations), the ranking of strategies is stable across all queries, suggesting that the observed trends are not merely due to random noise but reflect systematic interactions between chunk size and retrieval model type.

Table IV further shows that these performance patterns arise under markedly different structural profiles. Fixed-size strategies trade off shorter average chunk length (e.g., 198.8 tokens for Fixed-256) against a larger number of chunks and higher split rates (up to 69.9%). In contrast, Structure-Aware Chunking yields an average of 462.5 tokens per chunk with only 113 chunks and a split rate of 0.0%, indicating that each article is preserved as a single coherent retrieval unit.

TABLE II
Retrieval Performance with BM25 (Keyword Search)

Strategy	Hit@1	Hit@5	MRR
Fixed-256	0.527	0.602	**0.554**
Fixed-512	0.531	0.549	0.538
Fixed-1024	0.531	0.549	0.538
Structure-Aware	0.527	0.558	0.538

TABLE III
Retrieval Performance with Dense Retrieval (Semantic)

Strategy	Hit@1	Hit@5	MRR
Fixed-256	0.677	0.827	0.729
Fixed-512	0.681	0.832	0.733
Fixed-1024	0.681	0.841	0.736
Structure-Aware	0.681	0.841	**0.737**

TABLE IV
Comparison of Chunk Characteristics

Strategy	Avg. Tokens	Total Chunks	Split Rate (%)
Fixed-256	198.8	290	69.9%
Fixed-512	319.3	175	38.9%
Fixed-1024	423.6	127	9.7%
Structure-Aware	462.5	113	**0.0%**

B. Performance Inversion and Structural Analysis

A distinct **performance inversion** was observed between the lexical and semantic retrieval settings. In BM25, the shortest unit, **Fixed-256**, achieved the highest MRR (0.554), whereas in dense retrieval, **Structure-Aware Chunking** achieved the best MRR (0.737). This inversion arises because BM25 benefits from shorter documents with concentrated keyword occurrences, while dense retrieval models benefit from richer semantic context.

More concretely, the difference between Fixed-256 and Structure-Aware is $\Delta\text{MRR} = 0.016$ in BM25 (0.554 vs. 0.538), but only $\Delta\text{MRR} = 0.008$ in dense retrieval (0.729 vs. 0.737). Combined with Table IV, this indicates that the gain of Fixed-256 in BM25 comes at the cost of aggressive splitting: 69.9% of articles are fragmented across multiple chunks, and the total number of chunks increases to 290. Structure-Aware Chunking, by contrast, uses slightly longer units (462.5 tokens on average) but preserves each article as a single chunk (0.0% split rate), resulting in only 113 chunks overall.

This structural difference is particularly important in regulatory texts, where conditions, exceptions, and penalties are often tightly coupled within a single provision. Even when retrieval scores are similar, the probability that an LLM observes the *complete* legal logic in a single retrieved chunk is higher under Structure-Aware Chunking than under highly fragmented fixed-size strategies.

C. Impact of Context Saturation

We also observed a **performance saturation** point around 512 tokens in the dense retrieval setting. Fixed-512 and Fixed-1024 showed identical Hit@1 scores (0.681), while the MRR improved only marginally from 0.733 to 0.736. This suggests that increasing the chunk size beyond 512 tokens yields diminishing returns in semantic retrieval performance.

This phenomenon aligns with the dataset statistics: the average article length in the PIPA corpus is approximately 462.6 tokens (Table I). A 512-token window is therefore sufficient to encapsulate the full semantic context of an average legal article, and expanding the window to 1,024 tokens mainly introduces additional computational overhead without providing significant information gain in this single-statute setting. In practice, this implies that practitioners can often avoid excessively large chunk sizes when indexing statute-level corpora for dense retrieval, especially when combined with structure-aware segmentation.

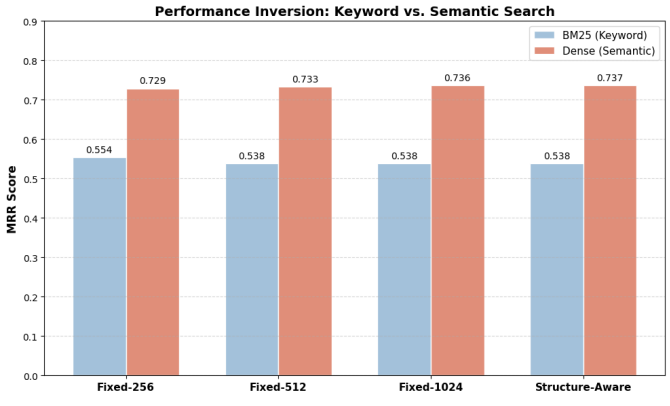

Fig. 2. Contrast in retrieval performance (MRR) between keyword search (BM25) and semantic search (Dense Retrieval) across different chunking strategies.

D. Qualitative Analysis: Case Study

To demonstrate the practical implications of our findings, we analyzed a specific failure case involving **Article 18 (Restriction on Use and Provision of Personal Information)**, a critical provision for compliance.

Consider a user query: *"Can personal information be provided to a third party for crime investigation without the subject's consent?"* In the **Fixed-256** strategy, the relevant article was split into multiple chunks due to the mechanical token limit. The retrieval system successfully fetched the second chunk containing the clause: *"... 7. Where it is necessary for the investigation of crimes..."*. However, the critical constraint mentioned in the article's header—*"provided that this shall apply only to public institutions"*—was located in the previous chunk and was not retrieved. Consequently, an RAG agent relying on this fragmented context may generate a legally non-compliant response, incorrectly advising private companies that they can provide data for investigations.

In contrast, the **Structure-Aware** strategy retrieved Article 18 as a single, coherent unit. By preserving the semantic link between the proviso (constraint) and the subparagraphs (conditions), it provided the LLM with the complete legal logic, allowing for a more accurate and safe interpretation. This case highlights that structural integrity is an important consideration for ensuring the safety of legal AI: even when fixed-size and structure-aware strategies yield similar Hit@k or MRR scores, they can differ substantially in the likelihood of omitting critical legal qualifiers that affect downstream decision making.

VI. Discussion

A. Trade-off and Hybrid Strategy

Our findings suggest a trade-off between retrieval precision and context coherence. Small fixed chunks (Fixed-256) offered higher precision for keyword matching in our experiments but exhibited limitations in preserving semantic context. Conversely, Structure-Aware chunks provided the semantic completeness necessary for LLM reasoning, though they sometimes yielded lower scores when queries lacked semantic depth. Therefore, a **Hybrid Retrieval Pipeline** could be considered as a practical solution: utilizing Fixed-256 chunks for initial high-recall filtering to capture specific keywords, followed by a semantic re-ranking stage using Structure-Aware chunks to enhance context integrity for the final generation. While the absolute performance differences between the best-performing strategies are numerically small (e.g., $\Delta\text{MRR} \leq 0.008$ in Table III), the trends are consistent across all queries and align well with the structural statistics of the corpus (Table IV). In particular, the saturation around 512 tokens coincides with the average article length (approximately 462.6 tokens), suggesting that chunk sizes slightly above the average provision length are sufficient to capture most of the relevant context in single-statute retrieval scenarios.

B. Computational Efficiency

Beyond retrieval performance, system efficiency is an important consideration. We observed that retrieval performance tended to saturate around 512 tokens (Table III). This implies that the average legal article (approx. 462 tokens) fits well within standard embedding models. Consequently, utilizing excessive chunk sizes (e.g., 1024 or larger) may introduce additional computational overhead without providing proportional performance gains. Adopting Structure-Aware chunking or Fixed-512 may thus offer a balanced approach for legal RAG systems.

C. Limitations

First, our experiments relied on a synthetic dataset derived from a single statute (PIPA). While this allowed for controlled evaluation, real-world legal queries often involve implicit reasoning or multi-hop retrieval across various laws, which our current setup might not fully capture. Second, our analysis focused on retrieval metrics (Hit@k, MRR). Although higher semantic relevance often correlates with better generation, future work should explicitly evaluate the downstream generation quality (e.g., faithfulness, answer relevancy) of LLMs when provided with different chunk types. Moreover, we restrict our analysis to retrieval metrics (Hit@k and MRR) in a single-statute setting; extending the evaluation to multi-hop, cross-statute retrieval and to downstream generation quality (e.g., faithfulness and hallucination rates under different chunk types) remains an important direction for future work.

VII. Conclusion

This study empirically analyzed the impact of chunking strategies on retrieval performance for legal RAG systems. Focusing on the Korean Personal Information Protection Act (PIPA), we compared fixed-size sliding-window chunking (256, 512, and 1,024 tokens) with an article-level structure-aware chunking strategy on a synthetic QA dataset consisting of 226 queries with explicit article-level ground-truth mappings. Across BM25 and dense retrieval settings, we observed a performance inversion pattern: keyword search (BM25) favored shorter chunks due to term density, whereas semantic

search (dense retrieval) favored structure-aware chunks that preserved logical boundaries. Furthermore, identifying a performance saturation point at 512 tokens provided insights into how the average article length (approximately 462.6 tokens) interacts with chunk size in statute-level retrieval.

Based on our experimental results, **Structure-Aware Chunking** appears to be a promising strategy for the legal domain. Even with retrieval scores that are numerically comparable to large fixed chunks, it achieved a 0% split rate in our tests, meaning that each article was preserved as a single coherent unit. This suggests that structure-aware segmentation can effectively provide RAG agents with complete legal provisions—inclusive of conditions, exceptions, and penalties—thereby potentially reducing the risk of hallucinations and legal misinterpretation in downstream generation. For practitioners building legal RAG services in ICT environments, our findings imply that moderate fixed chunks around 512 tokens or article-level structure-aware chunks can serve as reasonable default configurations, balancing retrieval effectiveness, semantic completeness, and computational cost.

Finally, while our analysis is restricted to retrieval metrics in a single-statute, synthetic QA setting, it provides an initial empirical basis for designing statute-level RAG pipelines. Extending this line of work to multi-hop, cross-statute retrieval and to end-to-end evaluations of generation quality (e.g., faithfulness and hallucination rates under different chunk types) remains an important direction for future research.

ACKNOWLEDGMENT

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) under the Innovative Human Resource Development for Local Intellectualization program grant funded by the Korea government (MSIT) (IITP-2025-RS-2024-00436773).

REFERENCES

[1] W. Zhang and J. Zhang, "Hallucination mitigation for retrieval-augmented large language models: A review," *Mathematics*, 2025, doi: 10.3390/math13050856.

[2] M. Hindi, L. Mohammed, O. Maaz, and A. Alwarafy, "Enhancing the precision and interpretability of retrieval-augmented generation (RAG) in legal technology: A survey," *IEEE Access*, vol. 13, pp. 46171–46189, 2025, doi: 10.1109/ACCESS.2025.3550145.

[3] N. Pipitone and G. Alami, "LegalBench-RAG: A benchmark for retrieval-augmented generation in the legal domain," *arXiv preprint* arXiv:2408.10343, 2024, doi: 10.48550/arXiv.2408.10343.

[4] B. Veseli, J. Chibane, M. Toneva, and A. Koller, "Positional biases shift as inputs approach context window limits," *arXiv preprint* arXiv:2504.11287, 2025.

[5] A. Louis, G. Van Dijck, and G. Spanakis, "Finding the law: Enhancing statutory article retrieval via graph neural networks," 2023, pp. 2753–2768, doi: 10.48550/arXiv.2301.12847.

[6] J. Li, Y. Yuan, and Z. Zhang, "Enhancing LLM factual accuracy with RAG to counter hallucinations: A case study on domain-specific queries in private knowledge-bases," *arXiv preprint* arXiv:2403.10446, 2024, doi: 10.48550/arXiv.2403.10446.

[7] V. Magesh, F. Surani, M. Dahl, M. Suzgun, C. Manning, and D. Ho, "Hallucination-free? Assessing the reliability of leading AI legal research tools," *arXiv preprint* arXiv:2405.20362, 2024, doi: 10.48550/arXiv.2405.20362.

[8] Hou, A., Weller, O., Qin, G., Yang, E., Lawrie, D., Holzenberger, N. *et al.*, "CLERC: A dataset for legal case retrieval and retrieval-augmented analysis generation," *arXiv preprint* arXiv:2406.17186, 2024, doi: 10.48550/arXiv.2406.17186.

[9] Y. Gao, Y. Xiong, W. Wu, Z. Huang, B. Li, and H. Wang, "U-NIAH: Unified RAG and LLM evaluation for long context needle-in-a-haystack," *arXiv preprint* arXiv:2503.00353, 2025, doi: 10.48550/arXiv.2503.00353.

[10] Peng, B., Zhu, Y., Liu, Y., Bo, X., Shi, H., Hong, C. *et al.*, "Graph retrieval-augmented generation: A survey," *arXiv preprint* arXiv:2408.08921, 2024, doi: 10.48550/arXiv.2408.08921.

[11] J. Ho, A. Colby, and W. Fisher, "Incorporating legal structure in retrieval-augmented generation: A case study on copyright fair use," *arXiv preprint* arXiv:2505.02164, 2025, doi: 10.48550/arXiv.2505.02164.

[12] Park, S., Moon, J., Kim, S., Cho, W., Han, J., Park, J. *et al.*, "KLUE: Korean language understanding evaluation," *arXiv preprint* arXiv:2105.09680, 2021.

[13] Park, Y., Shin, Y.,"Using multiple monolingual models for efficiently embedding Korean and English conversational sentences," *Applied Sciences*, vol. 13, no. 9, 2023, doi: 10.3390/app13095771.

[14] Yang, K., Jang, Y., Lee, T., Seong, J., Lee, H., Jang, H. *et al.* (2023). "KoBigBird-large: Transformation of Transformer for Korean Language Understanding," ArXiv, abs/2309.10339. https://doi.org/10.48550/arxiv.2309.10339.